Preface

This book is about OrCAD SDT 386+, the industry leader in schematic capture software. Unlike the hundreds of books written about mechanical CAD (Computer Aided Design) software such as AutoCAD, few books have been written about EDA (Electronic Design Automation), and this is the first one that covers OrCAD. *Inside OrCAD* was written in recognition of the needs of both new and experienced users. The book is organized into a series of tutorial exercises, preceded by two chapters giving an overview of schematic drafting and OrCAD installation. Most of the tutorial exercises are divided into two or three sessions. Each session can be completed in less than two hours and contains a mix of theory and practice. The disk that accompanies the book contains useful utilities and sample files for the tutorial exercises.

New users can read chapters 1 and 2 and then complete the first four tutorial exercises in chapters 3 through 6 to get off to a fast start using OrCAD. Experienced users can hone their skills with the advanced tutorials starting in Chapter 7. Experienced users can also refer to particular chapters for information about specific subjects such creating netlists for PCB design or editing bill of materials.

Certain prerequisites exist for the use and understanding of the material presented in this book. The reader is assumed to have some knowledge of electronics and schematic drafting. Some degree of PC and DOS literacy is also required. In order to complete the tutorial exercises, the reader will require a minimum 386 class PC with attached Hewlett-Packard LaserJet compatible laser printer and access to OrCAD SDT 386+ software.

Acknowledgments

First, I would like to thank Peter Vanderhelst, the grizzled chief engineer, who taught me the tricks of the trade, including the black art of high power electronics. The sorcerer's apprentice was lucky to have such a wise master.

This book is dedicated to the memory of my Grandfather, Hermann. After he retired from the German civil service, he collected books and wrote articles of historical interest for local newspapers. As a young boy during the 1960s, I spent many summers visiting my grandparents. My Grandfather and I would roam the countryside visiting old ruins and talking to farmers and innkeepers as he researched his articles. When we returned home, he would retire to his study and type on an ancient Olivetti typewriter. He had a definite influence and was a very

positive role model. He taught me to measure a man by his books. Over the years, I have found this to be a very accurate measure indeed.

Bob Pease, author and analog guru, also played an indirect role in this book. His recent book, *Troubleshooting Analog Circuits*, lead me to Butterworth-Heineman and the EDN Series For Design Engineers. Even more so, Bob's columns in *Electronic Design* Magazine reminded me that at a certain point in one's career, the time comes to feedback information to the next generation.

I would like to thank the staff at Crane Cams for encouraging me in this project. Special credit goes to my boss, Bill Gaterman, the vice president of engineering, for giving me the freedom to pursue creative thinking and for looking the other way all the times I've bucked the "system." I am also grateful to Troy Weippert for his suggestions after reading the first part of the manuscript and for absorbing some of the workload while I concentrated on finishing the book.

On a final note, I would also like to thank the staff at Butterworth-Heineman including Jo Gilmore, Liz McCarthy, and Marika Alzadon. Writing one's first book is not always an easy undertaking and usually takes longer than expected. Thanks for being so patient with a new author.

About The Author

Chris Schroeder received his B.S. in Engineering from the University Of Michigan in 1976. He is currently technical director for the automotive electronics business unit of Crane Cams in Daytona Beach, Florida and has previously been involved with sales and marketing of computer graphics equipment and design of industrial electronics. He enjoys playing with electronics and writing. Chris lives with his wife Tina and young son Garrett in Ormond Beach, Florida and spends much of his free time and most of his free money flying.

1

Introduction to Schematic Capture

Schematic diagrams are used to graphically represent the components and interconnections of electrical circuits. In the past, schematics were drafted using manual drawing techniques. Up until the late 1970s, the only schematic drafting aids were plastic drawing templates. The author still has a large collection of schematic, flow chart, and PCB (printed circuit board) design templates from Bishop Graphics and Berol dating back to those days. Other than the use of templates and new symbols for solid state devices, little had changed for almost fifty years. Back in the late 1970s, CAD (Computer Aided Design) systems based on mainframe and mini-computer technology were just starting to see use in larger companies, but these systems were very expensive. At over $100,000 per seat, simple tasks like routine schematic drafting were difficult to cost justify. That situation quickly changed with the advent of the IBM PC and low cost CAD software such as AutoCAD.

OrCAD SDT schematic capture software was introduced in 1985 and was quickly accepted, largely due to ease of use, speedy performance on PC workstations, and low cost. Today, with over 100,000 copies sold, OrCAD SDT has become the most widely used software for schematic capture. While OrCAD SDT was not the first package to offer schematic capture on the PC, it popularized the concept and is now the undisputed industry leader.

When the author first started working in the electronics industry back in the mid-1970s, CAD systems were rare and engineers usually drew up a rough schematic by hand. The circuit was then prototyped on a wire-wrap board. Once the circuit was debugged, the drafting department redrew the schematic and started the PCB layout. Schematic capture is now one of the first steps in the design process, and the term EDA (Electronic Design Automation) is used in place of CAD. In today's environment of competitive pressures to reduce time-to-market and with the widespread use of SMT (surface mount technology), the engineer usually creates the schematic using an EDA tool such as OrCAD SDT as the first step towards generating a prototype printed circuit board. Schematic capture creates a database of components and interconnections along with the graphical schematic.

Via postprocessing steps, this database can automatically generate netlists for printed circuit board and programmable logic device design, bill of materials listings, and various output formats used for timing and circuit analysis. This database capability results in significant cost and time savings in the overall design process.

While the focus of this book is familiarizing the reader with OrCAD SDT, a thorough understanding of modern schematic drafting practices is a prerequisite for effectively utilizing the software. OrCAD SDT enforces a degree of discipline through concepts such as the organization of multiple sheet schematics into a structured hierarchy and the use of standardized libraries of parts. In addition, OrCAD SDT has certain unique constructs, including invisible power and ground pins and module ports (internal connections between sheets) that are not commonly found in traditional manually drafted schematics. These concepts and constructs must be considered and adhered to in order to obtain all the benefits of schematic capture.

The assumption is made that the reader has some knowledge of schematic drafting and the use of PC systems. The orientation of this book is towards automotive, computer, and industrial control electronics. Most of the examples in this and subsequent chapters are taken from real world applications.

Using Electronic Symbols

Electronic schematics consist of symbols that represent the individual electronic parts used in the circuit. These symbols are interconnected with lines that represent the actual electrical connections. Figure 1-1 shows symbols for the most common of all electronic parts, the resistor. On a typical schematic, each symbol represents an individual part. The symbols are annotated with text. The basic schematic symbols are highly standardized, since the most common parts such as resistors and capacitors have been in use for almost a century.

During the last two decades there has been an ever accelerating proliferation of complex integrated circuit devices. An early attempt was made by the IEEE (Institute of Electrical and Electronics Engineers) to standardize the representation of these parts with a complex new symbology. This approach was feasible with decoders, counters, and bus-oriented devices such as latches and drivers. The IEEE symbols did not keep pace with the advent of VLSI devices such as communications controllers, microcontrollers, programmable logic, and other devices that sometimes have hundreds of pins. OrCAD SDT provides support for IEEE symbols, but few companies still use them. For the most part, they have been forgotten.

Reference Designators

Each symbol is annotated with text that includes a reference designator, for example R1 or R2, and a description of the part. In the case of R1 on Figure 1-1, the description consists of the value (1.0K or 1000 ohms) and wattage rating (.25 watts). Other descriptive text might include the tolerance (1% for R2), a voltage rating, or a manufacturer's part number.

It is very important to clearly understand the importance of the reference designator and the rules for assigning reference designators. An alphanumeric reference designator is used to uniquely identify each part. A given circuit might have ten 1.0K resistors used in different locations. Each of these resistors is given a unique reference designator, for example, R1, R5, and R7. In addition to the schematic, the reference designators also appear on the printed circuit board legend silkscreen, assembly drawing, and bill of materials. Manufacturing uses the reference designators to determine where to stuff parts on the board. Field service uses them to identify and replace failed parts.

Standards have evolved for assigning reference designators. A reference designator consists of an alphabetic prefix and a numeric suffix. Each class of electronic parts has a one or two letter prefix. Most companies use the ANSI (American National Standards Institute) reference designator prefixes with minor modifications as given in Table 1-1. The numeric suffix is numbered starting from one for each class of part, for example, C1, C2, C3, R1, R2, U1, and U2. With manual drafting, the convention is to number each class starting at the upper left-hand corner and then going from left to right in rows from top to bottom. OrCAD SDT can automatically annotate (assign and number) reference designators. The prefix is predefined in the part library, which is described in more detail later in this chapter. OrCAD SDT will number parts in each class in the order they are placed into the design. If the design is created with a reasonable flow, automatic annotation produces acceptable results.

Some parts, such as logic ICs, consist of multiple subparts or gates. In this case, common practice is to add an additional alphabetic suffix to the reference designator, starting with the letter A. For example, the four individual gates of a CMOS 4001 quad NOR gate might be designated U5A, U5B, U5C, and U5D.

Table 1-1 Reference Designator Prefixes

A	assembly, subassembly, device or function block which is separable and/or repairable
AT	attenuator, isolator (RF devices)
B	fan, motor

Table 1-1 Reference Designator Prefixes (Cont'd)

BT battery, photocell
CB circuit breaker
CP coupler, junction (RF devices)
D diode, any two terminal semiconductor device
DC directional coupler (RF device)
DL delay line
DS alarm, buzzer, visual or audible signaling device
E antenna, any miscellaneous electrical device
F fuse
FL filter
G generator (rotating machine)
H hardware
HY circulator (RF device), hybrid circuit
J receptacle (stationary connector)
JP jumper plug (common usage on computer boards)
K contactor, relay (**CR** often used in industrial electronics)
L inductor, coil (single winding - may have multiple taps)
LS loudspeaker, horn, any audio/ultrasonic output transducer
M meter, clock, strain gage, any miscellaneous instrument
MG motor generator
MK microphone, any audio/ultrasonic input transducer
MP mechanical device, any without electrical connections
P plug (removable connector)
PS power supply
Q transistor, MOSFET, SCR, any three terminal semiconductor
R resistor, any fixed or variable (OrCAD uses **RN** for resistor network)
RT thermistor
RV varistor
S switch, thermostat, thermal cutout
T transformer, including autotransformer with single winding
TB terminal board (obsolete)
TC thermocouple
TP test point
U integrated circuit (use of IC is obsolete), nonrepairable assembly
V electron tube, vacuum/ion device including high power RF
VR obsolete usage for zener diode or voltage regulator
W waveguide, transmission line (RF device)
X socket for lamp or fuse
Y crystal, ceramic resonator, tuning fork device
Z tuned cavity or circuit, other miscellaneous RF networks

Table 1-1 is by no means all-inclusive, some companies and industries use varying practices, and a constant evolution is ongoing. For example, the prefix CR has largely become obsolete and D is now used for most two terminal semiconductor devices, including LEDs. Likewise the prefixes IC and VR are obsolete, with U now being used for integrated circuits including voltage regulators. In some areas of industrial controls, CR is used to refer to relays and contactors and PL is used for plugs.

On a final note, remember that the reference designator gives information only about the class of part (resistor, diode, integrated circuit) and the location of the part on the schematic. The reference designator does not give any information about the electrical parameters of the part.

Part Descriptions

The part description must give concise information about all relevant electrical properties. An appropriate part description depends on the type of part. At first glance, it might appear that many passive components such as resistors and capacitors have been highly standardized and only the part value and tolerance would be required to specify the part. For example, one might assume that all 1000 ohm .25 watt 5% tolerance resistors are readily interchangeable, and therefore the part value and tolerance should be a sufficient description. In fact, most schematics are still drawn with such assumptions. Unfortunately, matters have been complicated by two recent trends: schematic capture as the starting point for the design process and the proliferation of SMT.

When schematic capture is used as the starting point, the bill of materials is generated from the schematic database. To the extent feasible, all relevant information available about a given part should be entered into this database. While this will entail more effort up front, updated bill of materials can be efficiently generated as the design evolves through successive engineering changes. This concept of the schematic as database is in sharp contrast to previous industry practice where bill of materials preparation and schematic drafting were distinct activities, and the schematic contained only a subset of the descriptive information contained in the bill of materials.

SMT is the other factor driving the need for more detailed parts descriptions. Today, a 1000 ohm .25 watt 5% resistor could be axial lead through hole, cylindrical MELF, or EIA 1210 size rectangular chip. In some cases, due to considerations of voltage standoff requirements or pulse power handling capability, a design could have a mix of both standard and SMT devices with the same electrical parameters.

OrCAD SDT makes provision for multiple fields in the part description. Efficient and rigorous use of this capability during the early stages of the design process will greatly reduce errors and manual editing of the bill of materials. Detailed guidelines as to what information should accompany individual classes of parts are given in the following sections.

First, let's review the units and associated symbols used to describe the values of electrical circuits and parts commonly found on schematics. There is an immediate problem that needs to be addressed. ANSI/IEEE standards call for a mix of upper case and lower case letters and some Greek letters, such as Ω for ohms, which is the unit for resistance. This is in direct conflict with drafting convention that only upper case letters appear in drawings and with the limitations of ASCII keyboards and output devices. OrCAD SDT supports lower case letters, however, problems can occur in later postprocessing operations. The use of mixed capitalization is not recommended. Table 1-2 gives multiplier suffixes for use with engineering units, and Table 1-3 gives the most common electrical units and associated symbols used by convention.

Table 1-2 Multiplier Suffixes

UNIT	ANSI SYMBOL	MULTIPLIER
femto	f	10^{-15}
pico	p	10^{-12}
nano	n	10^{-9}
micro	μ (use u)	10^{-6}
milli	m	10^{-3}
kilo	k	10^{3}
mega	M	10^{6}
giga	G	10^{9}

Micro and milli suffixes appear to be a problem if only upper case letters are used. The situation is not as bad as it appears because there are no common devices where both these suffixes are likely to occur. Resistors are usually in the range of .01 ohm to 22 megohm. By convention, when M is used with resistance values, it always stands for megohm (for example, 10M is 10 megohm).

Inductors are usually in the .1 microhenry to 10 henry range, with millihenry values quite common. By convention, MH stands for millihenry.

Table 1-3 Engineering Units

UNIT	CONVENTIONAL SYMBOL
Capacitance	
picofarad	PF
nanofarad	NF
microfarad	UF or no symbol (MFD is archaic)
Inductance	
microhenry	UH
millihenry	MH
henry	H
Resistance	
milliohm	write out in decimal, for example .001
ohm	no unit symbol (R used in Europe)
kilohm	K
megohm	M
Electrical Units	
microampere	UA
milliampere	MA
ampere	A
microvolt	UV
millivolt	MV
volt	V
kilovolt	KV
milliwatt	MW
watt	W
kilowatt	KW
Mechanical Units	
microsecond	US or USEC
millisecond	MS or MSEC
second	SEC
minute	MIN
hour	HR
mil (.001 inch)	MIL
inch	IN
foot	FT
centimeter (.01 meter)	CM
meter	M
ounce	OZ
pound	LB
gram	GM
kilogram	KG

While on the subject of units, let's briefly discuss numbering. Preferred practice with schematic capture is to use decimal values rather than fractions. Use .25 watts rather than 1/4 watt. Unlike practice on some mechanical drawings, leading zeros are not used in front of decimal points. Postprocessing routines may encounter difficulties unless these considerations are observed.

Dropping the units symbol is accepted practice with resistors. The appearance of ohms or the proper Greek symbol (Ω) is now rare. In Europe, the letter R is sometimes used both as a unit symbol and decimal point place holder. A 4R7 resistor is 4.7 ohms. Good practice is to add a text note to the schematic indicating that all resistance values are in ohms unless otherwise specified.

Common film and electrolytic capacitors are in the microfarad range. The use of the symbol UF is common. The usage of MFD has become archaic. The trend is to entirely drop the units symbol for microfarad range parts. Again, good practice would dictate a text note to this effect.

OrCAD Symbols for Electronic Parts

This section provides an overview of some of the most widely used types of schematic symbols available in the OrCAD SDT libraries and includes background information on appropriate part descriptions. Figure 1-1 shows OrCAD SDT symbols for the ubiquitous resistor - the most common of all electronic parts. Two styles are used for fixed resistors. R1 is the style used in Europe and industrial controls in North America. R2 is the more traditional style. The R1 style more closely represents modern film resistors, whereas the R2

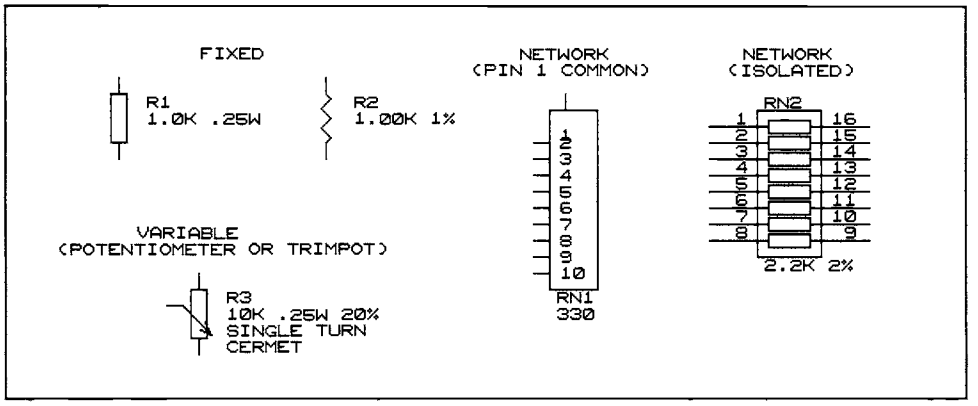


Figure 1-1 Resistor Symbols

style was derived from the wirewound construction of resistors dating back to the turn of the century.

The description of a discrete resistor such as R1 or R2 should include the part value in ohms, wattage, tolerance, temperature coefficient for precision resistors, material, and package size. Materials include wirewound (usually high wattage power resistors), carbon composition, carbon film, and metal film. Carbon composition resistors have almost been entirely supplanted by low cost carbon film types that can easily be manufactured in tighter five percent tolerance. In fact, they are so widely used that one can safely assume, if no other information is provided, that a fractional watt five percent resistor is carbon film. However, relying on such assumptions is not good drafting practice.

Axial lead resistor sizes are well standardized for fractional watt parts, and it is usually not necessary to specify additional package size information when the wattage rating is given. Schematics often have notes such as, ".25W RN55." The RN55 is an old MIL-SPEC package size designation commonly used for one percent metal film resistors. Years ago, .25W 1% metal film resistors were the size of .5W carbon resistors (MIL-SPEC RN60 package size) and RN55 parts were only rated .125W. Better processing techniques have raised the wattage rating of RN55 size parts. The ".25W RN55" statement precludes the use of older style parts that may be too large.

Resistors with wattage ratings greater than 1W do not have well-standardized package sizes. Good practice dictates that you include a manufacturer's part number, series, or type designation in the description or in a separate text note. Surface mount chip resistors have standardized package sizes, but a given wattage rating may be available in several sizes. For example, .06W resistors come in both 0603 and the new microscopic 0402 chip sizes. Again, good practice dictates that you include the size designation along with the wattage rating.

RN1 and RN2 are resistor networks. Most resistor networks with less than 14 pins are understood to be SIP (single inline package) devices and those with 14 or 16 pins to be DIP (dual inline package) devices. Construction and voltage ratings vary, so adding the manufacturer's part number to the description is usually a good idea. Variable resistors, such as R3, can be either small trimpots used to calibrate analog functions or panel mounted potentiometers (pots) for user adjustments. There are many different styles, materials, shaft configurations, and pin arrangements, therefore, a specific part number should be included in the description or in a separate text note.

Much of the information about the resistors on a given schematic is redundant and can be summarized in a few text notes. The use of text notes eliminates unnecessary clutter and makes the schematic more readable.

For example, the notes might state:

1. All resistors .25W 5% carbon film unless otherwise specified.
2. All 1% resistors .25W RN55 size metal film with 50 ppm temperature coefficient unless otherwise specified.
3. All 5W resistors Clarostat series VC5E or equivalent wirewound.

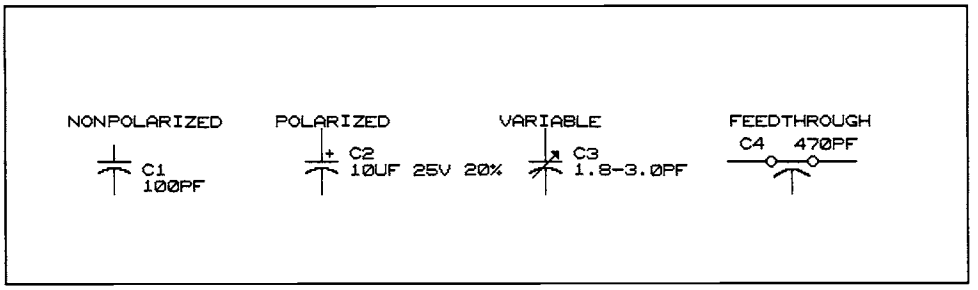


Figure 1-2 Capacitor Symbols

Figure 1-2 shows capacitors. The unit for capacitance is the farad (F), but most capacitors are in the microfarad or picofarad range, so the units UF and PF are used. OrCAD capacitor symbols are shown above. By convention, the capacitor symbol is always oriented towards ground or the lowest DC voltage. For circuits where the DC voltage is not fixed or known, nonpolarized capacitors are drawn with the convex side facing right or down. Additional information required to describe a capacitor includes the voltage rating, tolerance, and construction. Fixed value capacitors can be nonpolarized and polarized. Capacitors are classified according to the electrode and dielectric materials used in their construction. Other than surface mount devices, where a standard size code can easily be added to the part description, there are no real standards for capacitor sizes and the manufacturer's part number should be included in the description.

Nonpolarized types include ceramic, silver mica, and various metal foil-film and metallized films such as polystyrene, polyester, polypropylene, and polycarbonate dielectrics. Older types using oil or wax impregnated paper dielectrics are obsolete. Polarized capacitors include tantalum and aluminum electrolytic types.

Ceramic capacitors have two or three character alphanumeric designations for the dielectric properties that determine initial tolerance and temperature

characteristics. Ceramic capacitors used to bypass power at integrated circuits are usually .01UF to .1UF Z5U +80 -20% types. Oscillator, RF, and timing circuits often require more precise and stable ceramic NPO or COG 5% parts with values ranging up to about 1000 PF. Coupling and filtering applications commonly use ceramic Y5P and X7R 10% types ranging up to about .47 UF. It is not uncommon to find circuits where .1UF Z5U types are used for bypass and identical value .1UF X7R types for timing and filtering. The author can recall one incident in which an improper schematic description resulted in the inadvertent substitution of a Z5U ceramic capacitor into a critical circuit that subsequently malfunctioned.

Film capacitors using metal foil or metallized film are used for specialized applications that require tight tolerance, high voltage or pulse power handling capacity, or very low leakage. A detailed discussion of the design choices and tradeoffs for the various film capacitors is beyond the scope of this book. However, any incorrect substitutions in high voltage or pulse power applications can have disastrous and life threatening consequences. In addition, parts that have regulatory agency (UL, CSA, VDE) approvals are required for many antenna coupling and power supply across-the-line applications. The author suggests that a specific manufacturer's part number be given for such critical parts.

Similar considerations apply to polarized capacitors, especially the aluminum electrolytics. High frequency switching power supplies using aluminum electrolytic capacitors require that the parts be characterized for load life at high temperature, internal resistance (ESR), and ripple current handling capability. Inclusion of a specific manufacturer's part number is highly recommended. A final note of warning on capacitors has to do with polarity. Reverse polarity on a polarized capacitor can cause fire or explosion. Errors can easily occur when general rules of schematic flow (discussed later in this chapter) are not followed and polarized capacitors are drawn with varying orientations on the same sheet.

Inductive devices are found in RF circuits and power supplies. The unit for inductance is the henry (H), but as with capacitors, practical inductors have much smaller values. Inductors for RF circuits are usually air core or ferrite core (drawn same as iron core) in the microhenry (UH) range; those for power supplies can range up to hundreds of millihenry (MH) and are usually ferrite or iron core. The OrCAD part library has a very limited number of transformer symbols. Most power transformers have more than two windings or multiple taps on windings, requiring the user to create a custom part with the library editor.

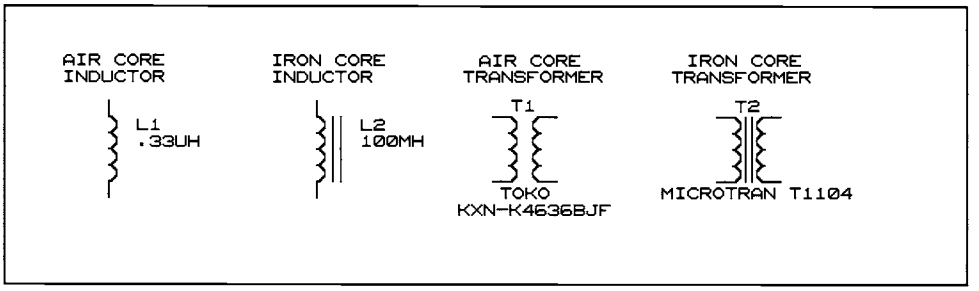


Figure 1-3 Inductor and Transformer Symbols

Almost all applications of inductors and transformers, such as those shown in Figure 1-3, involve consideration of a complex set of electrical parameters (Q factor, leakage inductance, winding resistance and capacitance, and core magnetic properties). A basic description of the device in terms of inductance for inductors or turns ratio, impedance ratio, or voltage levels for transformers is helpful for testing and troubleshooting purposes. A manufacturer's part number is required to specify the device, since there are no standards or generic parts.

Successful postprocessing of the OrCAD database for generation of a PCB design netlist (list of parts and interconnections), requires careful attention to pin numbering. When a custom transformer or other device is created with the OrCAD library editor, the OrCAD pin numbers must correspond to those used in the PCB design.

OrCAD has only a limited number of switches and relays in the part library. The most common are shown in Figure 1-4. For anything other than the most generic devices, the user will have to create a custom part. Even if it appears that one of the existing parts in the library can be used, careful attention must be paid to pin numbers if a netlist is going to be generated for subsequent PCB design. The manufacturer's part number should be included in the description of any part for which no industry standards exist. Good practice also dictates adding text to label switch functions and positions.

OrCAD supplies a rather unique but very graphically descriptive symbol for DIP switches. If text labels are added for on and off positions and the individual switch functions, this symbol makes board setup and troubleshooting more intuitive than individual switch sections scattered throughout a sheet.

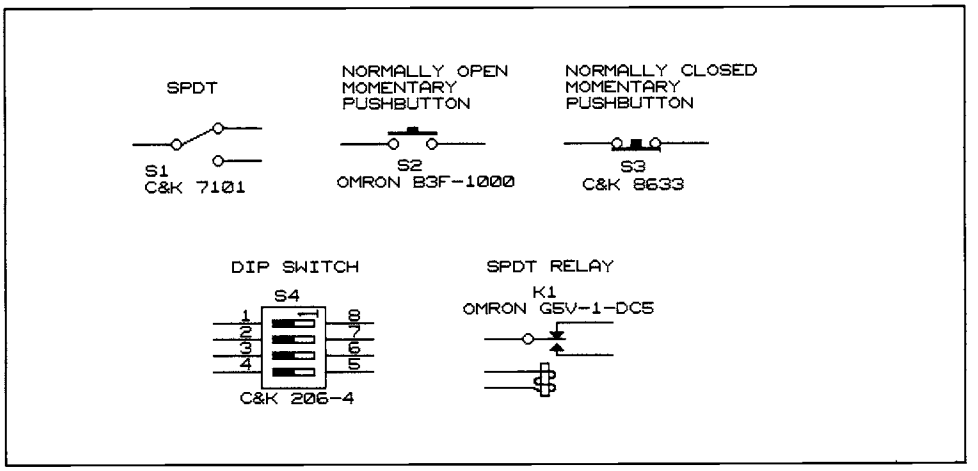


Figure 1-4 Switch and Relay Symbols

The most common semiconductor diodes are shown in Figure 1-5. Optoelectronic devices (LEDs and photodiodes) are covered further on in this section. An industry standard numbering and part registration system was started in the 1950s using 1N prefixes for two terminal diode devices and 2N prefixes for three terminal transistor devices. In theory, all devices with the same number, for example 1N4148 diodes or 2N4401 transistors, should be fully interchangeable. In general this holds true for low voltage, low frequency, or low power parts manufactured using mature technology. For high voltage, high power, or high speed devices, there are usually significant differences in electrical parameters between vendors' parts. Good practice is to specify at least one qualified vendor.

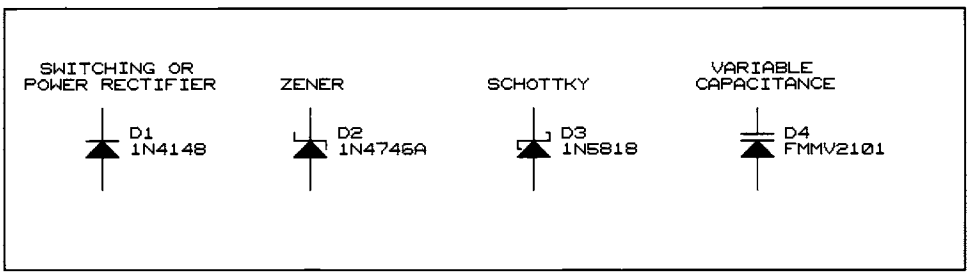


Figure 1-5 Diode Symbols

In recent years, registered numbers have given way to proprietary vendor part numbers, for example the FMMV2101 varactor (variable capacitance diode) shown above. Purchasing agents are bedeviled when looking at bills of materials calling out such devices. The vendor name should always be included.

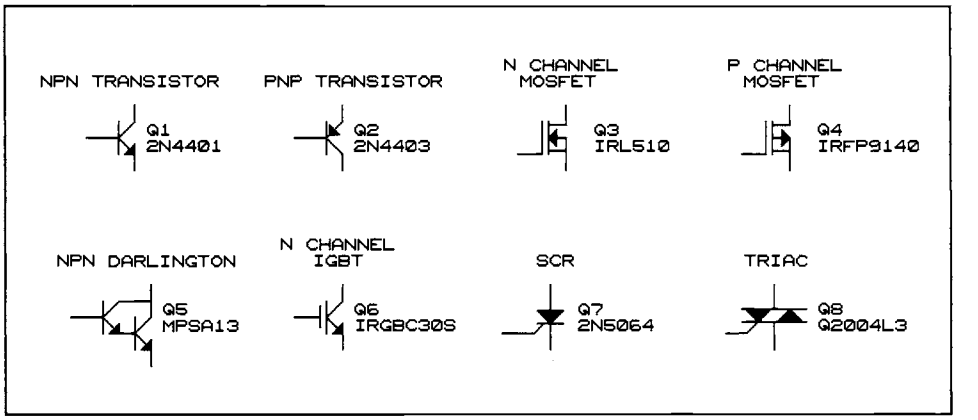


Figure 1-6 Transistor Symbols

Figure 1-6 shows the most frequently used transistors and other 3 lead semiconductor devices as they appear in the OrCAD library. Note that all of the devices in Figures 1-5 and 1-6 appear without any outer circle. Usage of an outer circle around semiconductor devices is considered archaic. Registered transistors and SCRs start with 2N for U.S. and 2S for Japanese devices, some older MOSFET devices start with 3N. Any part number starting with a letter is usually a part that originated with a particular vendor. The description guidelines suggested for diodes in the preceding section are also applicable to transistors and other 3 lead semiconductors.

A recent development is the introduction of so-called smart power devices. This category includes transistors, MOSFETs, and IGBTs with on-chip features such as current limiting, short circuit protection, gate or output voltage clamping, and over temperature protection. In a true sense these devices are really 3 lead integrated circuits. Accepted practice has been to use the basic symbol that most closely represents that output device and simply ignore the presence of the other circuit elements. There are pro and con arguments for this approach, but the author suggests that if it is done, an explanatory text note should be included. This note should include the maximum current, clamping voltage, or other applicable parameters.

Devices in Figure 1-6 are shown in their preferred orientation, with current flowing from top to bottom. If at all possible, the schematic should be drawn with the devices in this orientation. The resulting schematic will be easier to understand.

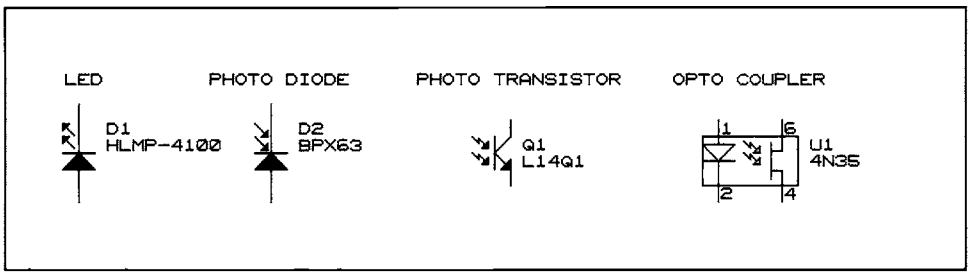


Figure 1-7 Optoelectronic Symbols

Considerations discussed above for diodes and transistors apply to the optoelectronic devices shown in Figure 1-7. LEDs, photo diodes and photo transistors have an optical function, such as a power on indicator or an IR (infrared) communications link. Good practice is to include a text label explaining the intended function.

The term analog or linear refers to circuitry with signals that can vary over a range of voltages. Analog integrated circuits, such as the examples shown in Figure 1-8, require certain unique considerations when drawn on a schematic. OrCAD comes with several analog part libraries, so a very wide range of parts is available. A triangular symbol has been in use for many years to indicate operational amplifiers (op amps) and comparators. All op amps and comparators use the same symbol, but the pinouts vary. Many op amp and comparator integrated circuits contain multiple devices. OrCAD automatically handles multiple device parts and allows the user to determine which device is being placed (drawn) onto the schematic. OrCAD uses the convention of numbering the individual devices with an alphabetic reference designator suffix starting with the letter A. For example, if a LM324 quad op amp is assigned reference designator U1, OrCAD would label the individual devices U1A, U1B, U1C, and U1D. Power and ground pins appear only on the first device, U1A in this case.

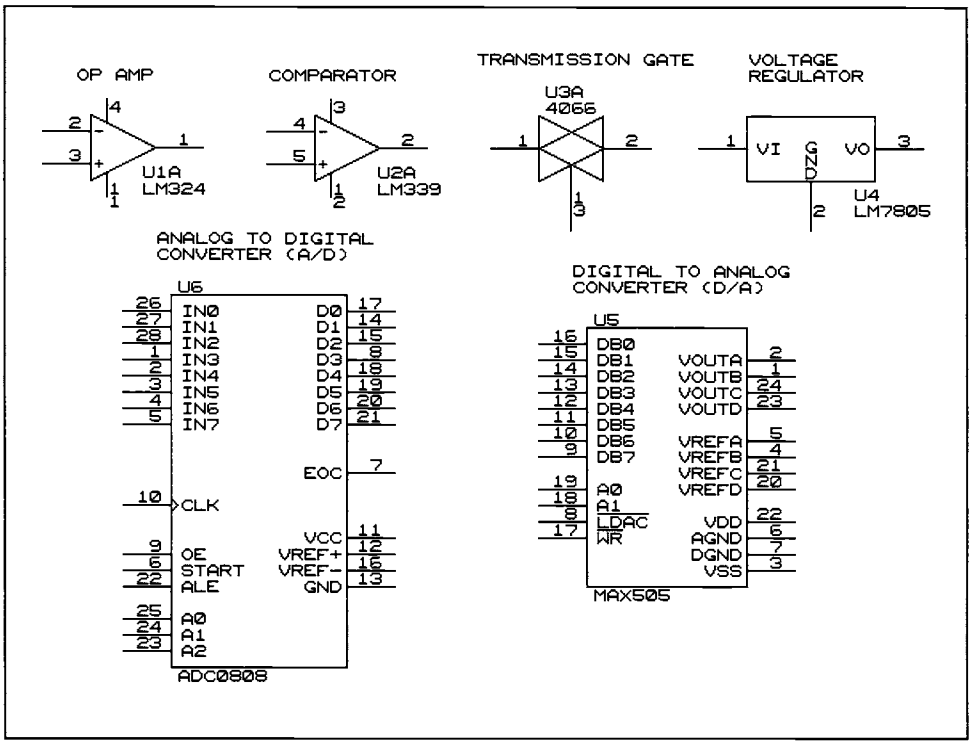


Figure 1-8 Analog Integrated Circuit Symbols

When placing parts on the schematic, OrCAD allows the parts to be rotated in 90 degree increments and mirrored. This high level of flexibility can be a disadvantage in that it allows the user to draw schematics that are difficult to interpret. Proper schematic "flow" will be discussed in detail later in this chapter. Two simple rules that should be followed for schematic flow when drawing op amps and comparators include:

- Draw op amps and comparators with the triangle symbol pointing from left to right as shown in Figure 1-8. This assures compatibility with the requirement that signals should flow from left to right.
- Draw the "-" inverting input on the top. Most electronics engineering textbooks and references show classic op amp feedback networks and comparator circuits with the inverting input terminal on top. Failure to follow this rule can lead to confusion and difficulty understanding circuit operation.

Transmission gates, once a novel part, are now widely used. The author has seen the symbol for the CMOS transmission gate, U3A, in schematics for industrial electronics. The symbol may originate from the similarity of the transmission gate to bidirectional buffers, or it may have evolved from a piping symbol for a valve.

No standard symbols exist for more complex analog integrated circuits. A simple rectangular block with labeled and numbered pins is used. Good practice is to include every pin on the device, even if it is not used or not internally connected. Pins should be arranged with inputs on the left, outputs on the right, power on top, and ground at the bottom. Multiple signals such as data or address should be arranged in order from top to bottom. An older practice is to arrange all pins in the same physical order as on the actual IC. This archaic practice usually results in a messy arrangement of signal interconnections.

Most analog ICs have part numbers that provide an adequate description. Many part numbers have alphabetic prefixes that identify the original vendor, for example CA for RCA (now Harris), LM for National, MAX for Maxim, and TL or Texas Instruments. Often second source vendors will use the same prefix. The LM series is widely sourced. Including a vendor name with the part description would only be a requirement for relatively new or unique parts. Part numbers usually have an alphabetic suffix that identifies the package. The suffix may also identify the temperature range or other electrical parameter such as offset voltage, speed, or power consumption. An LM324N comes in a 14 pin DIP, an LM324M comes in a SOIC (small outline IC) surface mount package, and an LM324AN is a low power DIP version.

Schematics are sometimes drawn as part of a reverse engineering project, and an entire book could easily be written on this subject. Determining the correct part number for a device can be difficult. Date codes can be mistaken for part numbers. The electronics industry uses four digit date codes for most ICs. The first two digits are the year and the last two digits are the week. A part manufactured the 12th week of 1994 would be stamped 9412. Products manufactured in very high volume often have in-house part numbers, even on generic devices. Identifying these devices and finding an equivalent commercial part number can be difficult.

The term digital logic refers to circuitry with signals that are restricted to a limited number of logic states. A detailed description of digital logic functions and theory is beyond the scope of this book. Logic families in widespread use today include TTL and CMOS. Both use standard Boolean logic with 0 and 1 states. Zero is represented by a low voltage (near ground) and 1 by a high voltage (2.5 volts or higher depending on the device). Some devices have an additional high impedance or "off" state. This allows the outputs of multiple devices to be

connected together, such as on a computer data bus, with only one device enabled at a time and the others in the off state.

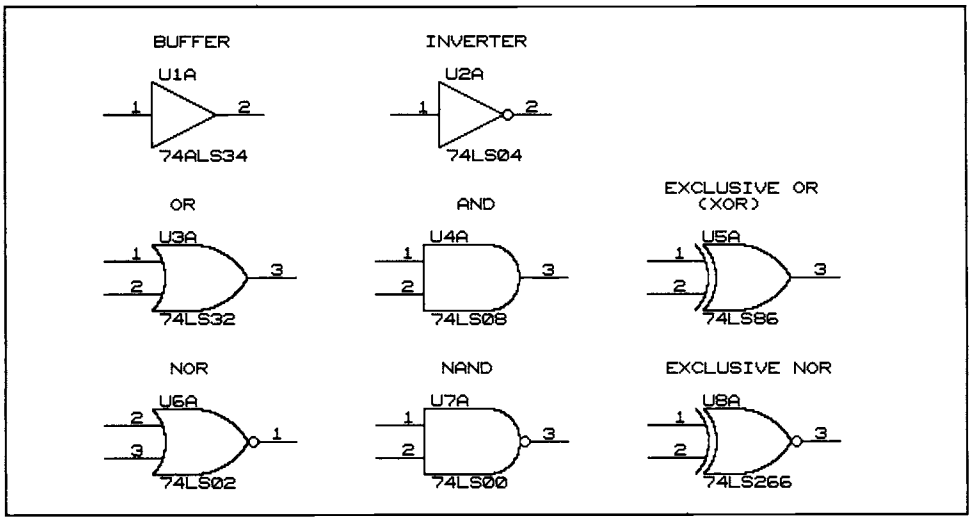


Figure 1-9 Digital Logic Gate Symbols

The four basic digital logic functions include the buffer, OR, AND, and XOR (exclusive OR). Symbols for these logic functions are shown in Figure 1-9. The four basic functions can also have inverted outputs. In this case they are referred to as the inverter (NOT), NOR, NAND, and XNOR (exclusive NOR) and the symbol is drawn with a small circle on the output. The small circle indicates an inverted signal, either input or output. Logic functions, when implemented in electronic circuitry, are referred to as gates. Buffer and inverter gates can only have a single input and output. XOR and XNOR gates, by nature of their logic function can only have two inputs. All other gates can have two or more inputs.

Digital logic ICs may contain from one to six gates. OrCAD handles digital ICs with multiple gates using the same principles as discussed in the previous section for analog ICs with multiple devices. Likewise, a part number description is generally sufficient for digital logic, since most parts are sourced by multiple vendors. TTL is the most widely used logic family. Some TTL part numbers are shown in Figure 1-9, such as the 74LS02 NOR gate. The 74 prefix identifies the part as commercial temperature range (0 to 70 degrees C); a 54 prefix is used for military temperature range parts (-55 to 125 degrees C). The LS designates the part as belonging to the low power Schottky logic family. The original TTL logic had no special designator. The part number for the quad NOR gate would have been 7402. Many different families have been introduced, each with their own

two or three letter designator. In fact many of these new families are not even based on transistor technology. CMOS variants, such as a 74HC02, use MOSFET technology with somewhat different logic levels, but retain the popular device functions and pinouts originated with TTL.

OrCAD makes use of the fact that pinouts are the same for a given device function in all TTL logic families. Only a single part definition is used in the TTL library, and the various logic families are listed as optional part names. This is an advantage for the user, because part names can be edited. If the requirement is for a new 3.3V logic device such as a Phillips 74LVT244, which does not yet appear in an OrCAD library, one can use a 74LS244 and simply edit the name. This is much more convenient than creating a new part.

Figure 1-10 shows advanced digital integrated circuits. As with advanced analog integrated circuits, these devices are drawn as rectangular blocks with named and numbered signal pins. Note that U2 and U3 both have inverted signal inputs and outputs denoted by a small circle. U1A also has noninverted and inverted outputs. Here the inverted output on pin 2 is denoted by a bar above the signal name. Both the small circle and bar above the signal name conventions are used by OrCAD library parts on a rather arbitrary basis. In some cases, the choice of convention appears to have been based on how the original vendor represented the part in the databook. Another signal convention is the use of a small triangle to represent edge triggered clock inputs, as shown on U1A pin 3 and U2 pin 15.

U5 and U6 introduce another special category - programmable memory and microcontroller devices. These devices, and others such as the broad class of programmable logic devices, contain firmware. Firmware is the same as software, except it is more or less permanently programmed into the device. With the advent of electrically erasable memory devices, the distinction between software and firmware has become more blurred, but convention is still to refer to any code loaded into non-volatile memory as firmware.

Whenever devices requiring or utilizing firmware appear on a schematic, it is good practice to add a note, either to the device description or as a separate text note, that gives the firmware name and the earliest compatible version. Why note the earliest compatible version? Today schematics are usually generated at the start of a design project - not as part of the documentation created after the fact. Often earlier revisions of a product will exist. Firmware used with earlier board revisions may not be compatible. Firmware seems to go through rapid changes and there are usually multiple firmware versions released before the board hardware is updated. Noting the earliest compatible version can help prevent mistakes during manufacturing, final testing, and field servicing.

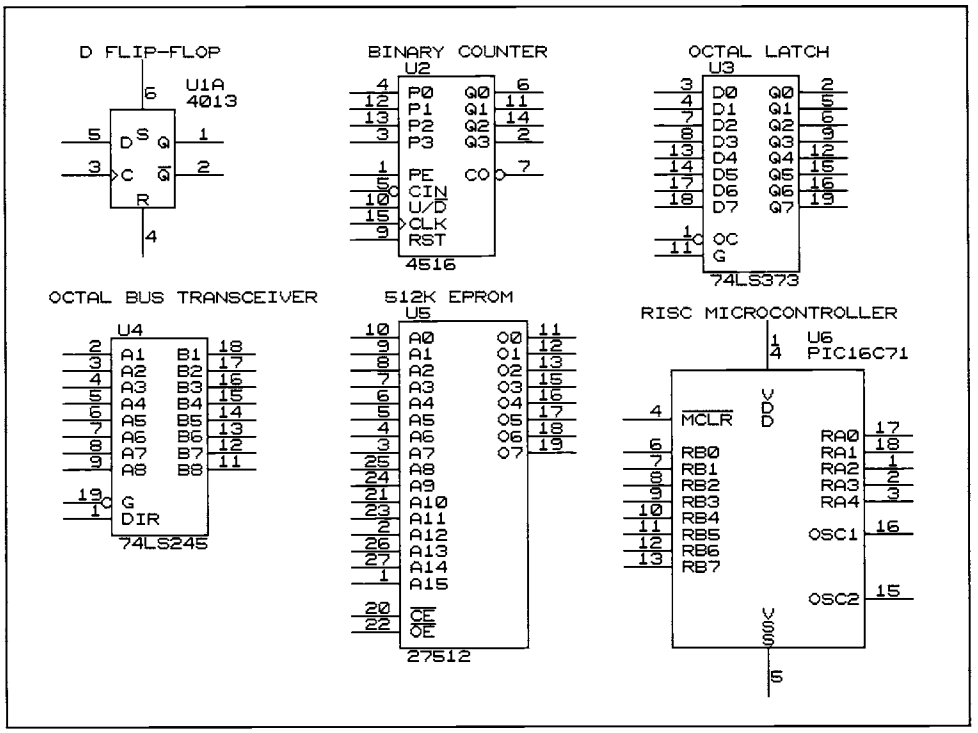


Figure 1-10 Advanced Digital Logic Symbols

Memory devices usually have a read or write access time, which is a very critical parameter in high speed computer systems. Often the part number has a dash suffix, such as -150, where the number identifies the access time in nanoseconds. If this parameter is critical and the device is available in multiple speed grades, this information must be included on the schematic. A similar consideration applies to programmable devices, where the programming voltage often varies between different vendors. If relevant, programming voltage information should also be included.

Figures 1-11 and 1-12 complete the overview of circuit symbols supplied with OrCAD. There are some inconsistencies between the OrCAD library parts and ANSI recommended reference designators. The ANSI designator for a microphone is MK, yet OrCAD uses X. Reference designators can be edited once the part has been placed onto the schematic, so this is not a serious problem. The reasoning for the letter A on the motor symbol is not clear.

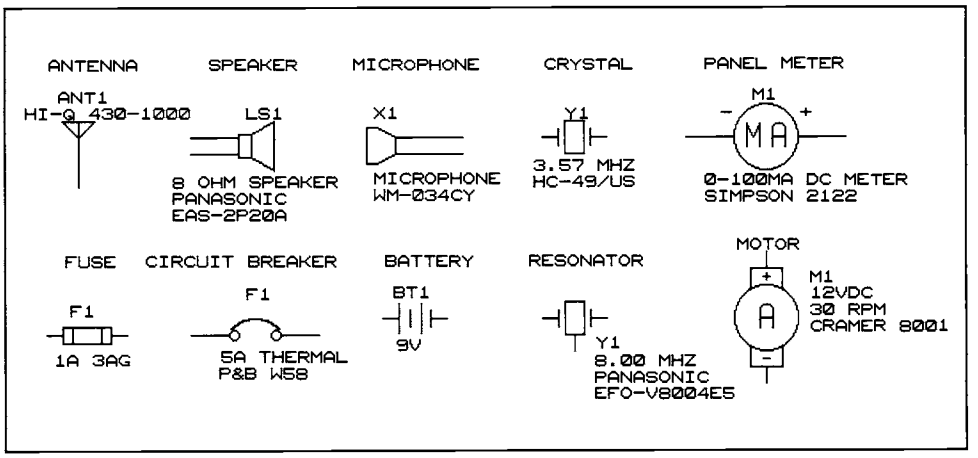


Figure 1-11 Miscellaneous Symbols

Batteries are represented by the symbol shown for BT1. Again there is a minor inconsistency. BT1, as drawn, shows a two cell battery. To be technically correct, a 9V alkaline or carbon-zinc battery should be drawn showing six 1.5V cells. In today's environment of complex computer systems, such details are often glossed over. As long as the battery is correctly described, using the OrCAD-supplied symbol shown above for any multicell battery, there should not be any confusion.

Some of the most commonly used connector symbols are shown in Figure 1-12. By convention, any fixed connector attached to a panel or motherboard is referred to as a jack and uses reference designator J. A removable connector, including the card edge connector on a daughterboard or PC bus expansion board is referred to as a plug and uses reference designator P or PL. With the advent of user configured computer expansion boards, the jumper or jumper plug has become ubiquitous. Jumpers are typically incorporated onto a circuit board as paired pads on .1 inch centers. For user configurable board options, a header can be installed and a removable jumper provided to make the connection.

Jumpers used for manufacturing options or calibration functions are often normally closed with a small trace between the pads. The trace can be cut with an Xacto knife to open the jumper. If required, a wire can be soldered between the pads to re-establish the jumper connection. It is very important to clearly describe the function of jumpers, and whether the function is asserted with the jumper open or closed. For complex functions set via multiple jumpers, a table listing is good practice. Normally closed jumpers should be shown with a line drawn between the circles, similar to a normally closed switch

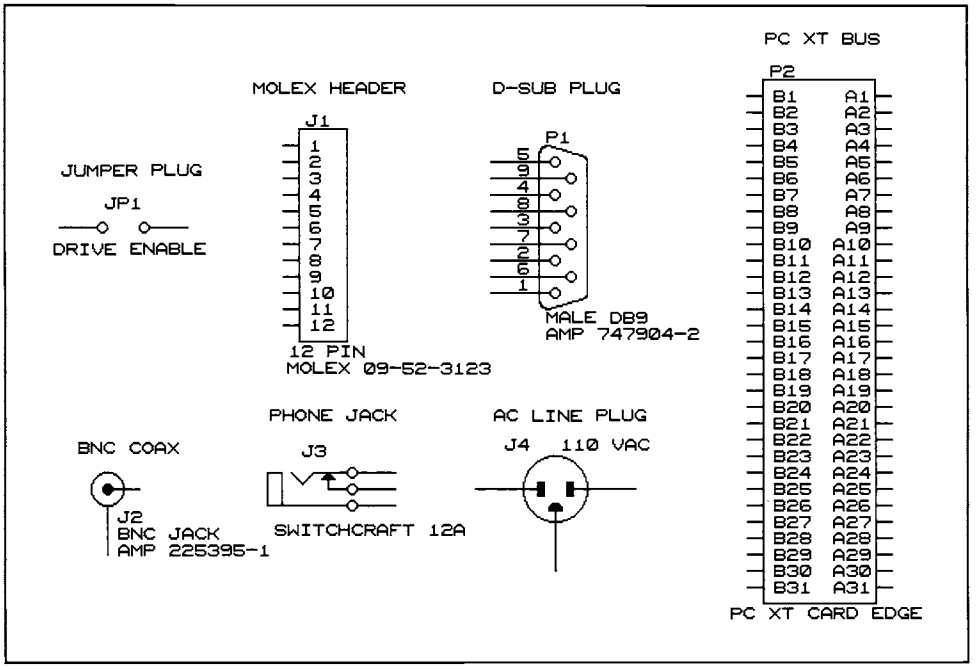


Figure 1-12 Connector Symbols

Good practice is to describe the physical configuration of all connectors, give a vendor name and part number if applicable, and add a text note that describes the function served or the connecting device.

Special Schematic Objects

OrCAD and other schematic capture programs use certain special objects to handle signal and power flow. These objects are shown in Figures 1-13 and 1-14. The most basic object is a wire interconnecting part pins. Placing a wire between two pins causes a line to be drawn. At the same time, the connection is stored in the schematic database. A connection between two or more pins is referred to as a net. Pins that are not part of any net are referred to as unconnected. After the schematic design is completed, the connectivity information in the schematic database can be postprocessed into a netlist. The netlist is used to transfer the connectivity information into other programs and systems, such as what might be used for PCB design.

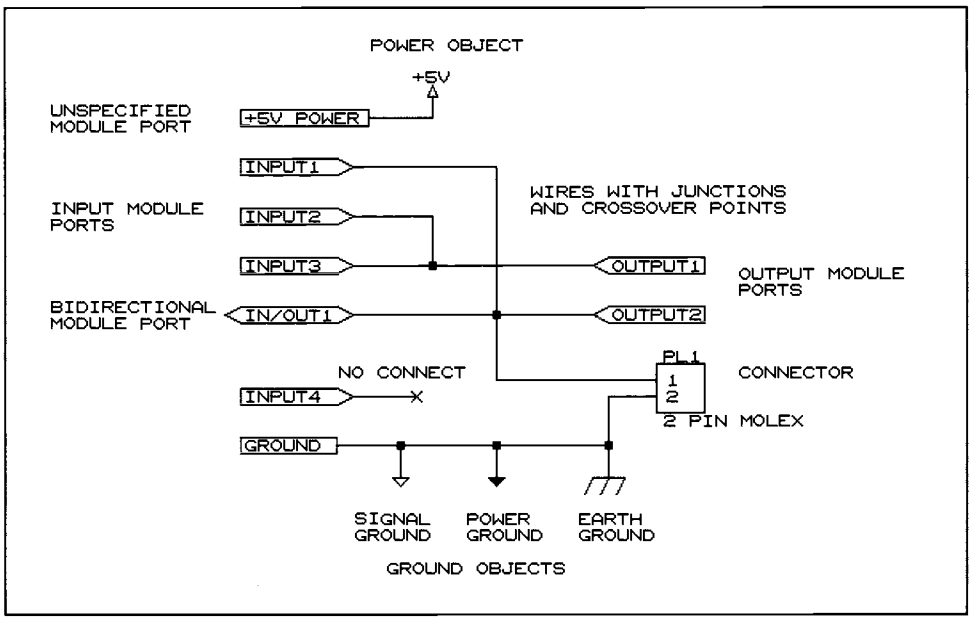


Figure 1-13 OrCAD Signal and Power Objects

Wires that cross one another are not connected or considered part of the same net unless a junction object is placed at the crossover point. OrCAD junctions appear as small, filled in squares. Manual drafting techniques often use a small round dot to represent a junction and a break in the line to represent a crossover point where no connection exists. In OrCAD, wires must run continuously from pin to pin. The only exceptions are when a signal label or module port is associated with the wire.

Power objects are used to represent connections to power and ground. OrCAD has three different style ground objects: signal ground, power ground, and earth ground. These three ground objects are electrically identical. They are *not* isolated from one another because they all share the same signal name, GND.

OrCAD provides an undefined power object. This can be drawn in several styles and rotated in 90 degree increments. The arrowhead style pointing up is most widely used and recommended. The undefined power object can be given any name when it is placed on the schematic. Pins and wires connected to power objects with the same signal name become part of the same net. The name of the net is simply the name of the power object. For example, all wires tied to +5V power objects become part of the +5V net. In this manner, any number of different power supply levels can be handled. Ground objects are treated just as any other power object.

The concepts of wires, junctions, crossovers, grounds, and power objects are generally familiar to anyone who has had experience in schematic drafting and cause few problems when making the transition from manual drafting to schematic capture. It is important to clearly understand the distinction between module ports and connectors. The module port is a concept somewhat unique to schematic capture and hierarchical schematics. Users new to schematic capture are often confused by module ports and try to use module ports to represent connectors.

The most important thing to remember about module ports is that they are not connectors and must never be used in place of connectors. A module port does not represent any physical part. Module ports are used with multiple sheet schematics. Their only purpose is to route signals between sheets.

OrCAD allows four types of module ports: input, output, bidirectional, and unspecified. The sense or direction of a module port refers to the sheet on which it occurs. For example, the input ports on Figure 1-13 represent signals coming from other sheets. Bidirectional ports are used for data buses where signals can flow in both directions. Unspecified ports are used for power and reference voltage connections between sheets. Special rules apply to unspecified ports.

Recommended practice is to place input ports on the left and output ports on the right side of the sheet. Placement of bidirectional and unspecified ports is arbitrary. The standard styles for the various ports are shown in Figure 1-13. However, OrCAD allows any style to be used for any type of port by editing the port after it has been placed into the schematic. Do not use this dubious feature. Stay with the standard styles and avoid confusion.

As with power objects, any wire tied to a module port with the same name will become part of a net with that same name. The style or type of module port is irrelevant, only the name matters.

As mentioned above, special rules apply to unspecified module ports. The term unspecified refers to signal flow. Unspecified ports are generally used only for analog signals and power supply connections. Signal flow direction is assumed to be unknown or inapplicable. The most common usage of an unspecified port is to isolate power and ground for a particular circuit block. The need for isolated power often arises in systems with logic running at multiple voltage levels. Examples include core logic for a Pentium processor at 3.3V interfaced to an ISA bus at 5V or a RAM memory subsystem with battery backup.

For now, just remember that power objects on a given sheet tied to an unspecified module port become part of a net with the name of the port, not with the name of the power object. This may sound very confusing. The rationale will

become apparent as you learn more about concepts such as invisible power pins and work on designs with isolated power supplies in the tutorial exercises.

A no connect object should be placed at pins or wires that are not connected to any other circuit element. While this may seem redundant, it is very important. One of the most useful features of OrCAD is the electrical rules checking routine. This routine has a list of rules, most of which relate to types of connections that are allowed. Violations are flagged to alert the user to possible problem areas and errors. Unconnected pins are flagged as an error unless a no connect object is placed at their end.

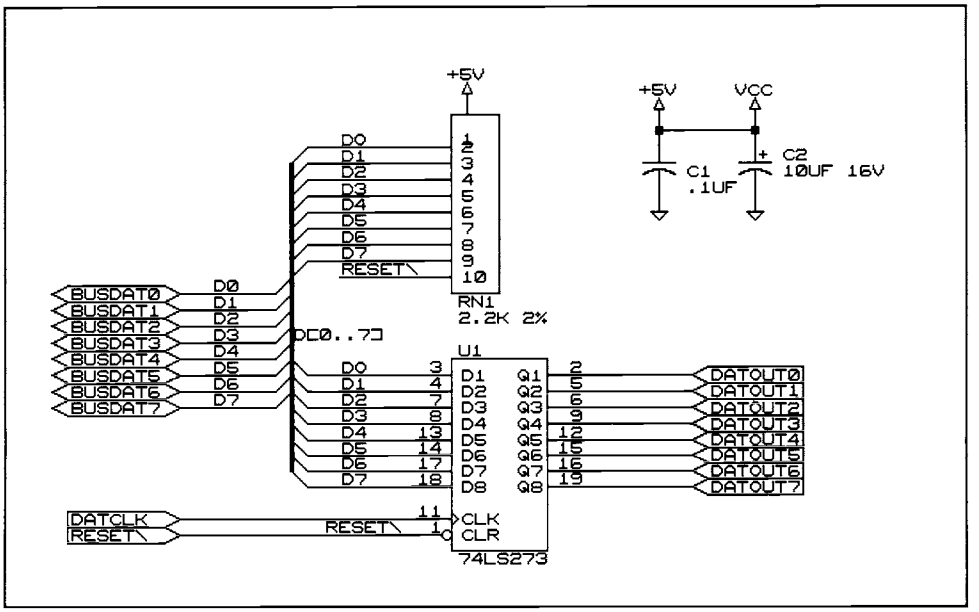


Figure 1-14 Advanced OrCAD Entities

Figure 1-14 introduces two more OrCAD schematic entities: bus structures and labels. The use of bus structures to represent a group of related signals, such as a data or address bus, became widespread practice after microprocessor technology was introduced in the early 1970s. Representing an eight bit data bus or 16 bit address bus with a single broad line is more convenient than drawing out each signal. Besides the obvious time savings in drafting, the signal flow becomes much clearer and easier to understand. With the advent of 32 and 64 bit bus structures, drawing out each signal is no longer an option.

OrCAD allows bus structures with any number of signals. Certain conventions must be followed. Signals enter (or exit) a bus via a 45 degree diagonal bus entry object. Merely drawing a wire to a bus will not result in a connection. A uniform name must be used for all signals on a bus, for example D0, D1...D32 or ADDR16, ADDR17...ADDR24. The name always consists of an alphabetic prefix and a numerical suffix. Corresponding bus labels of the form D[0..32] or ADDR[16..24] must be placed on the bus. Individual signal labels such as D2 or ADDR17 must be placed on wires entering the bus.

OrCAD signal labels are objects. Labels objects can be used not only with buses but wherever it is convenient to identify and connect wires representing the same signal. Figure 1-14 shows the RESET\ signal routed to pin 10 of RN1 by placing labels named RESET\ on the appropriate wires. This results in a neater appearance than if wires were routed from end to end. It is important to note that signal labels can only be placed on wires and that the first character of the label must touch the wire. In Figure 1-14, two RESET\ labels are used. Just placing a RESET\ label on pin 10 of RN1 does not establish a connection back to the module port. The name of a module port is not directly associated with any signal label and different names can be used. On a final note, the preferred convention for inverted signals is to place a backslash (\) at the end of the name. OrCAD provides no means of drawing a bar above label text.

Preferred Schematic Drafting Practices

So far, the discussion of preferred practices has focused on the use of schematic symbols and the content of descriptive information. The subject of schematic flow requires further discussion. Proper flow is critical to assuring readability of the finished schematic. Figure 1-15A shows a schematic with poor flow. Figure 1-15B shows the same circuit redrawn with improved flow.

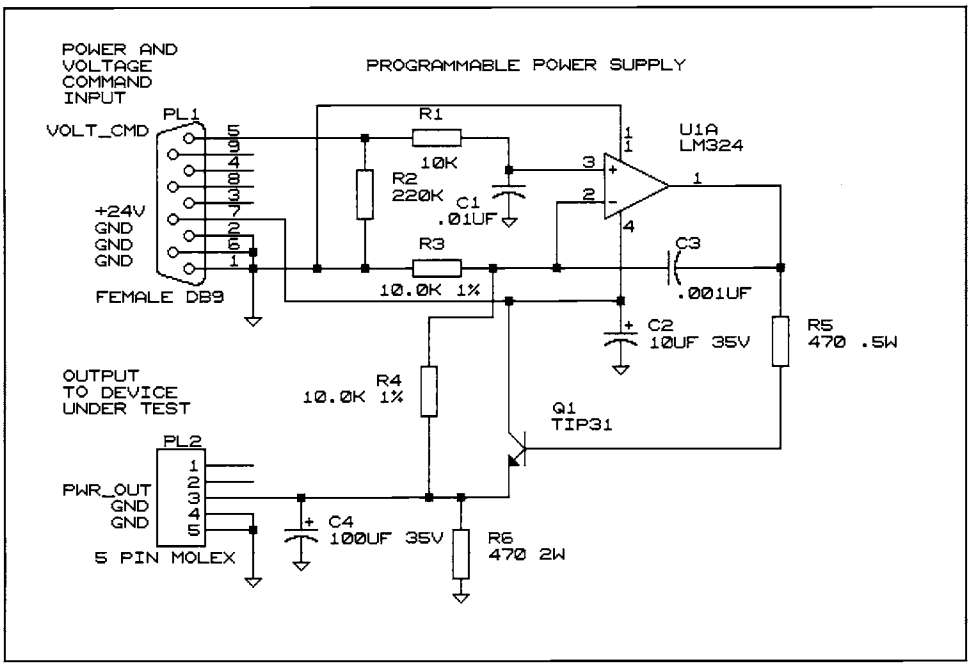


Figure 1-15A Schematic with Poor Flow

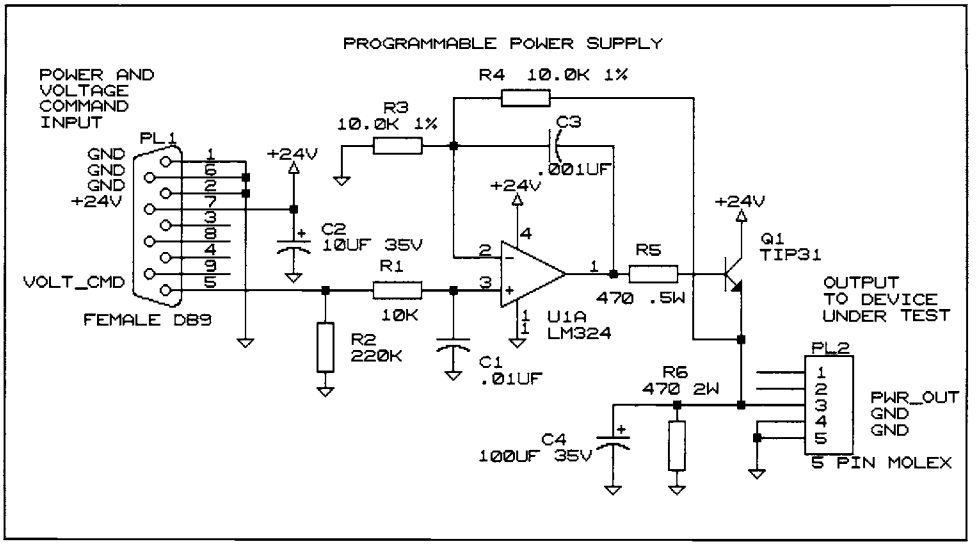


Figure 1-15B Schematic Redrawn with Improved Flow

Schematic Flow

The following list gives drafting recommendations for schematic flow:

- Signals should flow from left to right and in rows from top to bottom, as shown in Figure 1-16. Place connectors according to the predominant signals: input connectors on the left and output connectors on the right.
- Orient devices in accordance with the signal flow principle. Normal orientation is with inputs on the left and outputs on the right, positive power on top, and negative power or ground at the bottom.
- Arrange circuitry so that for each row, voltages increase from bottom to top. Locate ground objects facing down at the bottom of rows. Locate power objects facing up at the top of rows.
- Use bus structures to relieve signal congestion for all data and address signals. Use labels to implicitly join separated wire segments together rather than routing the wires over great lengths.
- Orient op amps and comparators so that their "-" inverting inputs appear at the top.
- Connect device power and ground pins to individual power and ground objects. Do not tie many leads together at a single power or ground object.

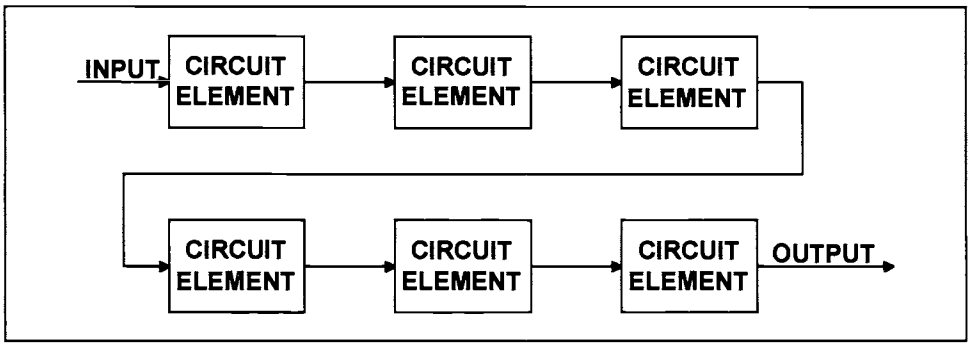


Figure 1-16 Schematic Flow in Rows

Signal Naming Conventions

Certain conventions should be observed for naming signals and power. As previously discussed, a backslash (\) following the signal name represents inverted signals or signals that are active low (logic function enabled or asserted at a zero logic level). Bus signals must be numbered in sequence, such as D0, D1, ...D7. While OrCAD does not impose any limitations, data and address signals are always numbered starting from zero. Signal names containing D, DAT, or DATA are common for data lines and A or ADDR for address lines. Edge triggered signals used to clock counters or flip-flops should contain CLK in the name, such as BUSCLK1. In general, signal names should be descriptive. The underline character (_) can be used to join segments of a signal name for clarity. For example, the signal name A2D_D0 (analog to digital converter data line 0) reads better than A2DD0. Avoid picking arbitrary signal names. If in doubt, construct an abbreviation from the plain language description of the signal.

Designs frequently utilize multiple power supplies and reference voltages. Choose appropriate names for power objects. +5V is more descriptive than VCC. OrCAD library parts have invisible power pins that use vague names such as VSS, VDD, and VCC since the actual voltage levels may vary from design to design. Do not use these types of names for known and well defined power levels!

Unless you are designing battery powered vacuum tube equipment, do not use B+ as a power name. B+ is an archaic term for the "B" or plate supply battery. For battery powered portable equipment, varying battery voltages can be represented by adding BAT to the power name, for example +9VBAT. Reference voltages appear in analog circuits, especially digital to analog and analog to digital converters. Precision reference circuits used to generate a reference voltage are limited to a low current. The naming convention should preclude mistaking a reference for a power supply. A name such as +5.00VREF identifies a precise reference voltage that can be expected to vary less than .01 volt from the nominal +5.00 volt value.

Title Block and Notes

All schematics should include a title block. Besides the schematic graphics, additional text notes are almost unavoidable. OrCAD provides a predefined title block format that is suitable for most organizations. Any amount of text notes can be added to the schematic. In the case of text notes, too much detail is better than not enough. A limitation of OrCAD, which the author hopes will be addressed in future Windows versions, is the inability to draw non-schematic

objects, such as signal waveforms. At present, this can only be accomplished by plotting the schematic as DXF (AutoCAD drawing exchange format) data, loading this data into a CAD drafting program such as AutoCAD, and then adding the appropriate annotations.

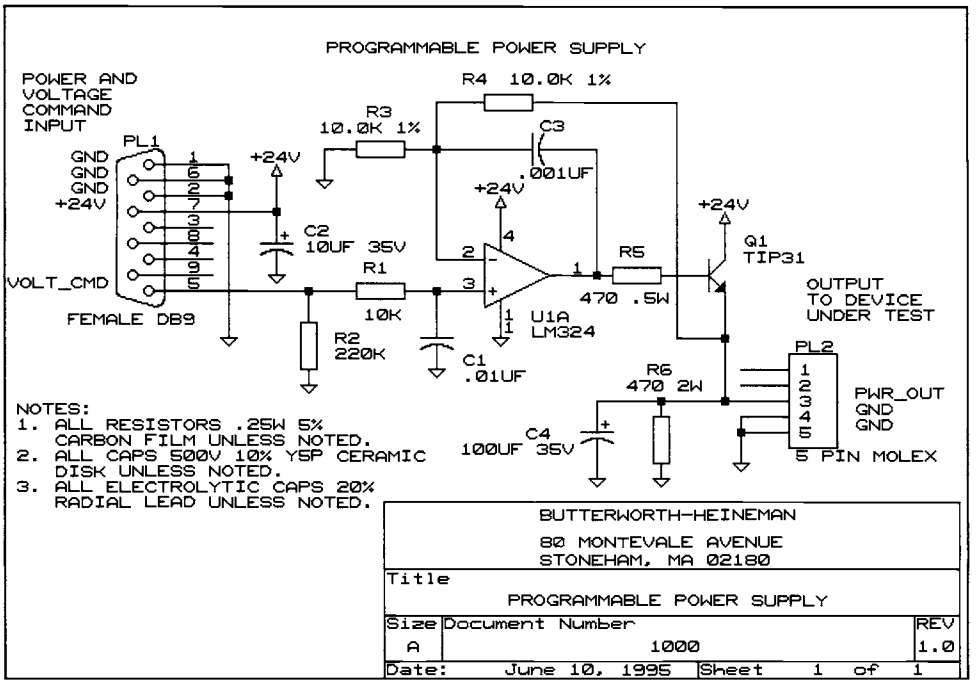


Figure 1-17 Finished Single Sheet Schematic

Figure 1-17 shows the complete version of the earlier schematic redrawn for improved flow. A title block and notes have been added. The title block for a schematic typically contains the following elements:

- **Company name and address**. Self explanatory.
- **Title**. The plain language description of the device or circuit.
- **Size**. Letter designators A-E are used for standard drawing sizes. Most schematics are now laser printed on A size paper (8.5 by 11 inches).
- **Document number**. Most companies track documentation by a document or drawing number.
- **Revision code**. Either a letter or number can be used. Decimal numbers are common in the electronics industry due to the rapid pace of changes.

Development prototypes are given revision codes less than one, such as 0.9 for the final prototype. First production is usually revision 1.0. A major change such as a new circuit board causes the revision to jump to the next integer; for example from 1.5 to 2.0. Minor changes, where only a few component values are modified or jumpers and cuts are made on the board, cause the revision code to increment by .1; for example from 2.0 to 2.1.

- **Date**. Automatically assigned and updated by OrCAD whenever any editing is done.
- **Sheet number and number of sheets**. Self explanatory.

Text notes complete the schematic. Back in the days of manual drafting on paper, editing text notes meant using an eraser. To avoid having to repeatedly erase and rewrite everything whenever a change or addition occurred, notes started at the bottom and ran in sequence going up. With the ease of CAD editing, this practice has become archaic. Notes are now written and numbered in normal sequence from top to bottom. Some of the items that should appear in the text notes include:

- Default values for components, such as resistor wattage rating and tolerance.
- Preferred or approved vendors for critical or unique parts and any other important information not contained in the part descriptions.
- Engineering change descriptions and history, including compatibility of previous revisions.
- Brief calibration and test instructions or a reference to a separate document.
- Data on firmware.

Hierarchical Schematics

In the last decade, the trend for schematics has been away from large C, D, and even E size pen plotted or hand drawn sheets. The large drawing sizes are cumbersome to handle and expensive to reproduce. Products have become very complex - to the point that even several E size sheets may not suffice for the schematic of a new VME bus computer board. Multiple sheet A size schematics are the answer. They can be quickly run off on a laser printer and inexpensively reproduced on any copying machine. The only problem is providing some means of oversight and continuation between sheets. OrCAD solves this problem by using the concept of a hierarchical schematic. An example, as shown in Figure 1-18, best illustrates the concept.

The first sheet provides an overview of the organization or hierarchy of the design. Sheet objects, each representing another sheet, are shown along with the interconnections between them. OrCAD uses the somewhat confusing term "sheet net" for the little arrow-like symbols on the sheet objects. These sheet nets correspond to module ports on the sheet. Just as with module ports, there are input, output, bidirectional, and unspecified type sheet nets. They even take on the same appearance as the corresponding type module ports, except drawn to a smaller scale and filled in. The net name is placed next to the sheet net symbol. The most important point to remember is that a sheet net with the same name must be created for every module port associated with a given sheet. Wires between the sheet nets interconnect the sheets.

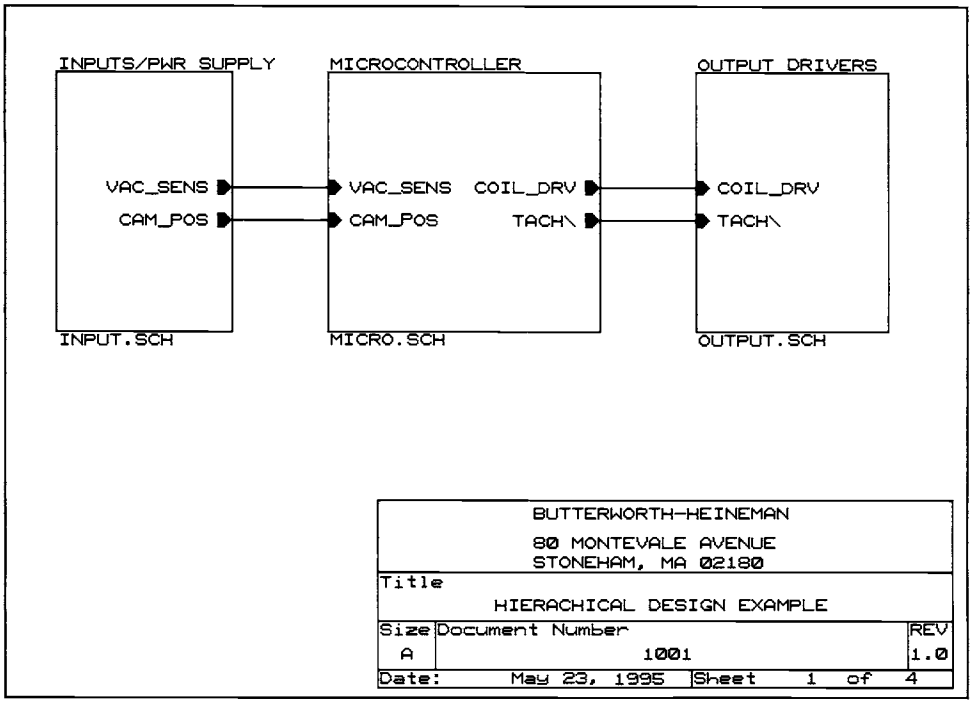


Figure 1-18A Hierarchical Design Example Sheet One

Sheet one of the hierarchical design gives an overview of the major circuit blocks and interconnections. In this context, the terms block and sheet are somewhat interchangeable. Most designers think in terms of circuit blocks. OrCAD uses the terminology sheet. A sheet can represent multiple blocks. It is useful to think of the first sheet as a block diagram of the circuitry. In this example, every sheet net is connected to another sheet net with the same name, such as VAC_SENS from

the input block to VAC_SENS on the microcontroller block. If required, VAC_SENS could be connected to a sheet net with a different name.

There is no limitation on the interconnection of sheet nets. The example in Figure 1-18A shows a single level hierarchy. Sheet one shows three lower level sheets. OrCAD supports multiple level hierarchies. For example, sheet two could contain several circuit blocks each represented by additional sheets. Designs with more than a few levels can become confusing. Two or three levels seems to work out for most practical designs.

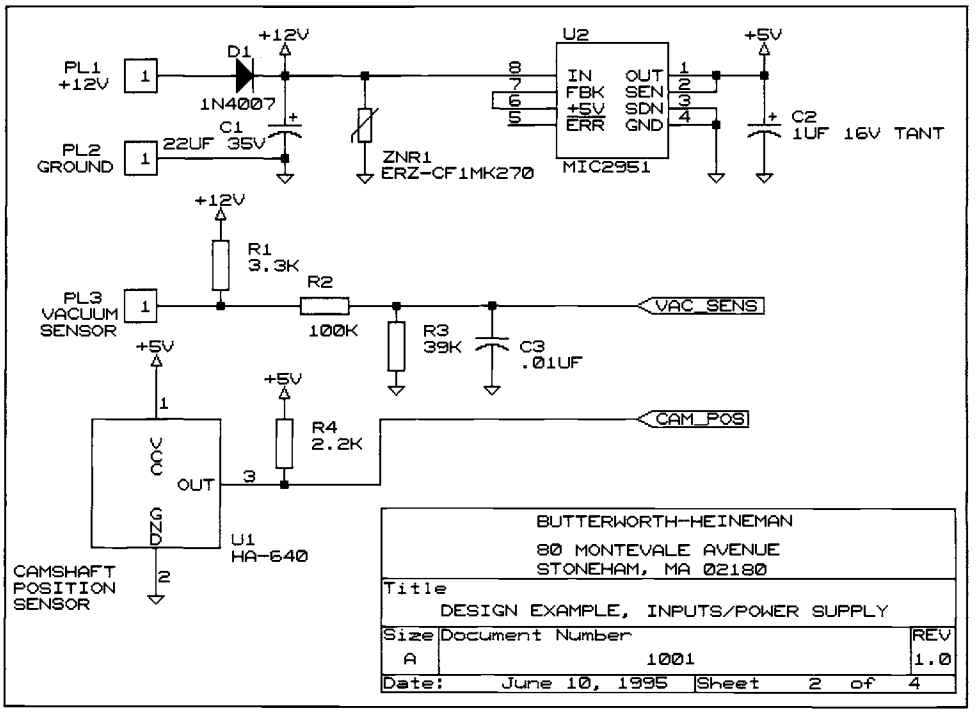


Figure 1-18B Hierarchical Design Example Sheet Two

Power is automatically carried across sheets unless special steps are taken to isolate it. In the example above, +12V power comes in at PL1. The power supply is connected to +12V and +5V power objects. Power is automatically routed to any other sheet that contains +12V or +5V power objects. Ground connections are treated the same. As mentioned before, OrCAD provides three different styles of ground objects (signal, power, and earth). All of these ground objects are tied together in a single ground plane that runs throughout the entire design unless special steps are taken to isolate one section of the circuit.

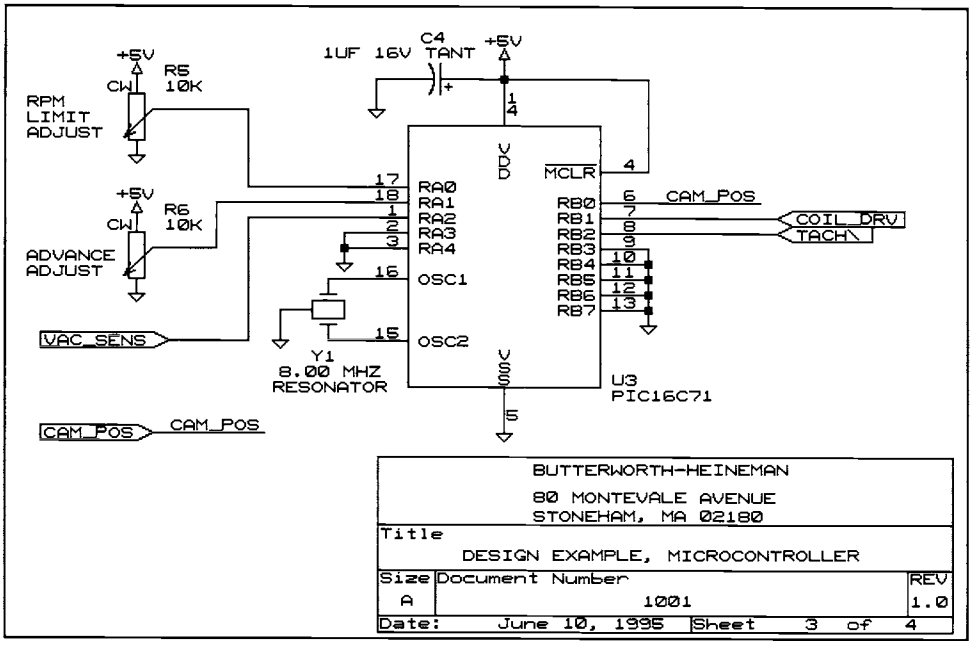


Figure 1-18C Hierarchical Design Example Sheet Three

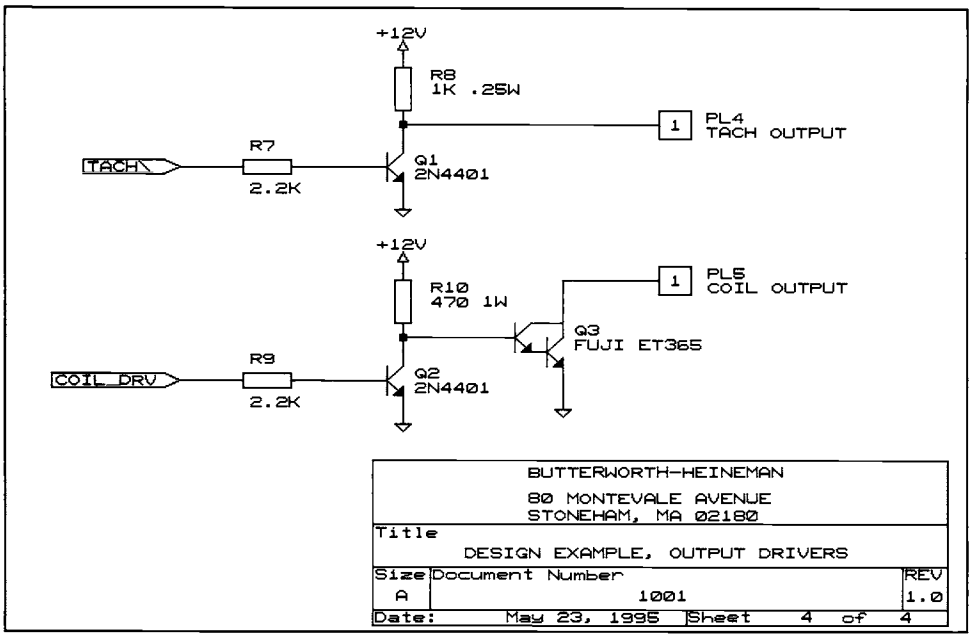


Figure 1-18D Hierarchical Design Example Sheet Four

OrCAD Part Libraries

One of the great strengths of OrCAD is the comprehensive set of part libraries supplied with the program. Basic electronic components such as those shown in Figures 1-1 through 1-7, 1-11, and 1-12 appear in a general purpose "DEVICE" library. Generic "CMOS" and "TTL" libraries include the parts shown in Figures 1-9 and 1-10. OrCAD includes many additional vendor specific libraries for parts from Intel, Maxim, Motorola, National, TI (Texas Instruments), and other large companies. All together, thousands of parts are available.

In addition to the libraries supplied with OrCAD, the user can easily create and edit custom part libraries. Since complex ICs are represented by a rectangular block with numbered and named pins, creating a new part definition and adding it to a custom library is not a difficult or time consuming task.

The user is cautioned that editing of the OrCAD supplied libraries should be avoided. If a standard library part must be modified, the recommended procedure is to export a copy of the part from the original library, import the part into the user's custom library, and then perform the required edits. The reason for not editing the OrCAD libraries is simple. OrCAD provides periodic software updates, which may include library updates. If a library has been edited, no procedure exists for updating the library without losing the previous edits. To avoid headaches, keep edited or newly created parts in a separate custom library.

Invisible Power Pins

Most digital logic integrated circuit parts are defined with invisible power pins. In this context power is understood to mean both the positive power connection(s) and ground. The power pins are normally hidden from view and automatically connected to the appropriate power and ground planes when the part is inserted into the schematic. Figure 1-19 shows the power pins that are normally invisible.

Schematic clarity improves when clutter from power pins is eliminated. The use of invisible power pins originated back in the days before large surface mount device packages were common. Most digital ICs had power and ground on diagonal ends of the package and little guessing was involved. With the larger surface mount devices, this is not always the case. Invisible and thus unknown power pins can cause headaches when it comes to troubleshooting.

Invisible power pins cause additional problems because OrCAD has no standard naming convention for these pins. TTL circuits use the names VCC and GND. CMOS circuits use VSS and VDD. There is no way to tell which names are used other than examining the part definition with the library editor. A further and

even more serious complication arises with designs that require multiple voltage levels. Multiple voltage designs, such as PC motherboards with 3.3V and 5V logic are now commonplace. Only two options exist for such designs: create custom parts with visible power pins, or separate all 3.3V and 5V devices onto different sheets and isolate power going to these sheets.

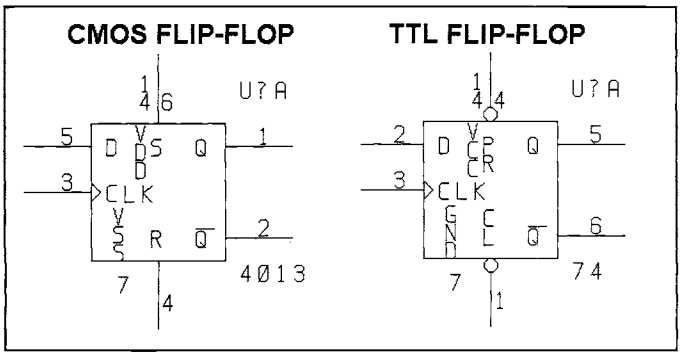


Figure 1-19 Invisible Power Pins

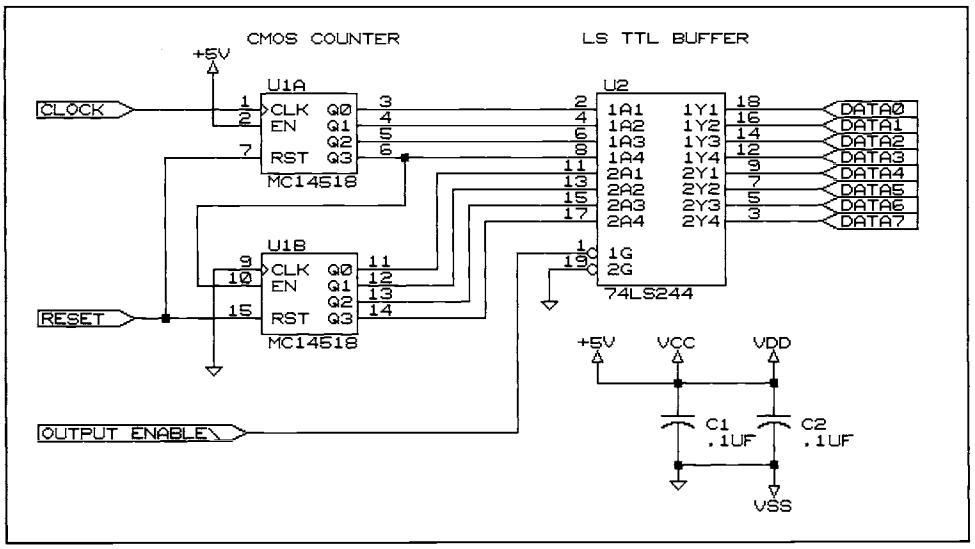


Figure 1-20 Design with CMOS and TTL Logic

The invisible power pins are automatically tied to nets with the same name. In Figure 1-20, the TTL part is already connected to ground since the invisible ground pin is named GND. The CMOS part has an invisible ground pin named VSS. In order to connect this pin to ground, you must place a VSS power object on the sheet and then tie that object to ground. The TTL part uses VCC for positive power, the CMOS part uses VDD. To connect these pins to +5V, you must add power objects with the names VCC and VDD and then tie them to +5V. You should add a text note explaining what these power and ground ties represent. Then add another text note listing which IC pins are power and ground. Do not assume that the average technician or even fellow engineer will understand invisible power and ground pins without some explanation or that they can remember which pins are used on every possible device.

Schematic Organization and Planning

Two schools of though exist as far as how to get started on a design using schematic capture software. Younger engineers often believe that the computer screen is the new demigod of design. Start with an empty screen and try to organize your thoughts as you go along. Use the computer as an electronic sketch pad. Older engineers, on the other hand, tend to view schematic capture as a drafting tool useful for preparing neat documentation at the completion of a design project.

Neither approach gives optimum results. Traditional desktop PCs and engineering workstations are not electronic sketch pads. Attempts to make PDAs (personal digital assistants) with stylus and handwriting recognition for use as electronic sketch pads have not met with much success - witness the AT&T EO and Apple Newton. A truly useful PDA is still years away. In the meantime, use an old fashioned lab notebook.

Designs can be evolutionary or revolutionary. Products evolve and even an entirely new design may consist largely of recycled circuitry. Truly revolutionary designs starting with a clean slate are rare. If the so called new design is merely a glorified engineering change, starting straight out in OrCAD, with the previous version of the schematic, may be the best way to go. The more revolutionary the new design, the more planning and organization required up front:

- Start with a clear design specification. The specification should include features and performance, interface requirements, physical characteristics (size, weight, appearance), operating environment (temperature, humidity, vibration), regulatory compliance (FCC, UL), and manufacturing cost target.

- List all integrated circuits and other specialized parts required for the design. Run copies of databook pages for all parts.
- List signal names and buses. Interface signals are often established ahead of time from the design specification. The choice of processor will determine data, address, and bus control signals.
- Plan power usage and representation.
- Create any required new parts in your custom library.
- Sketch out a rough block diagram and use it to plan out the design hierarchy.
- Draw the schematic using OrCAD. Add copious text notes. Run the electrical rules checking routine to catch errors that might slip by, then plot or print out the schematic.
- Run postprocessing routines to generate a bill of materials for manufacturing and a netlist for PCB design. Carefully examine the output. It will often reveal subtle errors.
- After the PCB design and board debugging is completed, go back and incorporate any engineering changes and other changes necessitated by PCB layout considerations.

Conclusion

In this first chapter you have learned the background information required for capturing schematics with OrCAD. A clear understanding of this information is a prerequisite for successfully using OrCAD. In subsequent chapters, the focus will shift to the internals of OrCAD.

2

Installation and Configuration

This chapter covers installation and configuration of the OrCAD ESP Design Environment and SDT Schematic Drafting Tools software. You will learn how to install and configure the software in preparation for the tutorial exercises in the following chapters. Please read this chapter very carefully.

The OrCAD SDT software runs in the OrCAD ESP environment. ESP is a graphical user environment which shares some similarities to Microsoft Windows. Programs and routines can be launched by using a mouse. Commands can be entered via menu selections using either the mouse or the keyboard. A macro capability allows user defined command sequences to be executed by pressing hot keys.

OrCAD evolved before conventions for the use of mouse pointing devices in graphical user environments had become well established. Consequently, command selection via the mouse can be tedious for users accustomed to Windows. Many simple commands, such as drawing a line, require an inordinate number of mouse clicks. To reach an acceptable level of productivity, one must utilize macros for command selection in addition to the use of the mouse.

System Requirements

OrCAD ESP and SDT run in an IBM compatible PC DOS environment. The current version is called "386 Plus" and uses 32 bit data structures. Recommended minimum system requirements for reasonable performance include:

- 486 or Pentium processor. For educational purposes, OrCAD will run on a 386, but with degraded performance.
- 8 MB RAM (2MB is the absolute minimum).
- 10 MB free hard disk space.
- VGA graphics. OrCAD works great with VGA graphics. Higher resolution graphics can pose a problem since lines and grid dots are single pixel width and may become hard to visualize unless a large monitor is used.

- Microsoft compatible, two button mouse.
- DOS 5.0 or higher. DOS 6.22 is highly recommended. OrCAD SDT uses the Phar Lap memory extender to access extended memory. The program is more stable under the latest revision of DOS.

Compatibility problems have been noted when attempting to run OrCAD under Windows or in Novell LAN environments. Generally, the program should be run only under DOS and without any TSR (terminate and stay resident) programs loaded.

Installation

Installation consists of loading and configuring the software. While loading the software is straightforward, the configuration process is quite complex in comparison to other EDA programs. ESP and SDT require separate configurations. SDT has both a global configuration that affects all schematic drafting related routines and additional local configurations for the individual routines. The terms routine and tool are interchangeable. This chapter reviews only the ESP and SDT global configuration process. Local configurations are discussed as the various SDT routines are introduced in subsequent chapters.

The installation process is detailed in the following sections. DOS or program commands to be typed into the keyboard are shown in bold; **<ENTER>** means press the enter key.

Starting the Installation

Insert the Install disk into your floppy disk drive. At the DOS prompt, change to the floppy drive in which you inserted the disk, for example type **A:\ <ENTER>**. Then type **INSTALL<ENTER>** to start the installation program. The installation program will prompt you for required information and ask you to insert the other program disks.

Video Drivers

Because OrCAD is not a Windows program, individual drivers are required for every combination of graphics board and screen resolution. OrCAD comes with a substantial set of drivers. You can install multiple drivers, but as a minimum you should always install the VGA driver and use it as your initial default. If a higher resolution driver is available for your graphic card, you can add it to your selection set. Once the initial installation is complete, a configuration utility is available from within OrCAD to readily change drivers and defaults.

WARNING: If you configure a driver that is not compatible with your graphics card, the program will hang. Even worse, you will not be able to access the configuration utility again to correct the problem. Unless you have made a backup copy of your configuration files (more about this later), you will have to reinstall the program.

Plotter and Printer Drivers

Select drivers appropriate to the hard copy devices attached to your system. You can install multiple drivers and later change the configuration from within the program. The DXF plotter driver is recommended if you plan to export data to AutoCAD. The tutorial exercises in this book assume you are using a Hewlett-Packard LaserJet or similar letter (A) size graphics printer for hard copy. Recommended printer driver resolution is 100 x 100 DPI (dots per inch). Lines and text are printed single pixel width. Higher resolutions result in very fine lines and text that may give poor copy quality when reproduced.

AUTOEXEC.BAT File

The installation program will make changes in your AUTOEXEC.BAT file and you must reboot your system after completing the installation, or the changes will not take effect. OrCAD will not run unless the following lines are in your AUTOEXEC.BAT file to set environment variables:

```
SET ORCADEXE=C:\ORCADEXE\
SET ORCADESP=C:\ORCADESP\
SET ORCADPROJ=C:\ORCAD\
SET ORCADUSER=C:\ORCADESP
```

In addition, your path statement must include C:\ORCADEXE. Normally, OrCAD is successful in making these changes, but if the program fails to run or gives an error message, verify that your AUTOEXEC.BAT file is correct. Again, remember to reboot the PC when the installation is complete in order for these changes to take effect.

Program Installation

Next comes the actual program installation. Select installation of both OrCAD ESP and SDT. The tutorials in this book assume default installation onto the C: drive. Install all product files. The installation routine will ask if prior release (such as Release IV) schematic and library files should be converted into the new

32 bit format. If you are doing a new installation, this is not applicable. You will also be asked to select a display driver; choose VGA for now.

Part Libraries

One of the great strengths of OrCAD is the extensive selection of part libraries. If you are not limited by disk space, you can install every available part library. To complete the tutorial exercises, you will require the following libraries:

ANALOG
CMOS
DEVICE (general electronic parts)
MEMORY
TTL

Netlist Formats

If you are going to be transferring data to a PCB design system, you will know which netlist format is required. As a minimum to complete the tutorial exercises, install the FUTURE (FutureNet) format. FutureNet has several advantages. It is easily read and modified with a text editor and it is widely supported.

Tutorial File Installation

A custom macro and part library file are supplied on disk. You will find these very useful. Insert the disk and copy the file CUSTOM.MAC to C:\ORCAD\TEMPLATE. This copies the custom macro file into the OrCAD template used to create new schematics. Next, copy the file CUSTOM.LIB to C:\ORCAD\ESP\SDT\LIBRARY. This copies the custom part library into the OrCAD library directory.

This completes the OrCAD installation. Remember to reboot your PC. The next step will be to configure OrCAD.

Configuration Overview

Both OrCAD ESP and SDT have separate global configurations. You will set up a default configuration for each program. Later, you can modify these default configurations for each schematic design. The configuration data is stored with each design.

OrCAD ESP Configuration

Start the program by typing **ORCAD<ENTER>** at the DOS prompt. The ESP screen shown in Figure 2-1 will appear. Besides OrCAD SDT, other design programs available from OrCAD appear regardless of whether or not they have been installed. Clicking on a program that has not been installed is ignored. Use Design Management Tools to select the active design prior to launching OrCAD SDT. Click on Exit ESP to return to DOS.

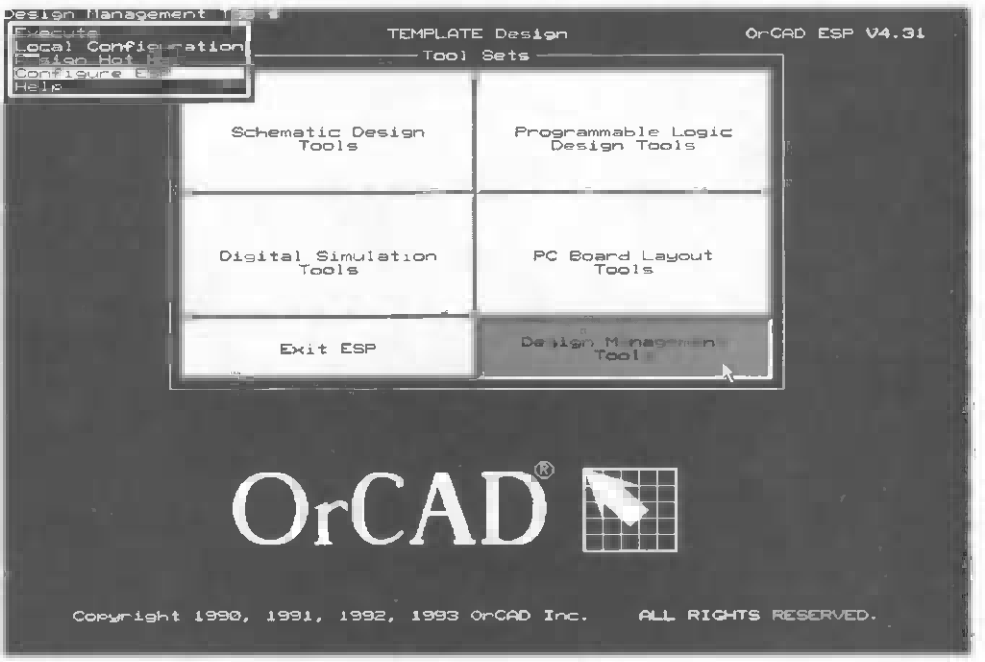


Figure 2-1 OrCAD ESP Main Screen

Using a mouse with OrCAD may take some getting used to for Windows users. The left mouse button acts like the <ENTER> key and generally causes the selection made with the mouse pointer to be executed. The right mouse button acts like the <ESC> key and causes the selection or task to be abandoned or exited. The right button is rarely required. Many actions that you might expect to be carried out by a single mouse click actually require two clicks.

After you have started OrCAD, move the mouse pointer (arrow) and click once on Design Management Tools. This brings up the pulldown menu shown in Figure 2-1. Unlike Windows programs, the mouse pointer now stays put. Moving the mouse vertically causes menu selections to be highlighted. Clicking the mouse again, launches the highlighted menu selection. You can immediately

launch a tool by clicking on it twice. This is the same as clicking on the Execute selection after bringing up the pulldown menu.

Move the mouse to highlight Configure ESP and click on it. The screen shown in Figure 2-2 appears.

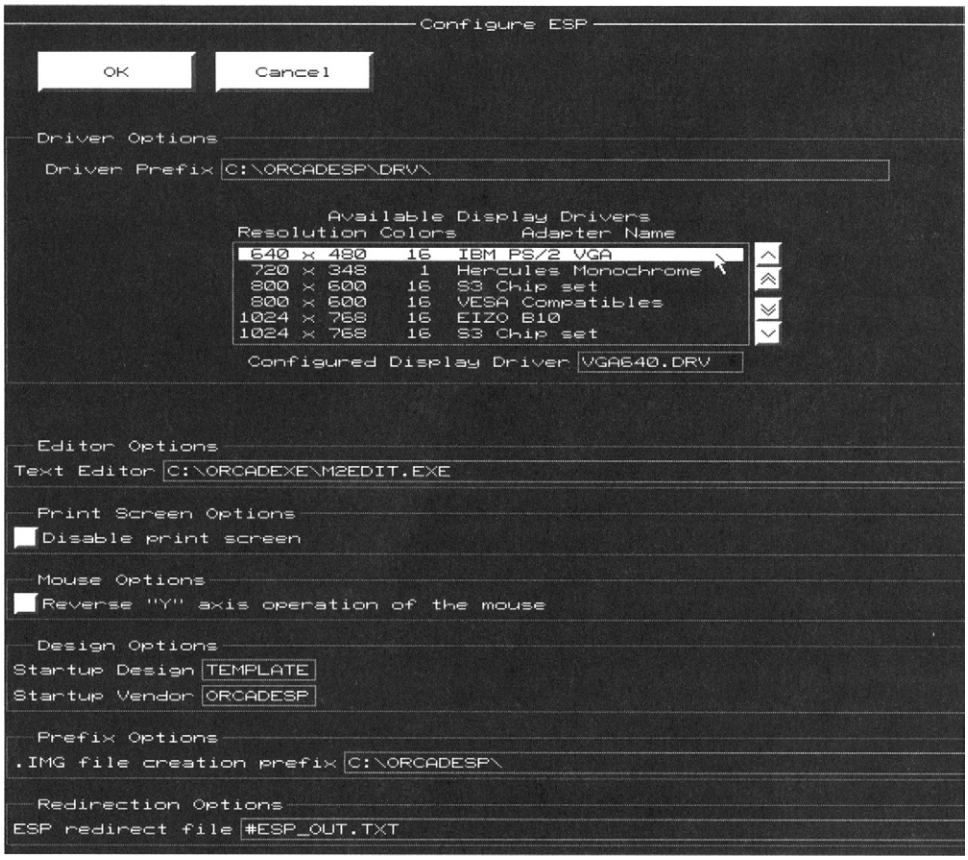


Figure 2-2 ESP Driver Configuration Options

ESP Video Drivers

Confirm that OrCAD ESP has been configured for the VGA driver. You can move the mouse pointer and click on the VGA driver listed under available display drivers to select it. The scroll buttons can be used to navigate through the list of available drivers. The Driver Prefix is actually the path to the subdirectory that contains OrCAD drivers. This was set during installation and should not be changed.

OrCAD uses scroll buttons to navigate lists. However, Windows-type scroll bars are not used with any OrCAD screens. To scroll past the bottom of the screen, you must move the mouse pointer to the very bottom. To scroll back up again, move the mouse pointer to the very top. This type of automatic scrolling or panning is common to all OrCAD screens. You can also use the keyboard: PgUp and PgDn to scroll up and down part of a screen or Home and End to scroll all the way to the top or bottom.

Additional configuration options appear if you scroll past the bottom of the screen shown in Figure 2-2. These options mainly involve file defaults and color selections and need not be changed at this point. Click on OK at the top when you are ready to save the configuration and exit back to OrCAD ESP. You can skip the rest of the material in this section and proceed to OrCAD SDT Configuration.

ESP Options

Additional configuration options shown in Figure 2-2 include choice of text editor, print screen disable, mouse reversal, and path/filenames. For now, leave all these options at the installed default settings until you have completed the tutorials.

OrCAD comes with the Stony Brook M2EDIT text editor and this is the default setting. At a later date, you will want to change to your favorite text editor. The text editor is launched when you select Edit File on the SDT main menu.

Print screen does not work with all graphics boards, and the function is generally not very useful. An option is provided to disable print screen for compatibility with certain TSR screen capture utilities. The purpose of the mouse Y axis reversal option is not clear.

Under Design Options, you can select the default design, which is normally TEMPLATE. TEMPLATE is a subdirectory that contains template files to use as a basis for creating new designs. You will select the design you are going to work on via the Design Management Tools (Figure 2-1).

The Startup Vendor selection has nothing to do with what the name implies. In fact, it sets the prefix for the OrCAD ESP configuration file (ORCADESP.DAT). There is no reason to ever change this.

When you launch an OrCAD tool, such as Design Management or SDT, or return to DOS, OrCAD ESP saves an image of itself. This image is saved to disk using the path specified under Prefix Options. About 230K of free disk space is required. If you have more than 2MB RAM, you can speed things up by loading

the image file in high memory. You can force OrCAD to load the image file into RAM by deleting the text in the Prefix Options box.

Under Redirection Options, you can select the file to which OrCAD tools send command line, program error, and status information. If a tool fails to execute properly, examining the redirect file (default is #ESP_OUT.TXT) can provide clues about what went wrong.

ESP Color Tables

Figure 2-3 shows the ESP color tables. For now, leave the default settings. OrCAD gives you a great amount of flexibility in changing color assignments.

ESP Local Configuration, Hot Keys, and Help

Refer back to Figure 2-1. Even though the menu selection appears, Local Configuration does not exist for any ESP tool. Apparently, OrCAD tried to maintain a consistent menu appearance. Local configurations do exist for the various routines within SDT.

Hot Keys allows assignment of a "hot key" to launch the particular tool. Few users will want to bother with assigning and then remembering hot keys for launching tools that are only a mouse click away.

Help within OrCAD is a disappointment. At most, one short paragraph gives a top level description of the tool. No details are available.

OrCAD SDT Configuration

Click once on Schematic Design Tools. Then click once on Draft. The screen shown in Figure 2-4 appears. Next, select and click on Configure SDT. The SDT configuration program has a long vertical screen that you can scroll through using the mouse. Scroll buttons are used to navigate through the list of available drivers or libraries. The driver or library prefix is a path to a subdirectory that was set during installation and should not be changed.

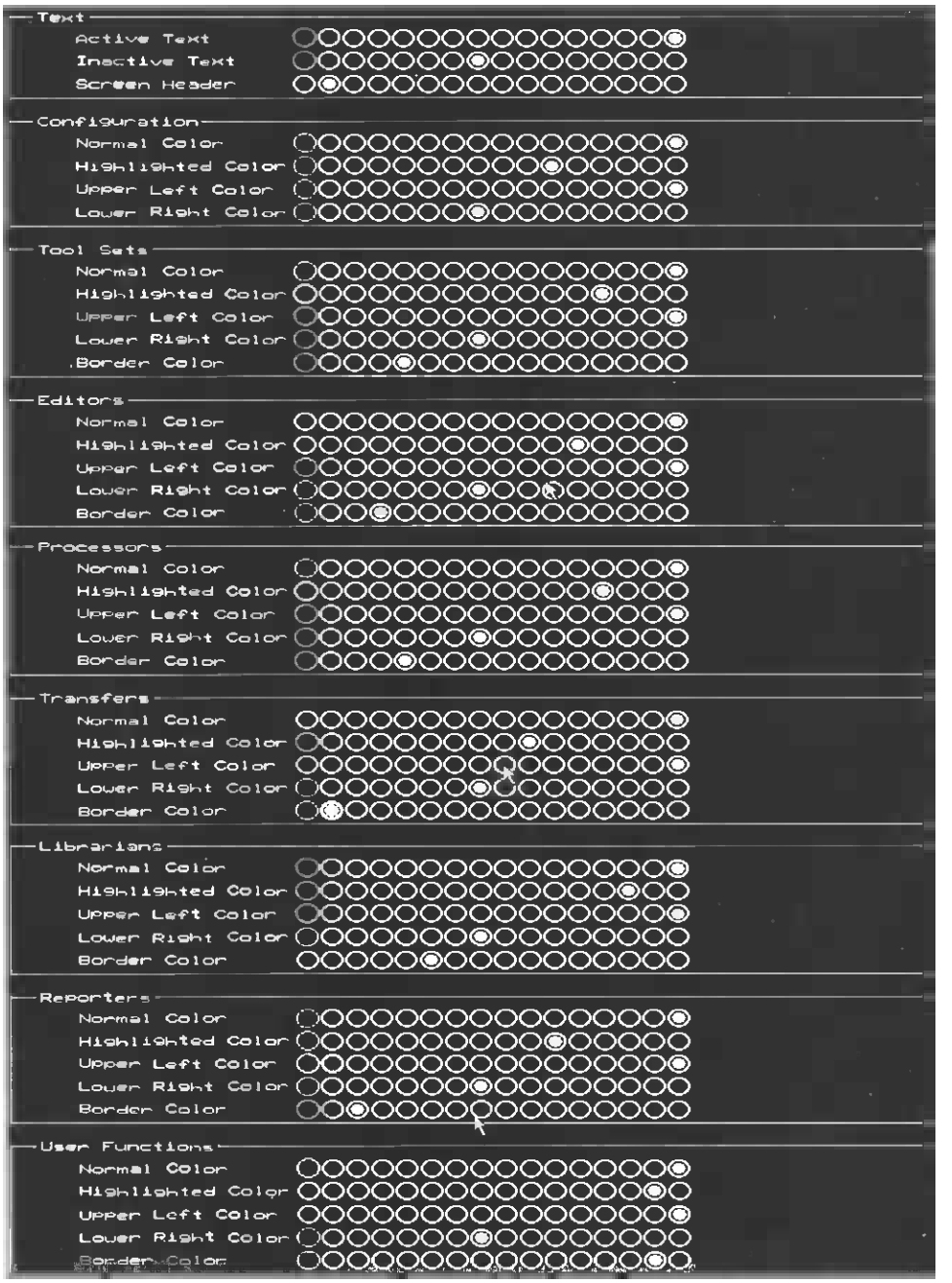


Figure 2-3 ESP Color Configuration Options

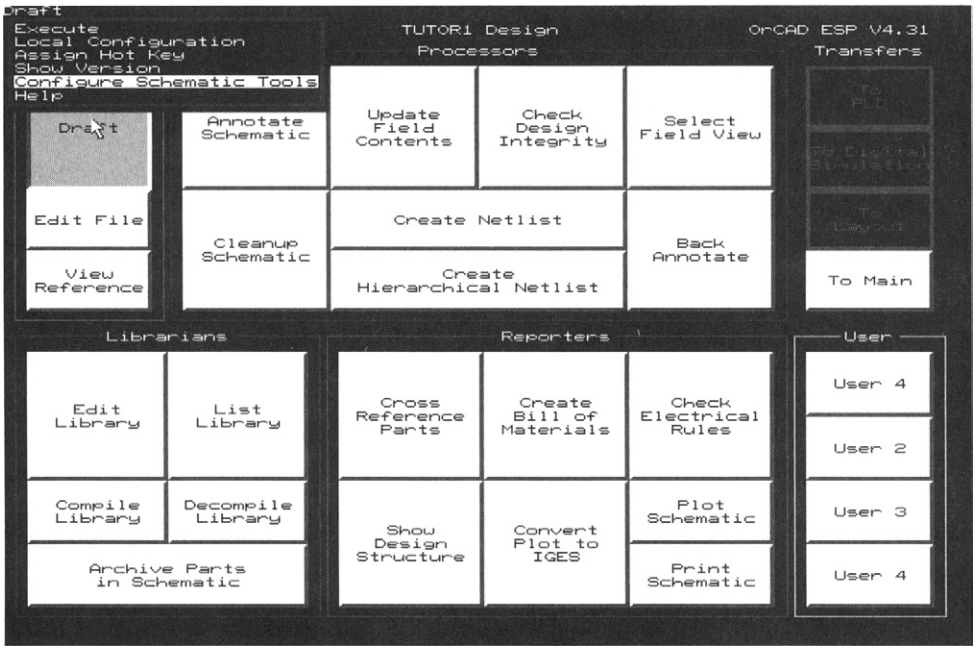


Figure 2-4 OrCAD SDT Main Screen

SDT Drivers

Verify that SDT has been configured for a VGA driver and appropriate printer and plotter drivers as shown in Figure 2-5. Buttons are shown for the printer and plotter output (communications) options. You can use the mouse to click on the desired settings. Printers are usually attached to LPT1. Printer ports LPT1-LPT3 do not have any associated baud rate, parity, or data/stop bit options. If you select LPT1-LPT3, these communications options will be dimmed.

Refer to your PC and hard copy device documentation for interfacing details. Note that your PC may not support all the ports or options available.

Figure 2-5 SDT Driver Configuration Options

SDT Part Library Options

Please read this section carefully so you will have a clear understanding of part library issues. Refer to Figure 2-6. Only configured part libraries are available when running SDT. When a part is inserted onto the schematic, the program searches through the configured library list in order from top to bottom and retrieves the first instance of the part. The custom library should appear at the very top of the list. If you modify a standard part and store it in the custom library under the original name, the modified part will always be retrieved. Remember the warning in chapter 1 about always storing modified parts in a custom library and never changing standard libraries. Otherwise, the changes will be lost when you load updates from OrCAD.

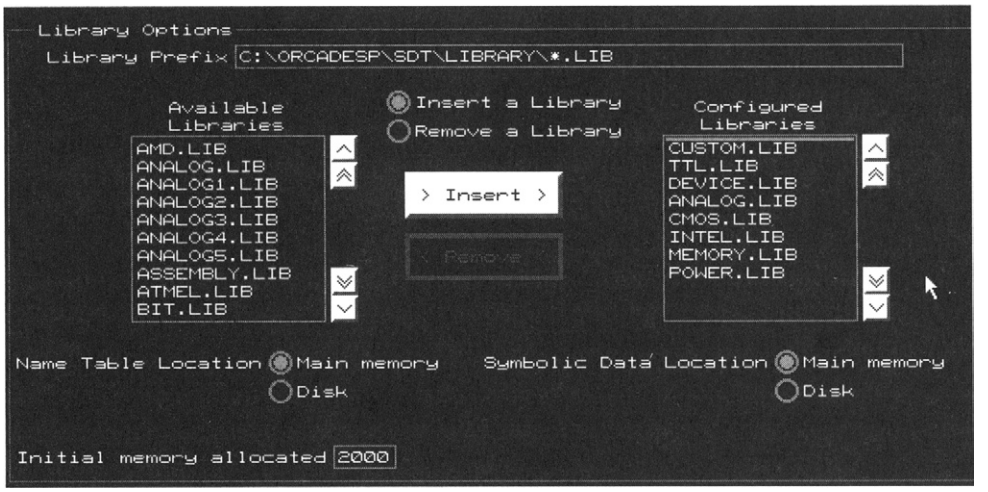


Figure 2-6 SDT Part Library Configuration Options

On startup, SDT reads the configured libraries and creates a name table containing an indexed list of every library part. Symbolic part data (graphics data) is stored separately. Buttons are shown for various memory options. In most cases, a total of six to ten configured libraries will suffice for a given design. Both the name table data and symbolic data can be stored in RAM memory. If you require a very large number of configured libraries and your PC has limited RAM, you can select disk storage for the symbolic data and only suffer a small sacrifice of retrieval speed.

To complete the tutorial exercises, make sure that the libraries listed on page 42 including the custom library (CUSTOM.LIB) are all configured. Libraries can be inserted or removed from the configured library list, however the procedure is

somewhat convoluted. Note that the configured library list has a green bar. This bar indicates the position in which the next library will be inserted. To move the bar, point the mouse cursor at the desired location and click the left button. Move the bar to the very top before you insert the custom library.

To insert a library, you must first click on the Insert A Library button to highlight it. Scroll through the list of available libraries and click on the desired library. Next, click on the Insert box to add the selected library to the configured library list. A similar process is used to remove a library. Click on the Remove A Library button, click on the desired library in the configured library list, and then click on the Remove box.

SDT Worksheet Options

This section of the configuration screen in shown in Figure 2-7. The term worksheet refers to the individual pages of a schematic design. The title block is the area at the bottom of the page that contains identifying and descriptive information. Grid references appear at the outside border and help identify locations on the page. SDT has a standard format for worksheets, title blocks and grid references. This standard format is used for the tutorial exercises. Buttons are provided to select the optional ANSI Y14.1-1980 format. Selecting the ANSI format title block option also changes the worksheet size. ANSI A size is slightly larger and the border will not print correctly on a standard laser printer. In ANSI format, grid references do not appear on A or B sized worksheets. Unless your company or industry requires ANSI format, use the standard OrCAD format.

Ignore the button for alternate worksheet prefix. This provides compatibility with earlier OrCAD versions and should be left blank. It could be used to set a path to worksheet data. The current OrCAD version uses a fixed directory structure determined by environment variables. This is explained in detail later on in the chapter.

The default worksheet extension is used for naming worksheet files if you do not specify a filename extension when creating a new worksheet. Stay with the convention SCH. This results in easily recognizable worksheet file names such as PWRSUPLY.SCH.

The remaining worksheet options are the initial defaults used for new worksheets. You can set up a default size (use A for the tutorials), revision code (use 1.0), and enter in your organization's name and address. The use of default values for document number and title (the description of the design) seems less appropriate, since these will undoubtedly change for every new design.

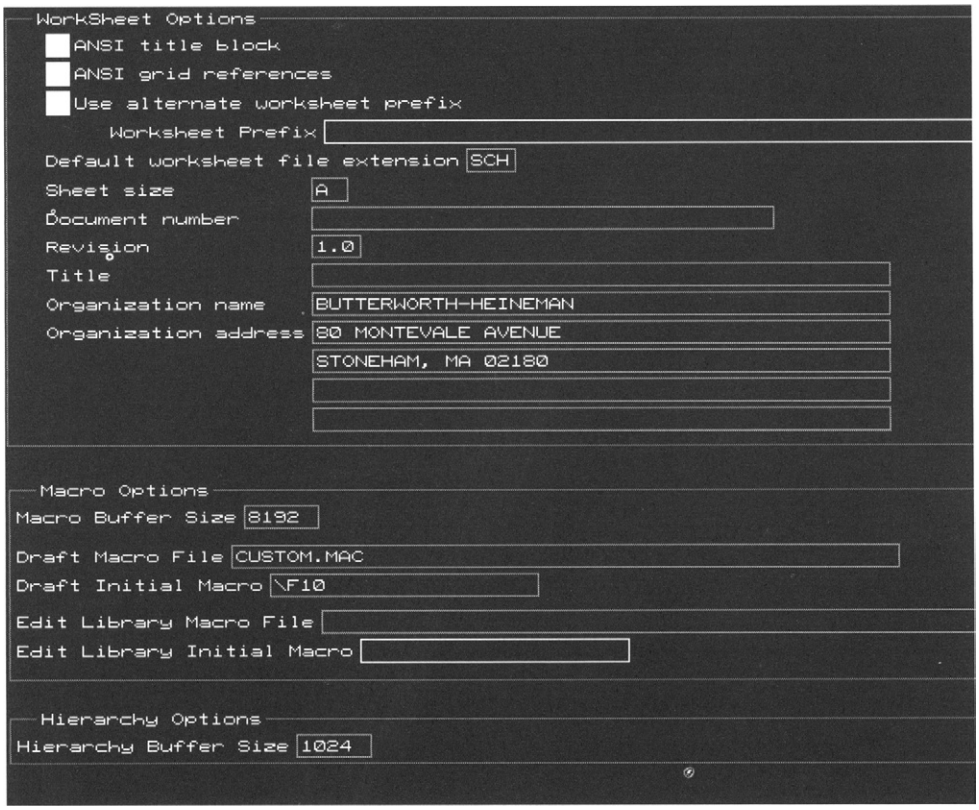


Figure 2-7 SDT Worksheet and Macro Configuration Options

The field for sheet size must be set to A, B, C, D, or E American standard sizes or to A4, A3, A2, or A0 metric ISO sizes. A template table that includes page size and layout data appears further down on the configuration screen. The fields for document number may contain up to 36 characters and for revision code up to three characters. Fields for title and organization name and address may contain up to 44 characters.

SDT Macro and Hierarchy Options

For best results with the tutorial exercises, SDT Draft (the actual schematic drafting tool) should be configured to use the macro file CUSTOM.MAC as shown in Figure 2-7. The initial macro command should be set to \F10, which is OrCAD shorthand for <ALT><F10>. Macro commands are less useful for SDT Edit Library, the parts library editor.

The Macro and Hierarchy Buffer Size values are given in bytes of RAM with default values sufficient for most users. You can increase both values up to 65,500 bytes. The Hierarchy Buffer size of 1024 bytes suffices for designs with up to 75-100 worksheets.

SDT Color and Pen Plotter Table

This section often confuses new users, especially pen number and width setting. First, consider that pen plotters are virtually obsolete. Today most plotting is done with high speed laser or ink jet devices. Modern practice is to print schematics on multiple A size sheets with a laser printer. Few situations require the use of color on a schematic.

Laser printer and ink jet devices emulate pen widths and colors. The mapping of these pen widths and colors can be complex, and unexpected interactions can occur. In most cases, some experimentation will be required to obtain optimum results.

Refer to Figure 2-8. If you are using a laser printer, you can leave the default settings unchanged with the exception of Grid Dots. The default dark gray is difficult to visualize on most monitors. Change the Grid Dots color to light gray by clicking on the button with the mouse.

Valid pen numbers for OrCAD are 1-16, though not all hard copy devices support multiple pens. Pen 0 causes the plotter to pause so you can change pens. Pen 99 cause the object to be ignored (not plotted). Some devices allow multiple plotting speeds. The DEF selection causes the plotting routine to use the plotter's default speed setting.

OrCAD uses only two line widths: thick lines for buses and thin lines for everything else. The user has no direct control over line width within OrCAD. The data entered for pen width has no effect on line width. Pen width information merely controls how many times the plotting routine strokes the pen back and forth to draw a thick bus or object fill. Enter pen width in inch units. If plots have white streaks on buses and filled objects, slightly reduce the pen width.

You can indirectly control line width via the pen number selection. For example, if you want a heavier title block, select a unique pen number for the title block object and then use a wider pen in that position. Similar techniques can be used with laser and ink jet printers which allow multiple pen widths.

Selection boxes appear for the 1st through 8th Part Fields. You can assign names for theses fields, for example a field could be used for a vendor part number. You are not required to assign a name in order to use a field, but doing so will make

the name appear when you edit the part descriptions during the design process. To change a name, click on the box and type in the new name.

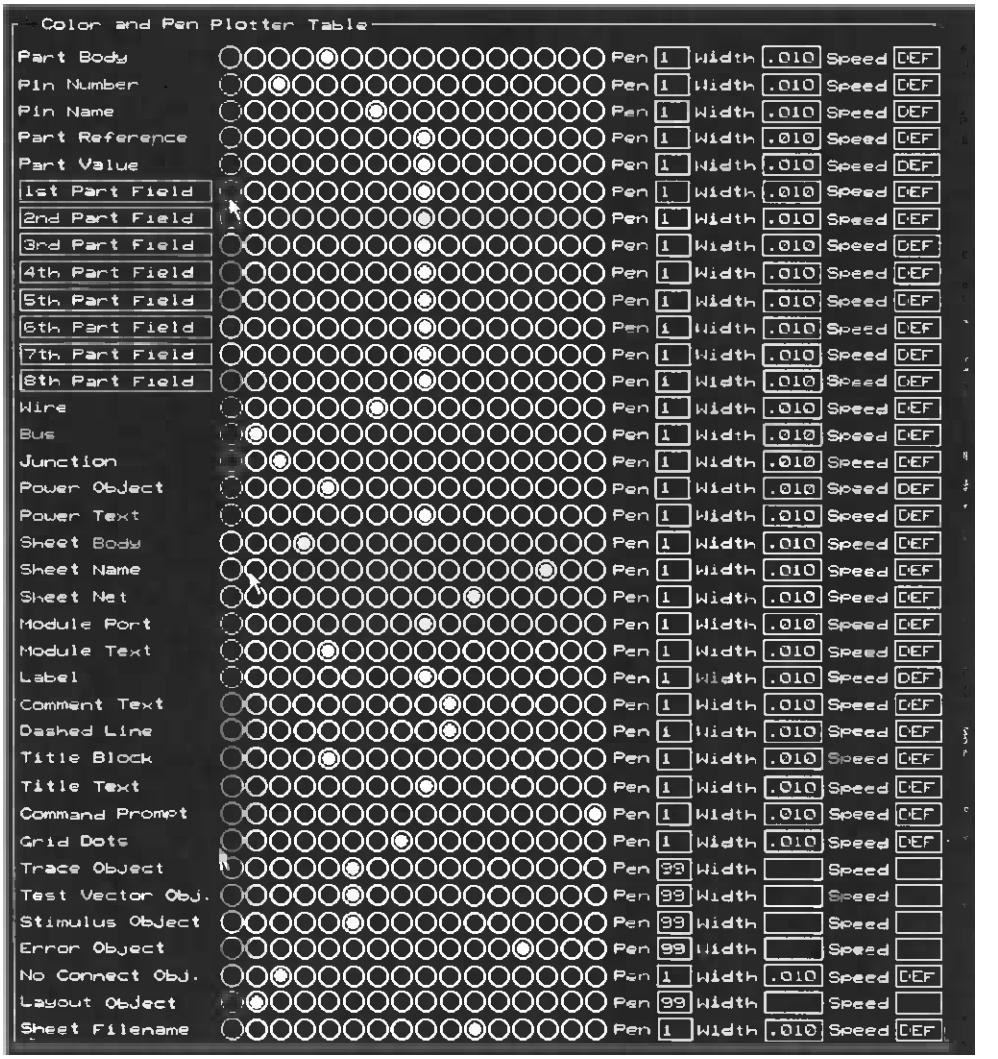


Figure 2-8 SDT Color and Plotter Configuration Options

SDT Template Table

The template table sets global parameters for drawing sheet and object sizes. Do not confuse the template table with the template file used for creating new designs. There is no connection between these. SDT supports both English (inch) and metric (millimeter) units. Five preconfigured sheet sizes are available for each unit selection.

The tutorial exercises use portrait oriented A size sheets and inch units. Change the horizontal dimension to 7.5 inches and the vertical dimension to 9.9 inches. This should give good results with most laser printers. All other values can be left with the defaults shown.

Maximum plotting dimensions are limited to 32 x 32 inches, irregardless of what is entered. The overall scaling is controlled by the Pin-to-Pin spacing value, which is nominally .1 inch. This setting also controls the spacing of grid dots on the screen display. The border is drawn inside the rectangle defined by the horizontal and vertical worksheet dimensions.

Most Hewlett-Packard plotters, and other plotters emulating HPGL (Hewlett-Packard Graphics Language), place the origin at the center of the drawing area. A bug in the OrCAD SDT plotting routine causes an improper origin with HPGL compatible plotters. To work around this, enter a negative offset of approximately 1/2 the plot dimension, as shown on the C size template in Figure 2-9.

Roll Form Size determines the amount of extra paper feed after completion of a plot and between successive plots. Roll Form Size applies only to roll feed plotters.

Spacing Ratio controls how text is spaced. The default value of 1.333 gives good results for most designs.

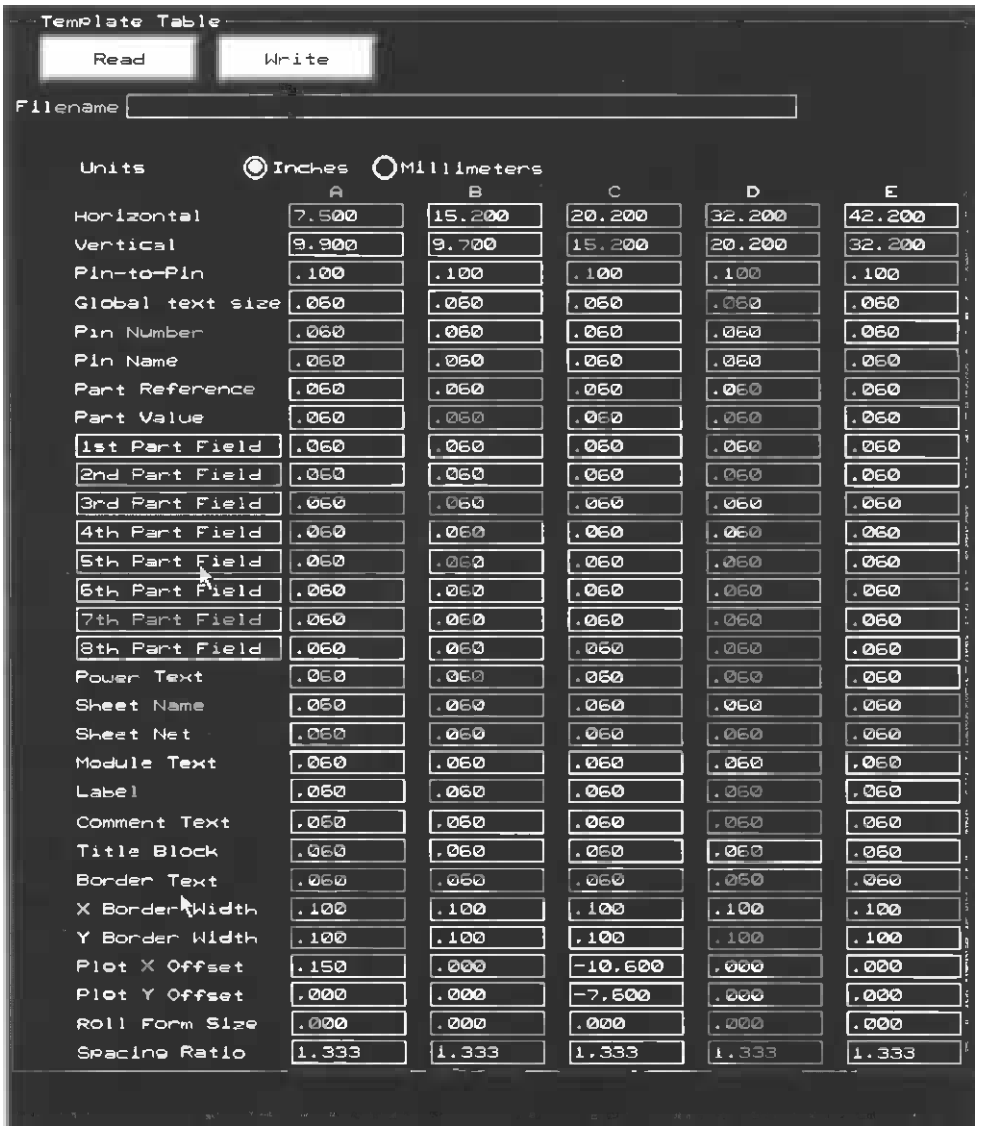
You can save and retrieve user defined templates with the Read and Write buttons after making an entry in the filename box.

SDT Key Fields

You can skip this screen for now. No changes or entries are required on the key fields screen shown in Figure 2-10.

Certain special postprocessing operations can compare, merge, or update information stored in the part fields shown in Figures 2-8 and 2-9. Key fields are used to control these postprocessing operations. The use of key fields is a complex operation that is rarely required. More details are given in chapter 7.

For most practical applications, the same results can be obtained with much less effort by using a text editor or spreadsheet program.



	A	B	C	D	E
Horizontal	7.500	15.200	20.200	32.200	42.200
Vertical	9.900	9.700	15.200	20.200	32.200
Pin-to-Pin	.100	.100	.100	.100	.100
Global text size	.060	.060	.060	.060	.060
Pin Number	.060	.060	.060	.060	.060
Pin Name	.060	.060	.060	.060	.060
Part Reference	.060	.060	.060	.060	.060
Part Value	.060	.060	.060	.060	.060
1st Part Field	.060	.060	.060	.060	.060
2nd Part Field	.060	.060	.060	.060	.060
3rd Part Field	.060	.060	.060	.060	.060
4th Part Field	.060	.060	.060	.060	.060
5th Part Field	.060	.060	.060	.060	.060
6th Part Field	.060	.060	.060	.060	.060
7th Part Field	.060	.060	.060	.060	.060
8th Part Field	.060	.060	.060	.060	.060
Power Text	.060	.060	.060	.060	.060
Sheet Name	.060	.060	.060	.060	.060
Sheet Net	.060	.060	.060	.060	.060
Module Text	.060	.060	.060	.060	.060
Label	.060	.060	.060	.060	.060
Comment Text	.060	.060	.060	.060	.060
Title Block	.060	.060	.060	.060	.060
Border Text	.060	.060	.060	.060	.060
X Border Width	.100	.100	.100	.100	.100
Y Border Width	.100	.100	.100	.100	.100
Plot X Offset	.150	.000	-10.600	.000	.000
Plot Y Offset	.000	.000	-7.600	.000	.000
Roll Form Size	.000	.000	.000	.000	.000
Spacing Ratio	1.333	1.333	1.333	1.333	1.333

Figure 2-9 SDT Template Configuration Options

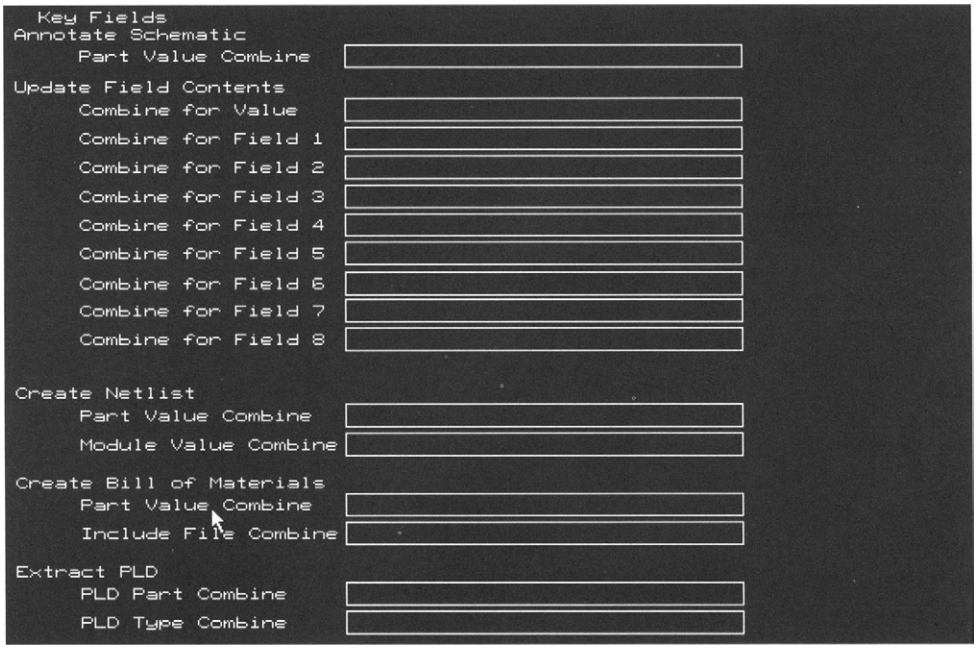


Figure 2-10 SDT Key Fields Configuration Options

SDT Check Electrical Rules Matrix

One of the most useful features of OrCAD is the Check Electrical Rules routine. This automatically detects many common design errors, such as tying inappropriate pins or signals together. The rules matrix shown in Figure 2-11 determines what types of connections are flagged as warnings or errors.

Much feedback from practical design experience has been distilled into this matrix. Occasionally a valid reason will exist for deliberately violating one of these rules. Because the various postprocessing routines can be told to ignore warnings and errors, you should leave the default matrix unchanged.

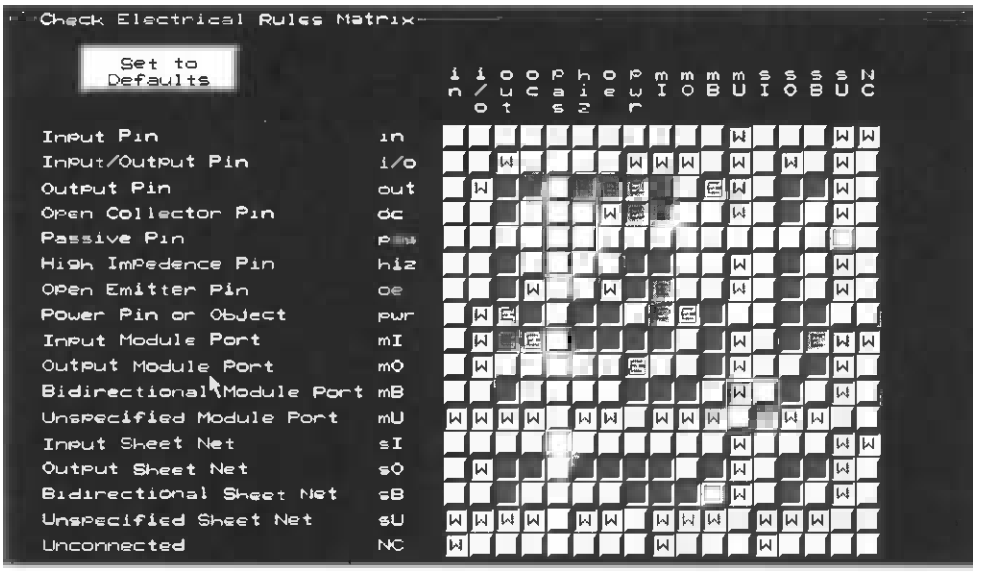
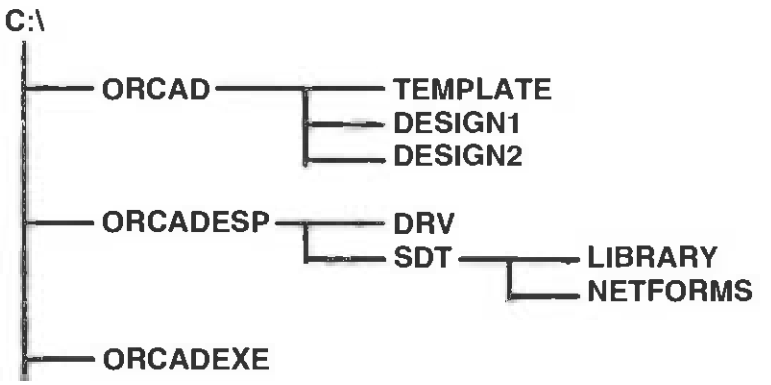


Figure 2-11 SDT Electrical Rules Matrix Configuration

OrCAD Directory Paths

The directories and paths created by OrCAD during the installation process are shown below. An understanding of this directory structure helps when doing file maintenance, especially for design, library, and configuration file backup.



All the files associated with a particular schematic design are stored in a subdirectory under ORCAD. Default ESP and SDT configuration and macro files are stored in the TEMPLATE subdirectory. When you use Design Management Tools to create a new design, it copies these files from the TEMPLATE subdirectory.

ORCADESP contains help files and additional subdirectories. The DRV subdirectory contains all the video, printer, and plotter drivers. Documentation files associated with these drivers are stored in the SDT subdirectory. All library files, including CUSTOM.LIB, reside in the LIBRARY subdirectory. NETFORMS contains netlist postprocessing routines used for data transfer to PCB design systems. OrCAD executable files reside in ORCADEXE.

Configuration Files

Each design has its own ESP and SDT configuration files. If these file are corrupted, the program may fail to execute properly or hang, especially if the video driver is incorrect. Configuration files exists in pairs, one is in an ACSII format that can be edited with a text editor and the other is in a binary format. OrCAD checks the file date and time stamp to determine which is more current. If the ASCII file is more current, OrCAD recompiles it to create a new binary file. File names are as follows:

ESP.CFG	ESP ASCII Configuration
ESP.BCF	ESP Binary Configuration
SDT.CFG	SDT ASCII Configuration
SDT.BCF	SDT Binary Configuration

Make a backup copy of the .CFG files in the TEMPLATE subdirectory. As an exercise, read these files with a text editor and make a printout of each file for future reference. If you ever run into a problem, such as inadvertently changing the configuration, you can always use a text editor to reconstruct the .CFG files.

Two other important files stored in each design subdirectory (and in TEMPLATE) are ORCADESP.DAT and #ESP_OUT.TXT. ORCADESP.DAT contains local configuration data. #ESP_OUT.TXT contains ASCII screen output, result codes, status information, and error reports from the last tool that was executed.

Once set, you will rarely need to change the global SDT and ESP configurations discussed in this chapter. Additional local configuration options exist for the various SDT routines. You will frequently modify these local configurations during the design process. Local configurations are discussed in detail as part of the tutorial exercises in subsequent chapters.

Backing up Design and Custom Library Data

To avoid disaster, always have three independent backups for any mission critical computer data. Assume that data stored on a hard disk drive can disappear at any time without warning. Keep one backup copy on floppy disk and update this data after every computer session. Keep a second backup copy on magnetic tape media, such as QIC-80 tape cartridges. This type of media lends itself to backing up the entire disk. Make a current backup tape every week and store it off-site in case disaster strikes. The third backup can be a paper hardcopy.

OrCAD designs do not include electronic parts data, only the name and instance (location, orientation, part description, and reference designator). The part data (graphic symbol, pin numbers, and pin names) are stored in the part libraries. If library files required for a given design are corrupted or lost, the associated parts disappear from the design. Almost every OrCAD user has been affected by this at one time or another. This is another reason to never modify any of the OrCAD-supplied libraries. Always keep new or modified parts in a custom library. Keep a backup copy of the custom library in with the design data. An alternate technique requires creation of an archive library. The archive library concept is discussed in chapter 6.

3

Tutorial 1 - OrCAD Basics

In the previous chapter, you learned how to install and configure OrCAD. The next seven chapters are tutorial exercises intended to get you off to a fast start. In this chapter you will learn how to create a new design (OrCAD uses the term design to refer to a schematic), place parts, route signal wires, edit part fields, add text notes, run basic post processing operations, and print out hard copy. Your task will be to recreate the programmable power supply schematic shown in Figure 1-17. This figure will be the model for the first tutorial. Take a moment and leaf back to page 30 in chapter 1 and familiarize yourself with the model. You may want to run a copy of the figure and keep it handy for reference.

Creating a New Design

Start OrCAD by typing **ORCAD<ENTER>** at the C: prompt. The ESP design environment screen shown in Figure 3-1 appears. Next double click on Design Management Tools. The screen shown in Figure 3-2 appears.

Click on Create Design. The screen shown in Figure 3-3 appears. Enter the name TUTOR1 for the new design, and click on OK. The program will create a new subdirectory named TUTOR1 under ORCAD (refer back to page 58 for details on the directory structure used by OrCAD). All files associated with your new design are located in this new subdirectory.

You will recall that schematics can be a single sheet or hierarchy with multiple sheets. The data for each sheet is stored in a separate file. Schematic files always have the extension .SCH. The filename prefix for a single sheet schematic or for the root sheet of a hierarchy must be the same as the design name, TUTOR1.SCH in this case. You can assign arbitrary filename prefixes for other sheets in a hierarchy. Because these are DOS filenames, the prefix is limited to eight alphanumeric characters.

The screen shown in Figure 3-3 also has buttons for selecting copy options. When you create a new design, all the files are copied from the TEMPLATE subdirectory. In this case, the schematic template file TEMPLATE.SCH is copied to TUTOR1.SCH. ESP and SDT configuration files are also copied if the Copy all files button is highlighted. This is the default and in most cases, you will

want to copy the configuration files. You can always edit the configurations later. Each design can have its own unique configuration.

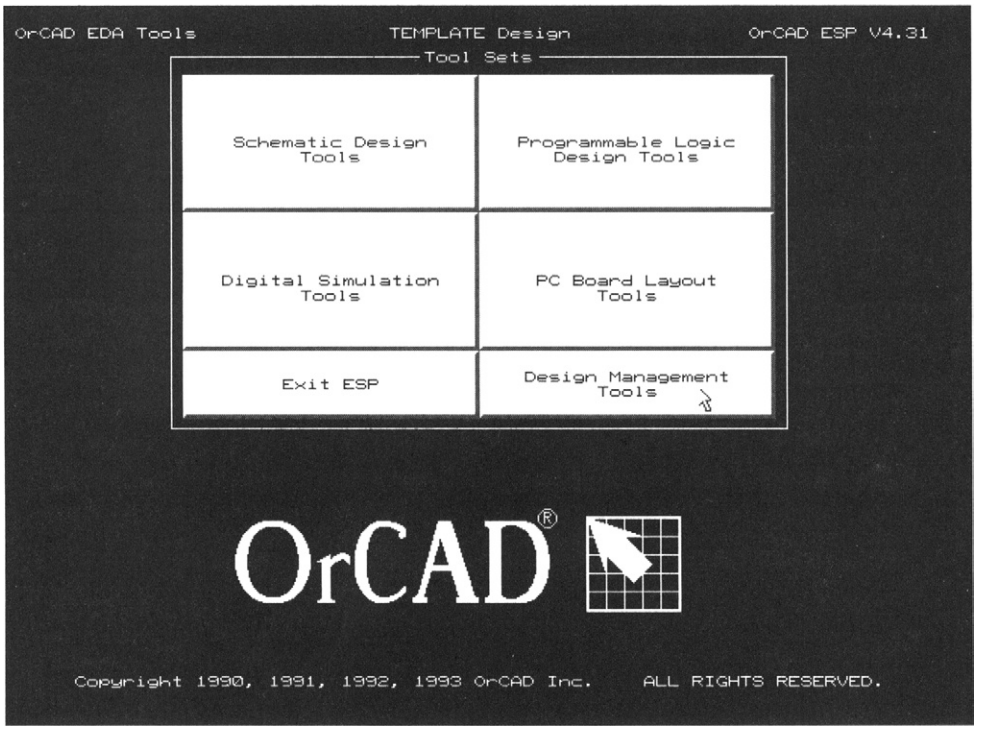


Figure 3-1 Launching Design Management Tools

Using DOS to Create a New Design

You can also create a new design by using DOS commands. To create the TUTOR1 design, you would first use DOS to make a new subdirectory named TUTOR1. You would then copy all the files from the TEMPLATE subdirectory to TUTOR1. Finally, you would change to the TUTOR1 subdirectory and rename the schematic template file TEMPLATE.SCH to TUTOR1.SCH. The sequence of DOS commands would be:

```
MD C:\ORCAD\TUTOR1<ENTER>
CD C:\ORCAD\TEMPLATE<ENTER>
COPY *.* C:\ORCAD\TUTOR1<ENTER>

CD C:\ORCAD\TUTOR1<ENTER>
```

REN TEMPLATE.SCH TUTOR1.SCH<ENTER>

In many cases you will want to copy an existing design, perhaps to make a major revision or to uses parts of it for a new project. You will find that using DOS to copy files and create the new design is actually quicker and more convenient than using the OrCAD Design Management Tools.

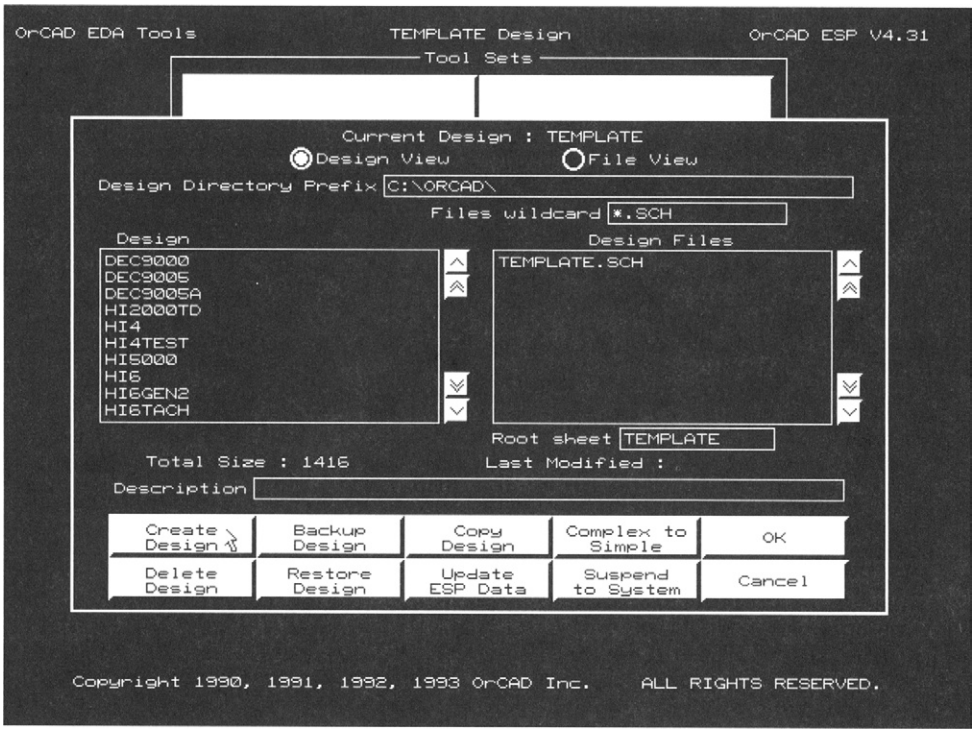


Figure 3-2 Design Management Tools Main Screen

Selecting a Design

Design Management Tools provides a very convenient and fast way to select a design prior to launching SDT. Refer to Figure 3-4. The screen shows the design directory prefix, which is the OrCAD term for the path to the directory where designs are stored. This is normally C:\ORCAD\ and should not be changed. Note that the path must end with a backslash (\).

Buttons are provided for Design View and File View. Design View is the default and allows you to work on entire designs. File View allows manipulation of

individual files, somewhat like the Windows File Manager. You will find DOS commands much easier to use.

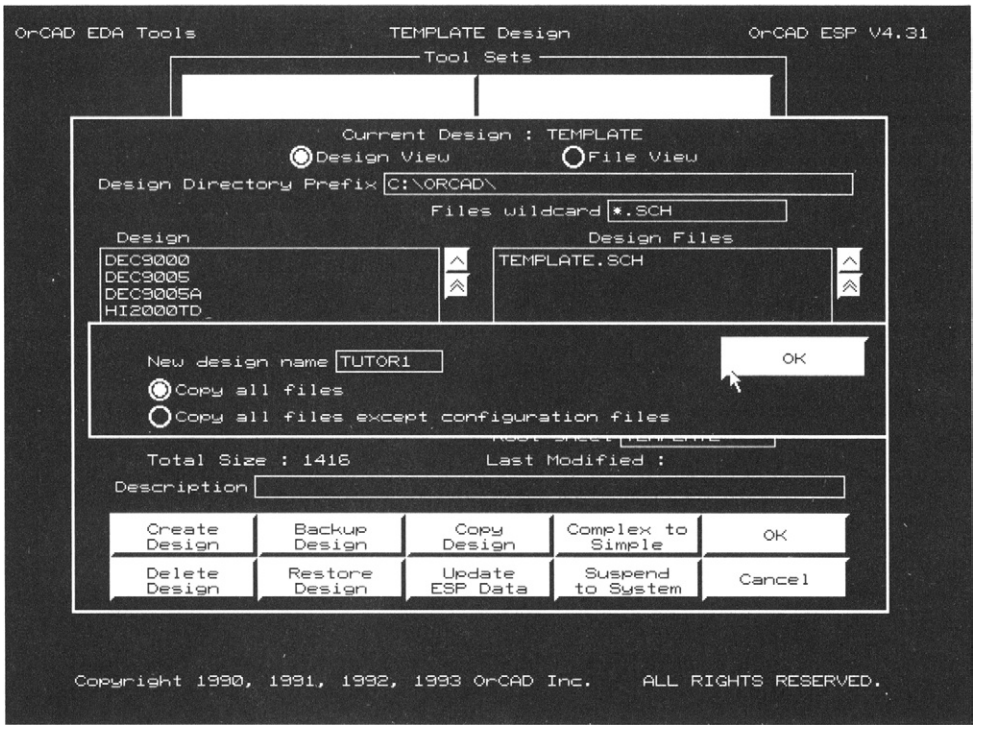


Figure 3-3 Entering the Name for the TUTOR1 Design

The screen is split into two list boxes that you can scroll through. The list box on the left displays all designs on disk. OrCAD can handle up to 150 designs under a given design directory. Use the mouse to scroll through the list of designs. Then click on the desired design, TUTOR1. It will become highlighted to show that you have selected it.

The list box on the right displays all files in the selected design that match the files wildcard entry at the top. Using *.* as wildcard entry will display all files in the design. If you want to see just the schematic files, use *.SCH. OrCAD can display up to 1600 files in one design.

The root sheet name is also shown on the right side. This is normally the same as the name of the design. OrCAD allows you to change the name of the root sheet, but doing so can create problems later on.

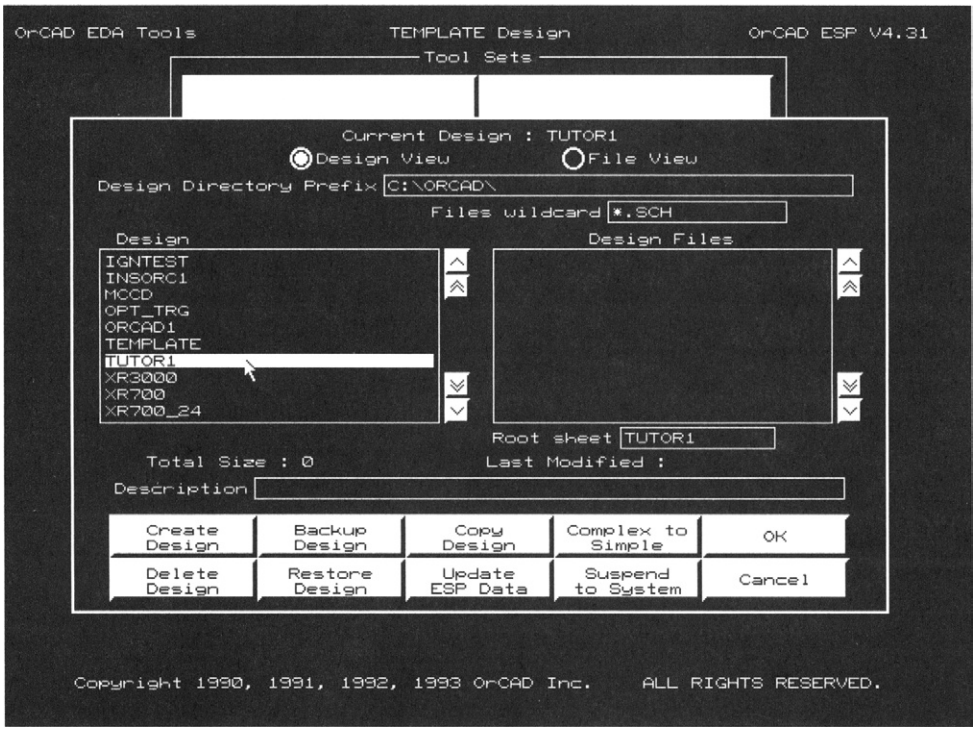


Figure 3-4 Selecting the TUTOR1 Design

After you have selected the TUTOR1 design by clicking on it, click on OK. This terminates Design Management Tools and returns you to the main ESP screen shown in Figure 3-5.

Next, double click on Schematic Design Tools to launch the program using the selected design. You are now ready to start capturing your first schematic.

The Schematic Design Tools screen shown in Figure 3-6 appears. This screen shows all of the individual tools that are available for schematic capture, library maintenance, and postprocessing. The Draft tool is used to draw the schematic. Double click on Draft to launch the tool.

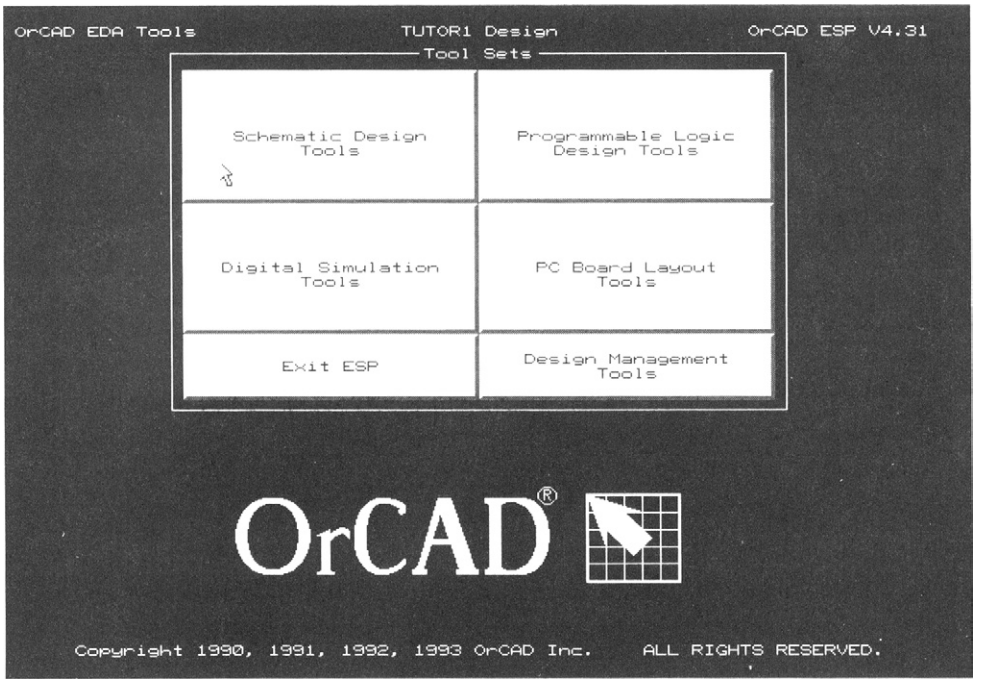


Figure 3-5 Launching Schematic Design Tools

Selecting Draft Commands

After you have launched Draft, the screen shown in Figure 3-7 appears. If you click the left mouse button, the main command menu appears. Once a menu appears, the arrow cursor remains stationary. Vertical mouse movements cause menu commands to become highlighted. The highlighted menu command can be selected by clicking the left mouse button again. A sub menu will appear, revealing more menu selections. Remember that the left mouse button is always used to select a highlighted entry; the left mouse button is the same as the <ENTER> key. The right mouse button is the same as the <ESC> key and can be used to escape or close a menu.

Two other methods can be used to select and execute commands. You can type in the first letter of the command. In most cases, you have to go down two menu levels to access an actual command, so you have to type in two letters. For example, the most frequently used command is Place Wire. This causes Draft to start a "wire," which is an electrical connection between two points. Typing the characters PW (no <ENTER> is required) would start the Place Wire command.

Typing in command abbreviations is a somewhat of a throwback to the old days. Today, no one has time to learn and remember dozens of abbreviations for commands. With the ascent of GUIs (graphical user interfaces), the mouse has become the primary means of command entry. In modern EDA environments, use of the keyboard has been relegated to launching a few macro commands and typing in part data and text notes.

In OrCAD, the use of macro commands greatly reduces the number of mouse clicks required to execute common command sequences. Macro commands can be launched by pressing a function key. You can keep one hand on the mouse to move the cursor and click on points. The other hand can remain on the keyboard to select macros. This leads to good ergonomics and enhanced productivity.

Most keyboards have at least ten function keys and you will rarely need to access more than ten macros. The CUSTOM.MAC macro file, loaded as part of the configuration process in chapter 2, contains ten macro commands accessed by keys F1 through F10. These macros will handle 90 percent of the most common schematic drafting tasks.

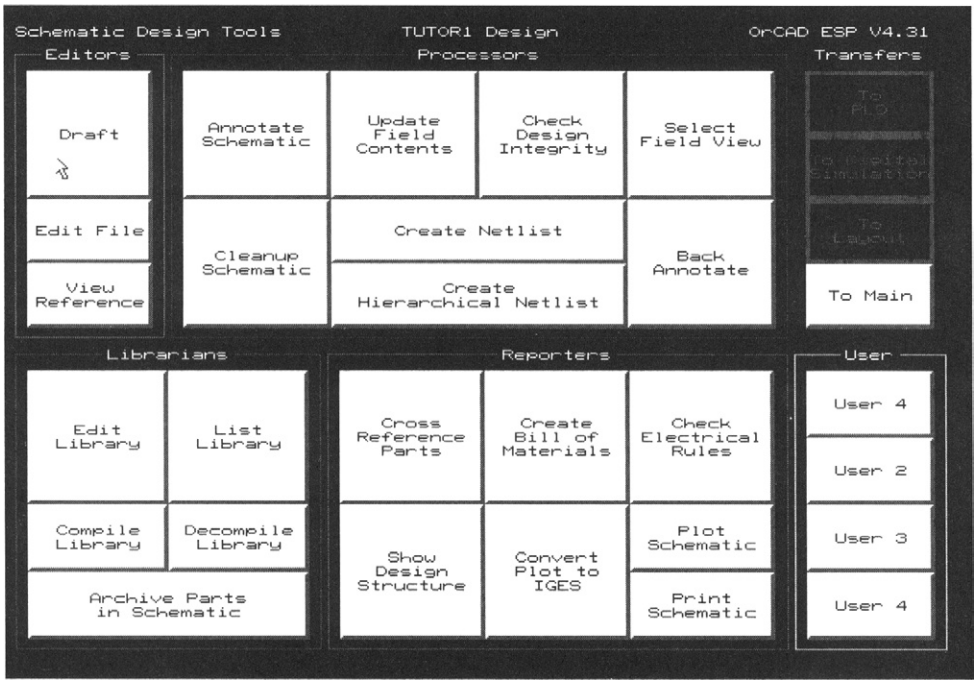


Figure 3-6 Launching Draft

Draft Command Overview

The main Draft menu is shown in Figure 3-7. An overview of the main menu commands follows. Most of the commands are explained in detail further on in the tutorial lessons. Surprisingly, almost half of the commands are of limited use and can be ignored.

Again Repeats the last command, but has a flaw in it so that it only repeats the first menu level. For example if the last command was Place Wire, using Again would only get you to the Place menu. You can forget about this command.

Block Commands on the Block menu allow manipulating selected areas of the drawing. You can move, rubberband, copy, and paste areas of the drawing. Block commands are extremely useful for editing.

Conditions Another command you can forget about. Used to check how much memory was left back in the days when PCs came with 640 KB of RAM and a large design could use up the available memory.

Delete Allows deleting individual objects or entire areas of the drawings. Also has a lifesaving Undo option if you make a mistake.

Edit Used to edit text, part descriptions, reference designators, and many other objects and fields. Unlike most other Draft commands, you must first position the mouse cursor to select an object and then launch Edit. Options will depend on the nature of the object selected.

Find Locates the cursor at the object that contains a character string entered with the Find command. Occasionally useful for finding components based on part value or reference designator.

Get Accesses the OrCAD libraries for placing a part on the schematic.

Inquire Specialized command for accessing hidden text associated with certain objects. Not frequently used.

Jump Moves the cursor to a preset tag location (see Tag command), to a grid reference, or along the X or Y axis. Forget about this command and use your mouse.

Library Displays available libraries and the parts in the libraries. This command duplicates much of the functionality of the Get command, except that parts cannot be placed into the drawing. Use Get instead.

Macro Used to capture macro commands and manage macro files. Not required to access the CUSTOM.MAC file loaded at startup.

Place Used to place (draw) various objects onto the schematic, including wires, buses, junctions, module ports, sheets, and text.

Quit Similar to the File menu found in Windows programs, Quit provides access to file management functions. Also used to navigate between levels in hierarchical schematics.

Repeat Duplicates the last object placed into the schematic. By setting parameters with the Set command, labels or module port names can be automatically incremented and the objects placed on adjacent grid locations. This can result in significant time savings for 16 or 32 bit wide address or data buses.

Set Allows changing parameters for numerous Draft options including the worksheet size and grid.

Tag Sets up to eight tag locations for use with the Jump command. Not frequently used.

Zoom Used to change the zoom scale. Normal scale is 1, which provides the greatest level of detail. Scales 2-10 zoom out to show larger areas. Unless you configure OrCAD for very high resolution and have a large monitor, only the normal scale 1 is useful.

Before you proceed, spend a few minutes, select the most frequently used commands (Block, Delete, Edit, Get, Place, Quit, and Set) and explore their submenu options. These options will be explained in more detail as they are introduced during the tutorial exercises.

CUSTOM.MAC Macro Overview

The use of the macro commands in the CUSTOM.MAC macro greatly facilitates completing the tutorial exercises. You will also find these macro commands useful as a basis for creating your own custom macros as you start to use OrCAD on a professional level. If you followed the installation and configuration steps outlined in chapter 2, CUSTOM.MAC will automatically be loaded whenever you launch Draft. The following macro commands are available:

F1 **Place wire** starting at the cursor location. The wire "rubberbands" with the mouse cursor. Click the left mouse button to add corner points. Click the right mouse button to exit.

F2 **Place junction** at cursor location and exit.

F3 **Place +5V power object** at cursor location and exit.

F4 **Place +12V power object** at cursor location and exit.

F5 **Place ground object** at cursor location and exit.

F6 **Get resistor** from parts library. The part moves with the mouse cursor. Click the left mouse button to place the part at the cursor location. The macro remains modal until the right mouse button is clicked to exit.

F7 **Get capacitor** (functions the same as F6).

F8 **Get electrolytic capacitor** (functions the same as F6).

F9 Get diode (functions the same as F6).

F10 **Get NPN transistor** (functions the same as F6).

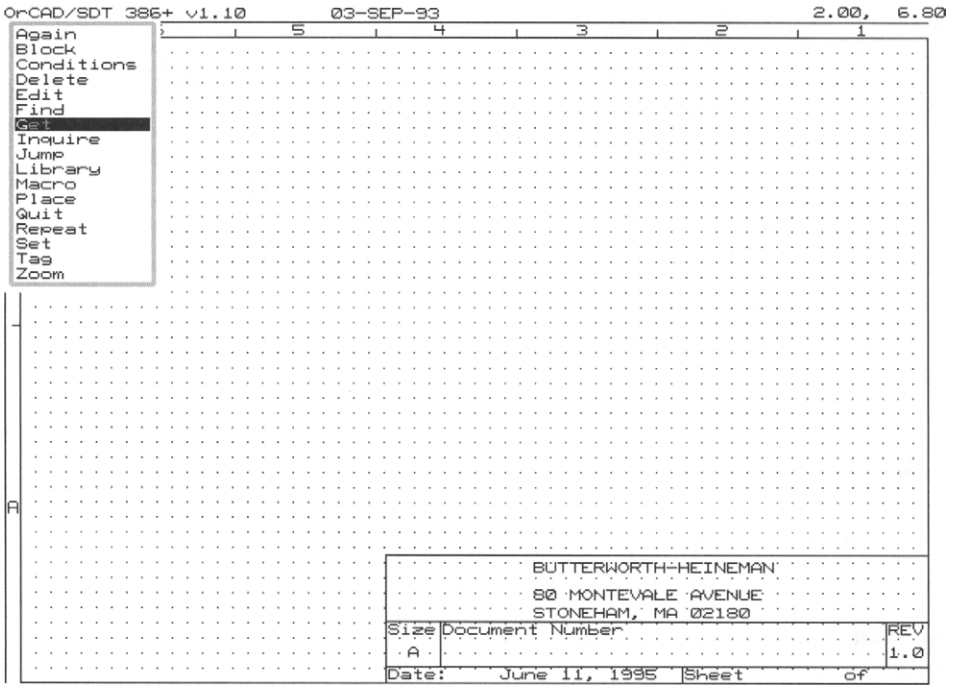


Figure 3-7 The Design Screen and Main Menu

In addition to these macro commands, a special startup macro command, <ALT> F10, is automatically executed when Draft is launched. The startup macro conveniently sets the following Draft options:

- **Drag buses**. Allows bus objects to be dragged (moving them during editing operations). This requires more memory and processing power and slows things down slightly. With a 486 class PC, the difference is imperceptible.
- **Left mouse button release sends <ENTER> keystroke**. This improves the operator interface by eliminating the requirement to double click the left mouse button when placing objects. Speeds up the overall schematic capture process.
- **Show cursor X and Y coordinates**. This feature was made an option due to a speed penalty on slow PCs caused by constant screen updates for the coordinate display
- **Grid references**. Shows alphanumeric grid references on edge of screen. Most users find grid references helpful for maintaining a sense of orientation on the sheet.
- **Visible grid dots**. Most users prefer visible grid dots.

Starting the Tutorial Exercise

The first tutorial exercise covers 80 percent of the commands that you will routinely use in OrCAD for schematic capture: getting parts from the libraries and placing them on the sheet, interconnecting the parts, adding ground, power, junction, and text objects, and editing descriptions.

Keep a copy of Figure 1-17 handy for reference, because this is the model for the tutorial schematic. Start by orienting yourself on the sheet. Move the mouse cursor to pan to the lower right corner of the sheet as shown in Figure 3-7. Please note that whenever the tutorial instructions call for clicking the mouse, you must move the mouse to highlight the menu selection or place the cursor in the desired location and then click the left mouse button.

Plan on spending several hours to complete the first tutorial. If you run out of time and must stop, skip ahead to Saving The Design on page 107. Save what you have completed. You can launch the TUTOR1 design again when you are ready to continue the tutorial.

Getting a Library Part

The procedure for getting and placing a part is to use the Get command to select a library, then select the part, orient the part, and finally place the part on the sheet.

Place PL1, the female DB 9 pin connector. Click the left mouse button to bring up the main menu as shown in Figure 3-7. Click on Get. Click the mouse again to bring up the library menu shown in Figure 3-8. Click on DEVICE.LIB, the library of general electronics parts. Next, scroll through the available parts in DEVICE.LIB. Note that connectors start out with the description CONN or CONNECTOR. Click on CONNECTOR DB9 as shown in Figure 3-9.

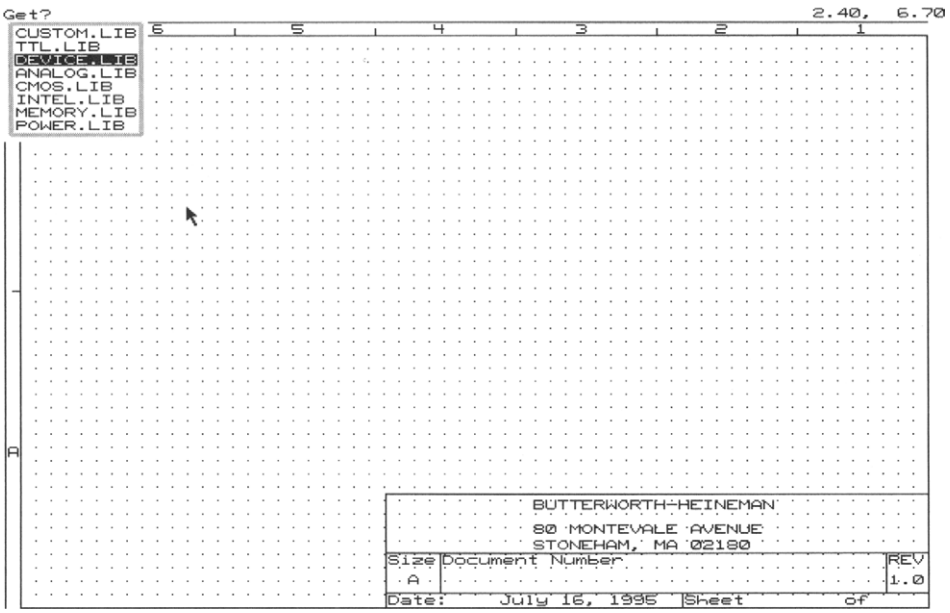


Figure 3-8 Parts Library Selection Menu

The procedure given above for selecting a part involves two steps: scrolling through a list of libraries to select a library and then scrolling through a list of parts to select the actual part. Use this procedure if you don't know the exact name of the part. If you know the exact name of a part, a faster procedure is to simply click on Get and then type in the name of the part. OrCAD searches through the libraries in the same order as they are listed in Figure 3-8, and retrieves the first match.

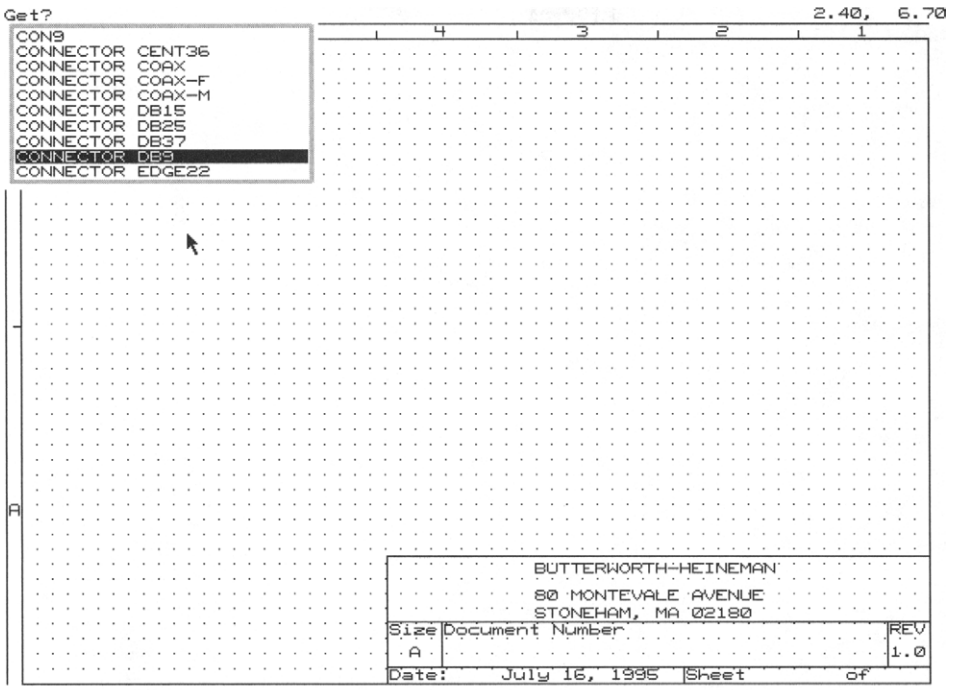


Figure 3-9 Selecting a Part From the Library

Next, orient the part so that the pins face towards the right and pin 1 is on top. Press and hold down the left mouse button. The part orientation menu shown in Figure 3-10 appears. Pull the mouse down to highlight Rotate and release the mouse button. The part is rotated 90 degrees. Repeat the process to rotate the part another 90 degrees.

In most cases the Rotate and Mirror commands will give the quickest results as far as obtaining the desired part orientation. The Normal command returns the part to the default orientation. Up, Over, and Down are sometimes useful. But depending on how the part was created, these commands do not always give intuitively obvious results.

Once you have oriented the part, the part can be positioned by moving the mouse. Move the part into the approximate position shown in Figure 1-11 and click the mouse. The part will be placed on the sheet. The Get command remains modal until you press the right mouse button to escape. You can place multiple copies of the selected part and even change the orientation. Caution: if you double click the mouse, you can inadvertently place two copies of the part in the same location.

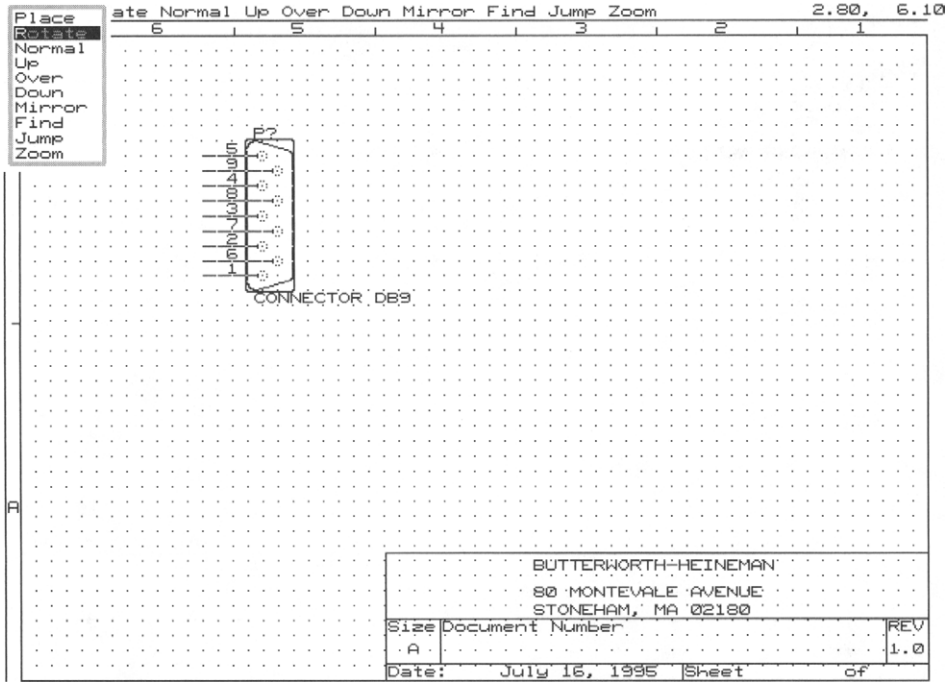


Figure 3-10 Orienting the Part

Placing Wires

Wires are the electrical interconnections between pins on parts. While most people accustomed to manual schematic drafting are used to thinking in terms of drawing lines or connections, OrCAD uses the terminology "placing wires." OrCAD uses the Place command to draw all entities other than library parts. To give valid results, wires must be drawn in strict accordance with the following rules:

- Wires must begin and end on: a part pin, module port, power or ground object, junction, or bus entry. There is one exception listed below.
- Wires must be drawn as a continuous line from start point to end point. Never draw breaks where wires cross over one another. When it is not feasible to draw a continuous wire, label objects must be associated with each segment. This will be explained in more detail in subsequent tutorials.
- Wires that cross over or intersect one another are not electrically joined unless a junction object is placed at the intersection point.

- Do not overlap wires onto pins. The wire must start at the grid location where the pin ends.
- Do not overlap segments of the same wire. If a wire is extended, start the new segment at the grid location where the last segment ends.

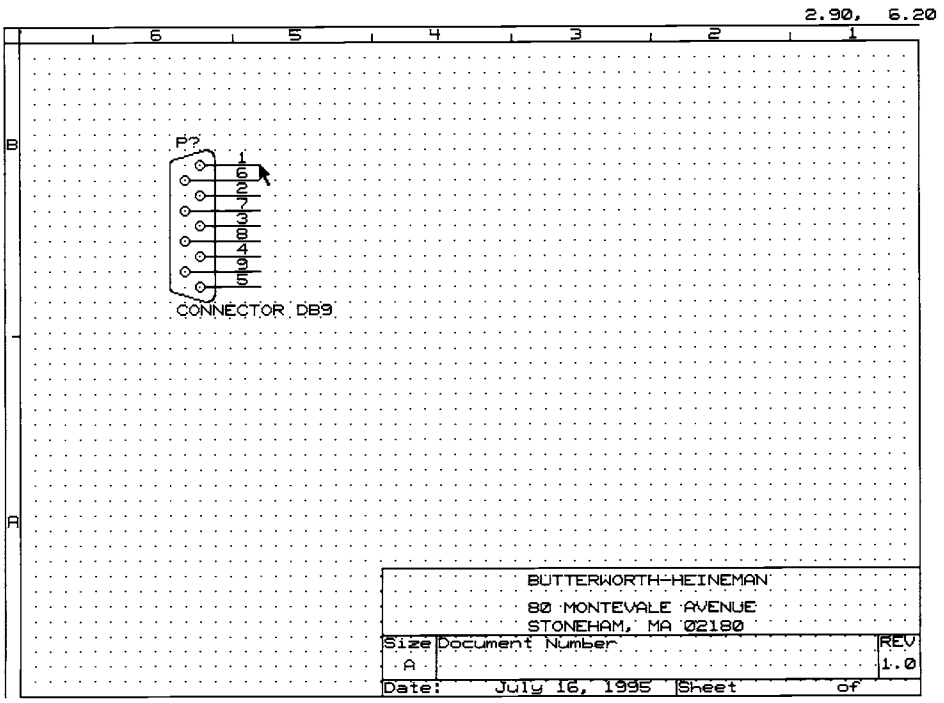


Figure 3-11 Placing the Part and Beginning a Wire

Draw the wire between pin 1 of the DB 9 connector and ground. Move the cursor to the end of pin 1. Press **F1** to start the Place Wire macro. Move the cursor to the first corner point as shown in Figure 3-12. The wire is drawn to the corner point. Click the mouse to place the corner point. Move the cursor to the end point shown in Figure 3-13 and again click the mouse to place the end point. Then click the right mouse button to escape.

As with most OrCAD commands, Place Wire remains modal until you escape. You can add any number of corner points to a wire. Another feature that you will find very useful is that OrCAD will automatically place intermediate corner points. You could have drawn the wire in Figure 3-13 by starting at pin 1 and then directly moving the cursor to the end point. OrCAD will automatically draw two orthogonal lines. Use this feature to speed up drawing long, complex wires

by clicking on every other corner point. The only limitation is that Orthogonal mode must be turned on (under the Set command on the main menu). Orthogonal mode on is the normal default.

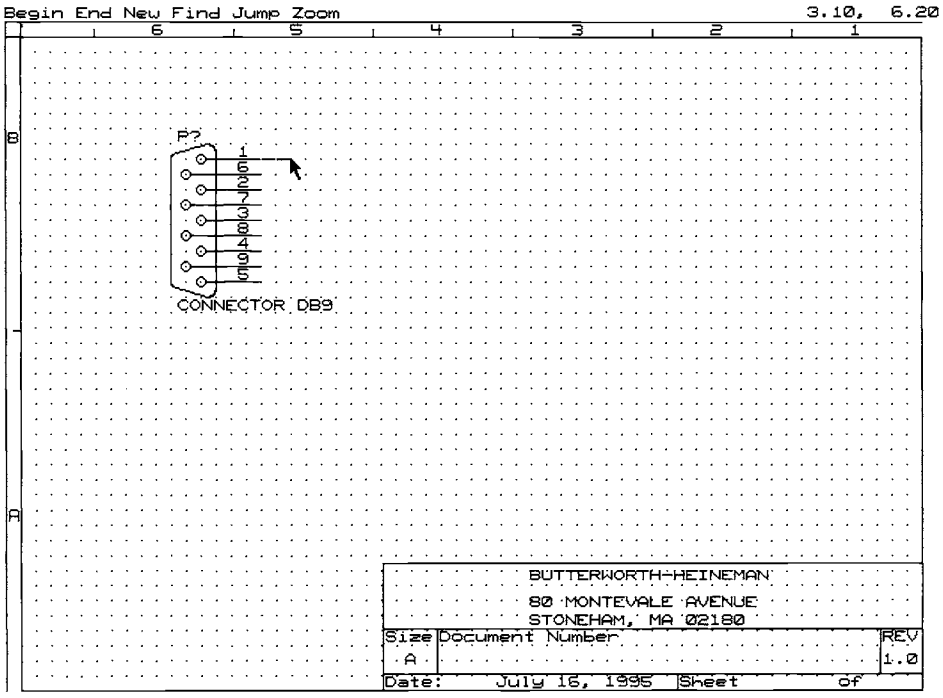


Figure 3-12 Placing a Corner Point on the Wire

Placing Ground Objects

OrCAD provides three predefined ground objects: signal ground, power ground, and earth ground. Use macro command F5 to place signal ground objects, which are the most common type. The other ground objects can be accessed via the DEVICE.LIB library using the Get command.

Place the ground as shown in Figure 3-14. Move the cursor to the end of the wire and press **F5**. The Place Ground macro places a single ground object and then exits.

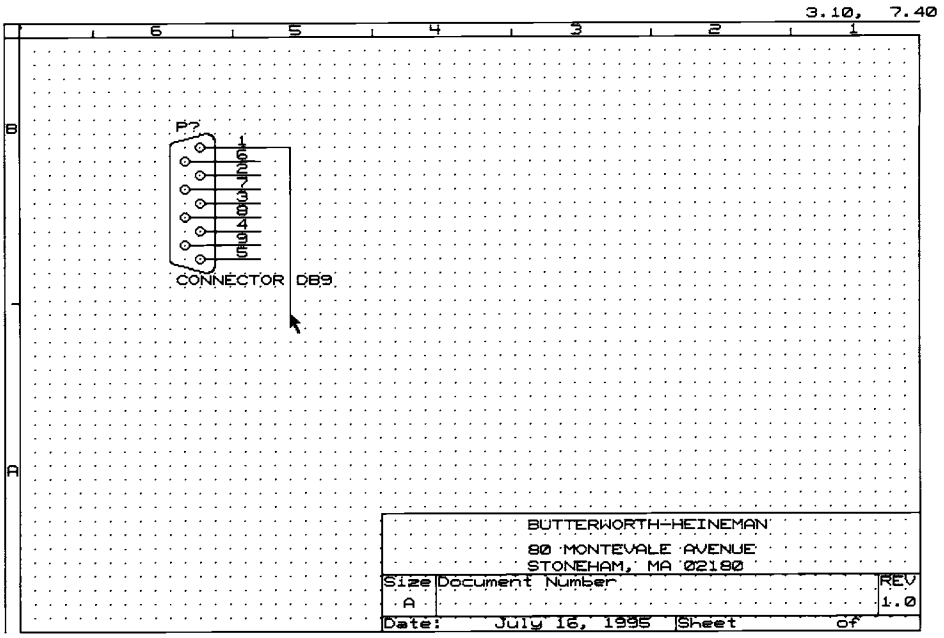


Figure 3-13 Completing the Wire

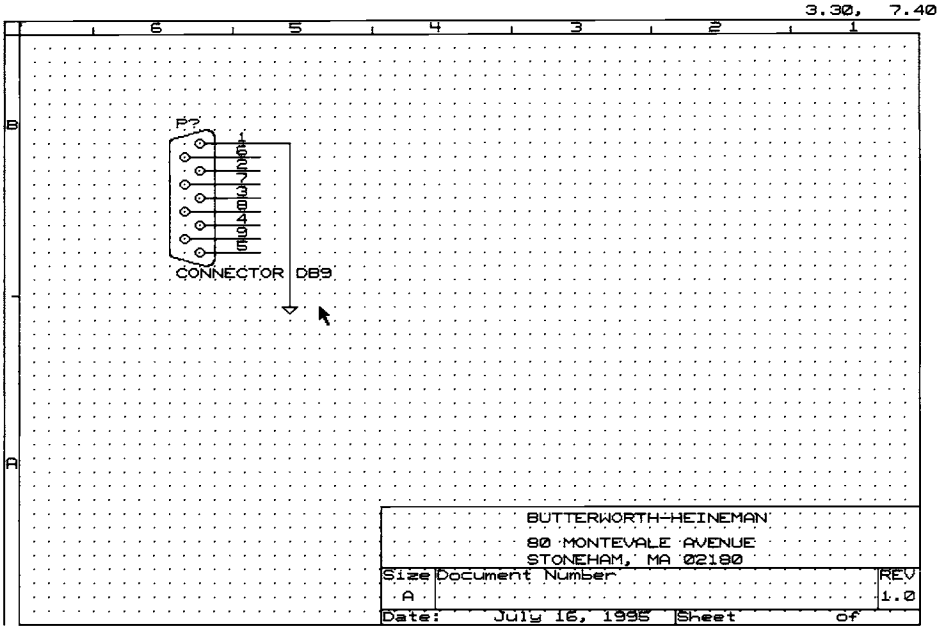


Figure 3-14 Placing a Ground Object

Placing Junction Objects

Wires that intersect are considered electrically joined only if a junction object is placed at the intersection point. OrCAD provides a junction object for this purpose. Use macro command F2 to place junction objects.

First add the two wires from pins 6 and 2 as shown in Figure 3-15. Next, place the two junctions as shown in Figure 3-16. Move the cursor to the grid location for the top junction and press **F2**. The Place Junction macro places a single junction object and then exits. Repeat the process for the lower junction.

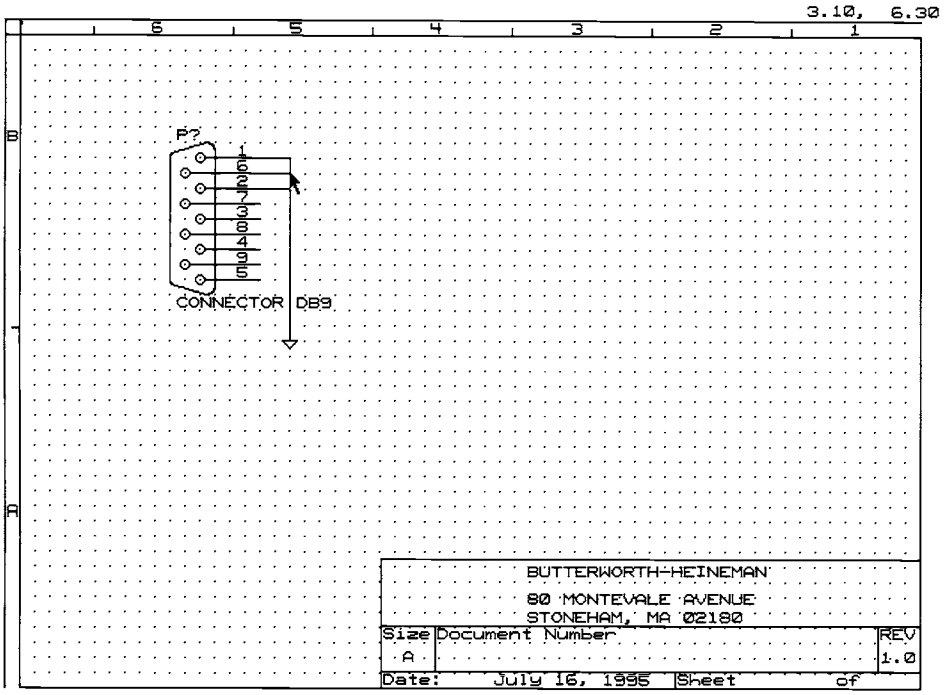


Figure 3-15 Adding Two More Wires

If you've made it so far without making a mistake, you are doing very well. Before you go on to look at more drawing functions, let's take a quick look at three very important editing functions: Delete, Move, and Drag. Let's begin with Delete.

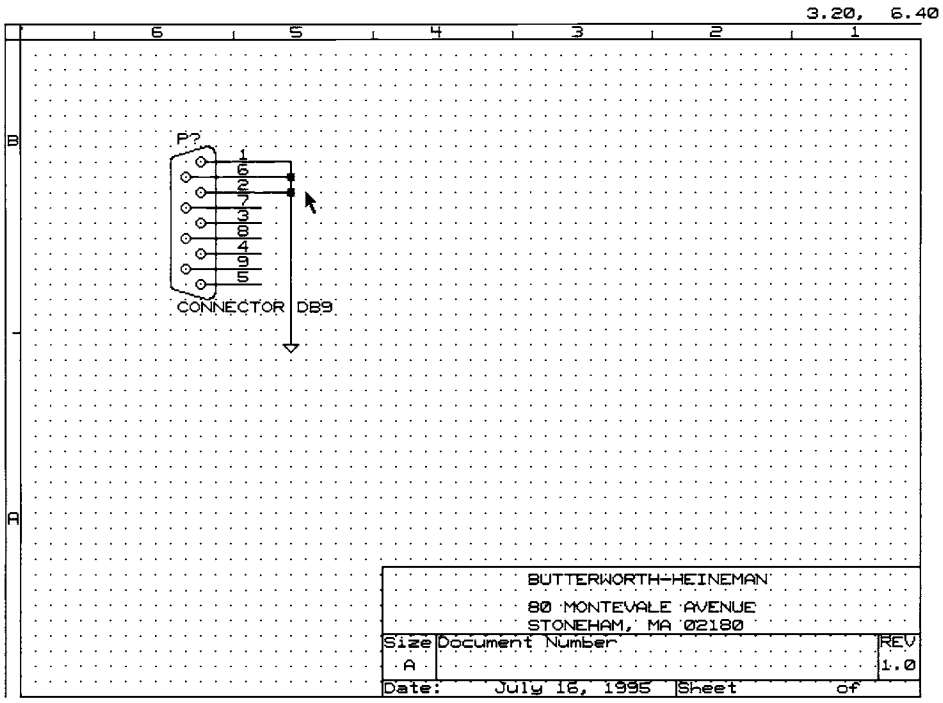


Figure 3-16 Placing the Junctions

Deleting Objects

The Delete command allows you to delete individual objects or all of the objects within a defined rectangular block. In case you make a mistake, an Undo function restores all objects deleted while the Delete command was modal. When using the basic Delete command, objects at the mouse cursor location are deleted. If more than one object is found, a submenu appears and allows you to choose which object to delete.

To practice using the delete command, delete the ground object previously placed in Figure 3-14. After you finish deleting it, use the Undo function to restore it. Refer to Figure 3-17. Move the cursor to the ground object. Start the Delete command. Click the left mouse button to bring up the main menu. Then highlight and click on Delete. Next highlight and click on Object, to select deleting individual objects as shown in Figure 3-18. Delete will find two classes of objects at the cursor location: a wire and a ground "part."

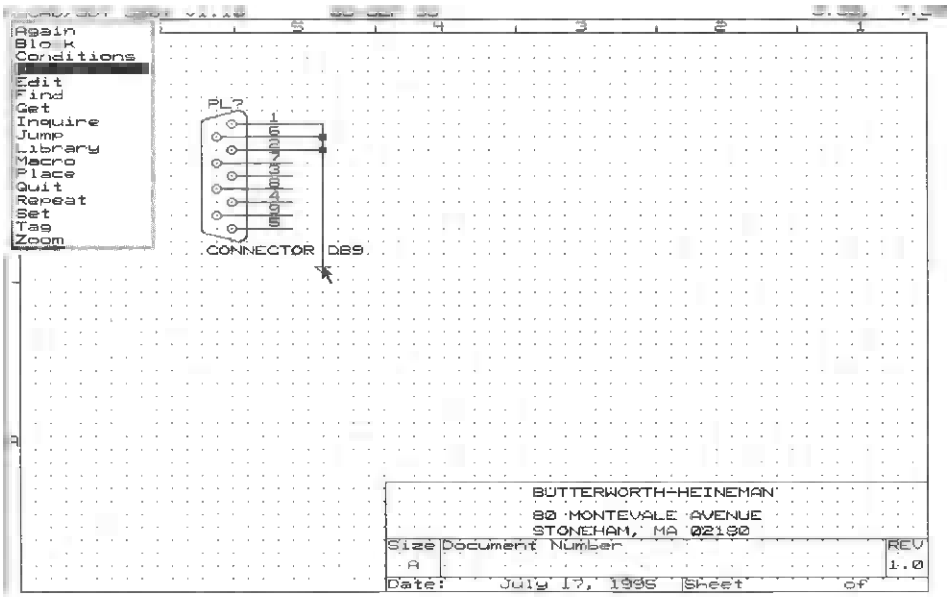


Figure 3-17 Starting the Delete Command

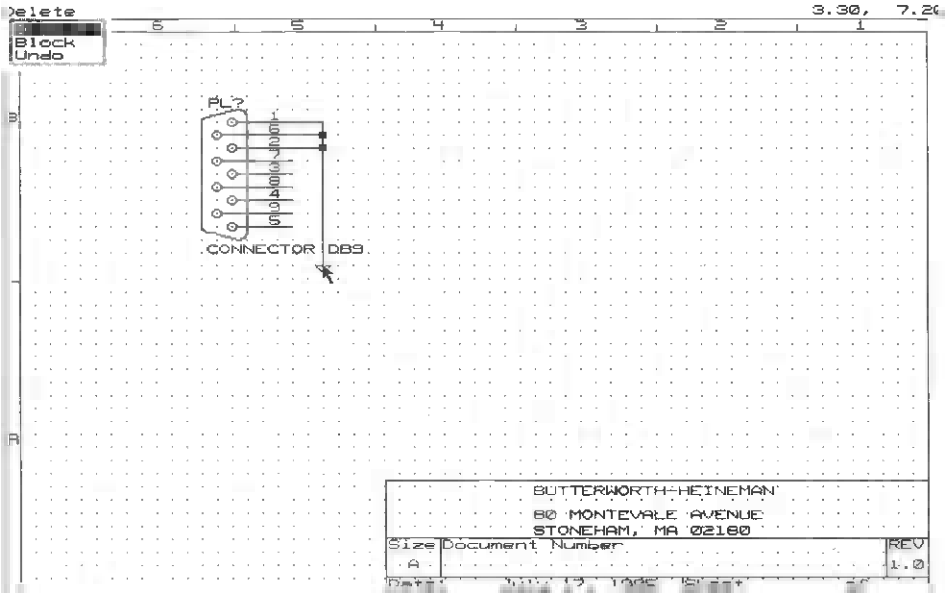


Figure 3-18 Selecting Delete Object

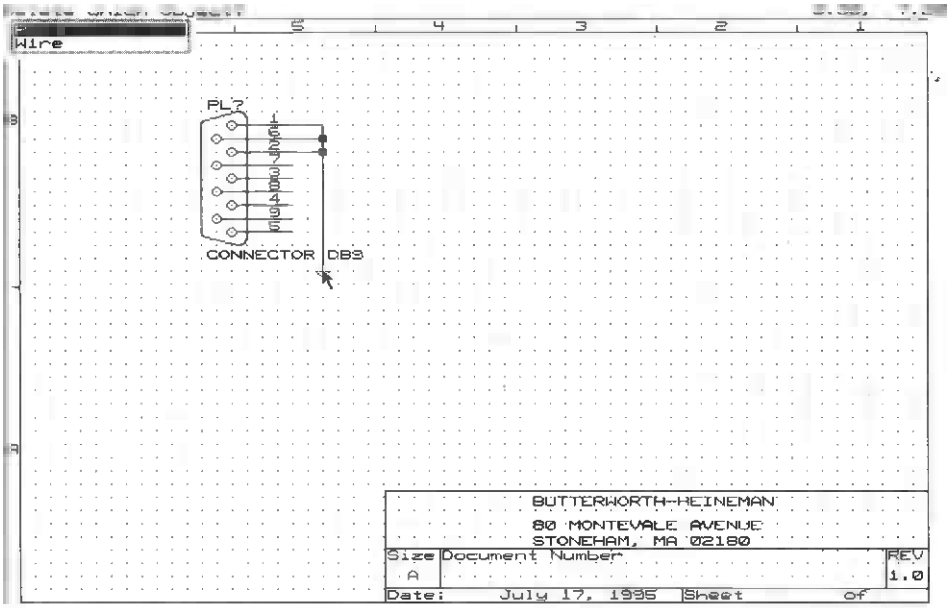


Figure 3-19 Selecting Delete Part

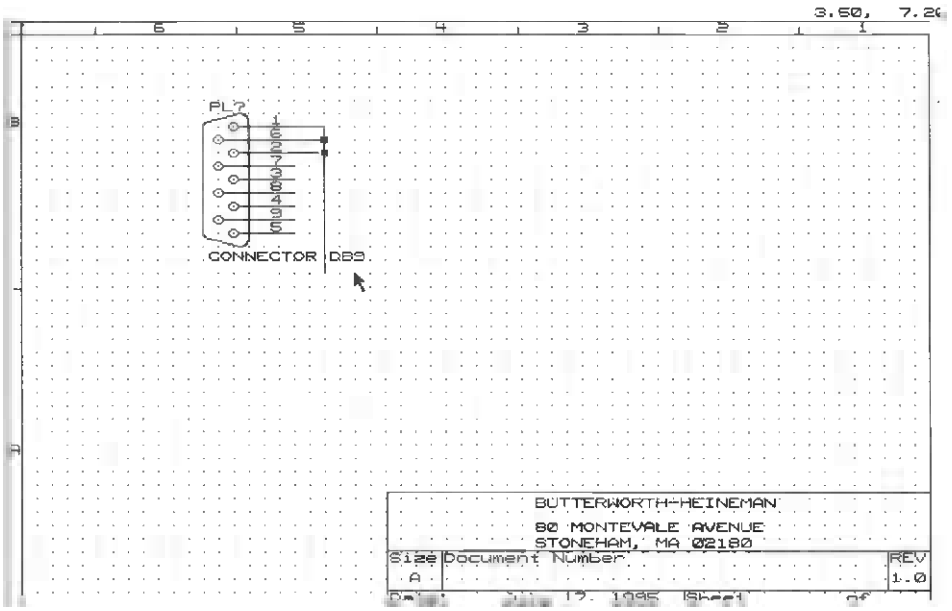


Figure 3-20 Completed Delete Operation

Highlight and click on Part as shown in Figure 3-19. The ground object is deleted and the sheet now appears as in Figure 3-20.

Because you didn't really want to delete the ground object, go back to the main menu, click on Delete again and then click on Undo. The ground object reappears.

In the next section, you will practice defining a block by dragging the mouse across a rectangular area. The Delete command also offers a block option. This is handy when doing major edits in which an entire area must be deleted.

The Block Move Command

The Block Move command will move all the objects that are within or cross a defined rectangular block. Block Move contrasts with Block Drag, which is discussed in the following section. Block Move does not stretch wires. If a wire crosses the rectangular block area, the wire segment is moved and broken off from whatever it was connected to. In many cases, you may find that this is not the desired outcome and Block Drag is more useful.

Block Drag stretches wires, which requires more processing power and graphics manipulation, especially for thick bus objects. On older PCs, Block Drag runs slowly.

Use Block Move to move the connector and wires more towards the center of the sheet. Start the Block Move Command by bringing up the main menu and clicking on Block. Then click on Move as shown in Figure 3-21. Next, use the mouse cursor to define a block area. Start at the upper left hand corner as shown in Figure 3-22. Move the cursor to this location and click the left button. Now move the cursor to the lower right hand corner as shown in Figure 3-23 and click the left button again.

You can now move all the objects in the block area to a new location by moving the mouse as shown in Figure 3-24. When you the click the left button, the objects are locked into place, the Block Move command exits, and the sheet appears as shown in Figure 3-25.

Windows users will find two aspects of this process disconcerting at first. In most Windows programs, you drag the mouse (moving the mouse while holding a button down) to define areas or move objects. The concept of "dragging" never appears in OrCAD. Also, OrCAD creates a ghost copy of the objects while the command is active and objects are being moved. This may lead you to think you are actually copying objects. When you finally click to lock the objects into place, the ghost objects go to the foreground and the original objects disappear.

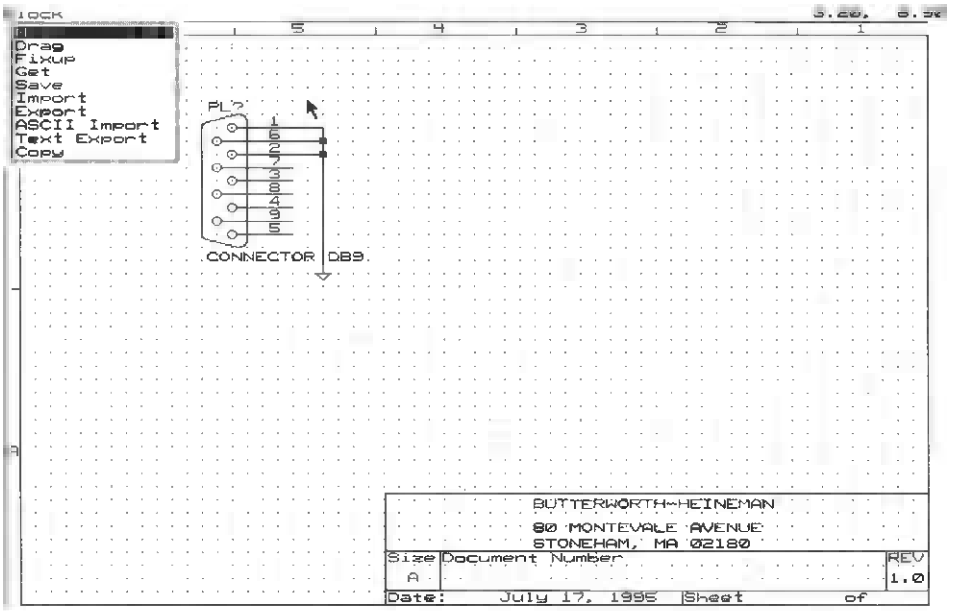


Figure 3-21 Starting Block Move

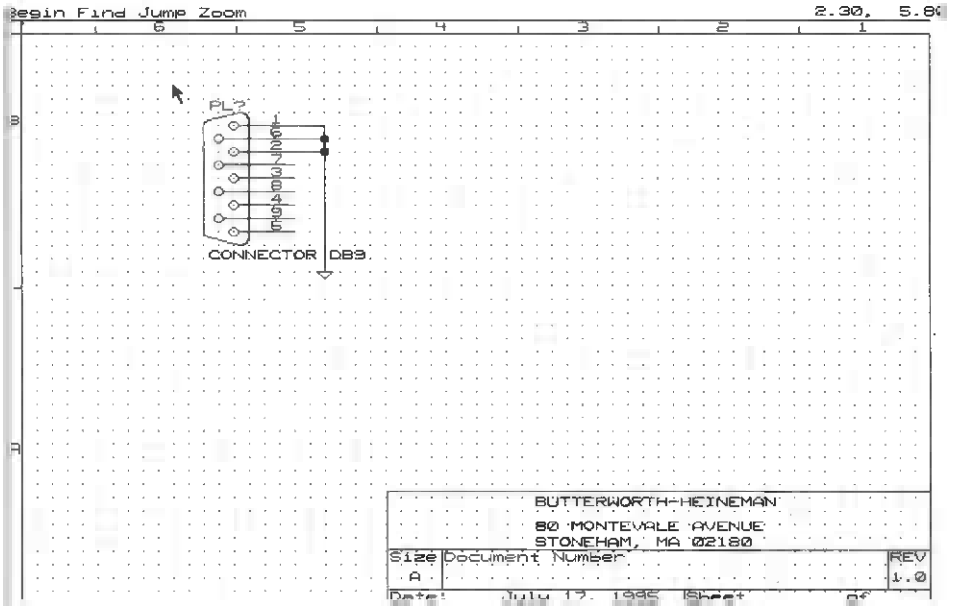


Figure 3-22 Defining the First Corner of the Block

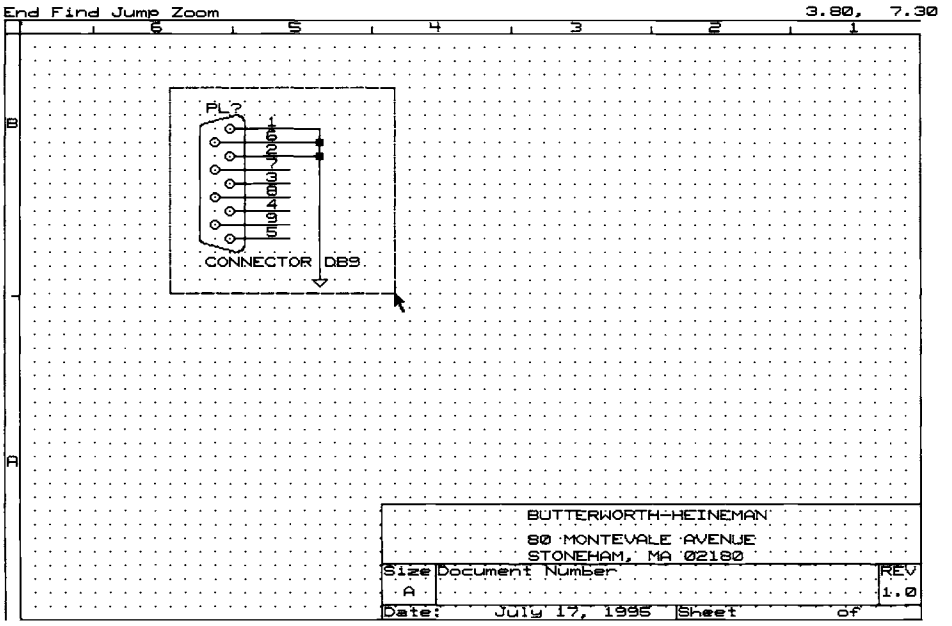


Figure 3-23 Defining the Block Area

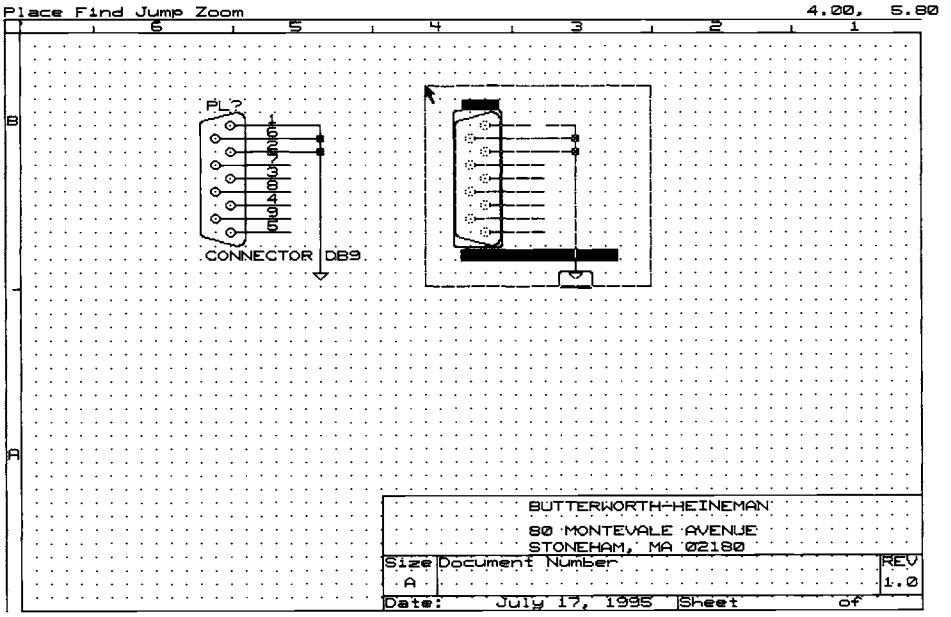


Figure 3-24 Moving the Block Area

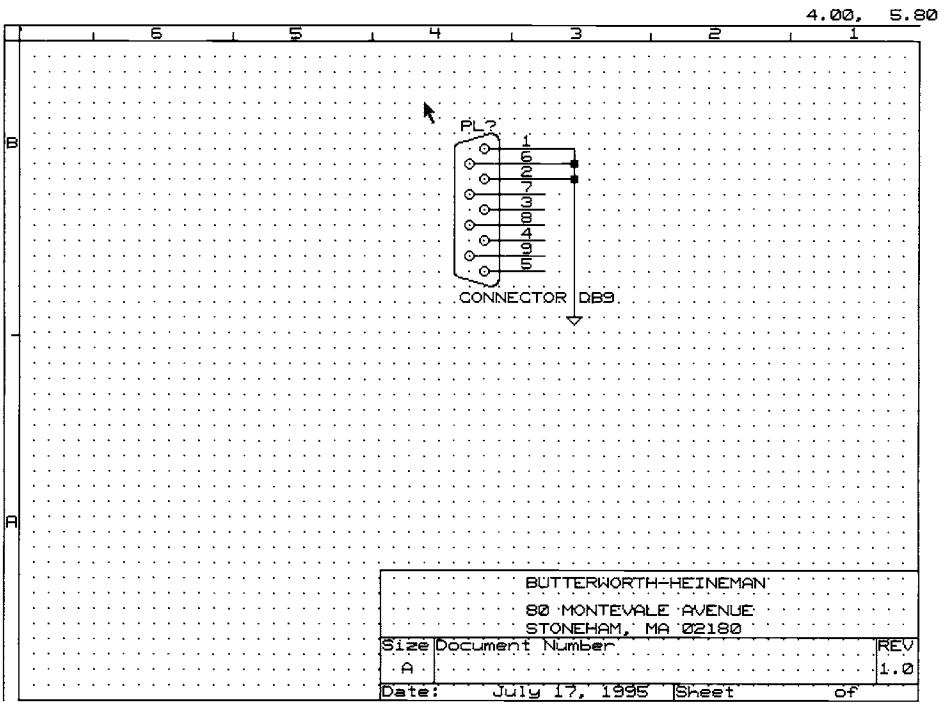


Figure 3-25 Completed Block Move

The Block Drag Command

The Block Drag command is usually preferable to Block Move. Any wires and busses (if Drag Busses is enabled under Setup) that cross the block area are stretched as required. You will use the Block Drag command for most editing that requires moving objects from one area to another on the sheet.

Block Drag functions similar to Block Move. Use Block Drag to stretch the ground wire away from the connector as shown in Figures 3-26 through 3-30. First launch the Block Drag command, define the corner points of the block areas, and then drag the block.

Please note that the left mouse button is not held down while dragging the objects. Once the command is launched, three mouse clicks are required. Two to define corner points of the block area and a third to define the final location that the block is dragged to.

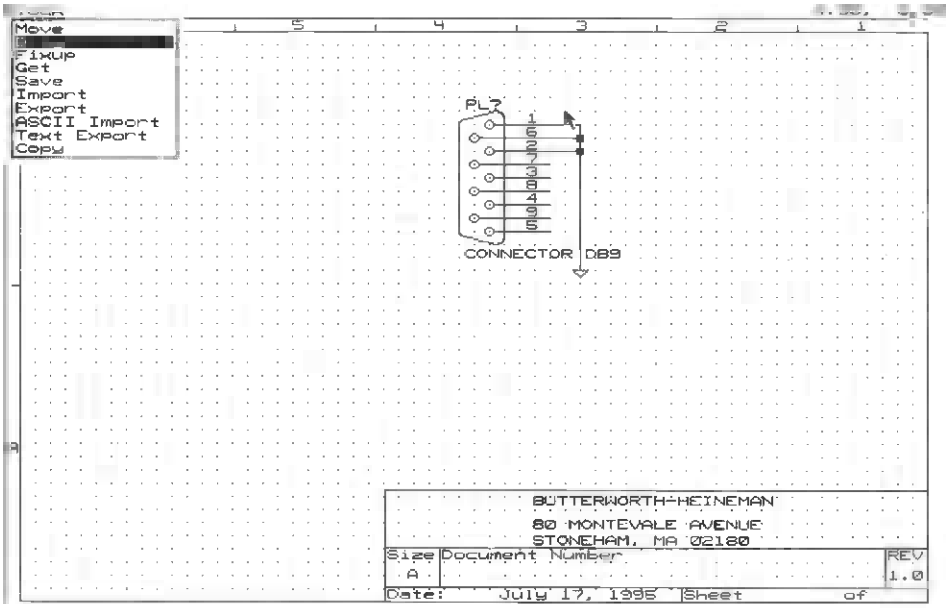


Figure 3-26 Starting the Block Drag Command

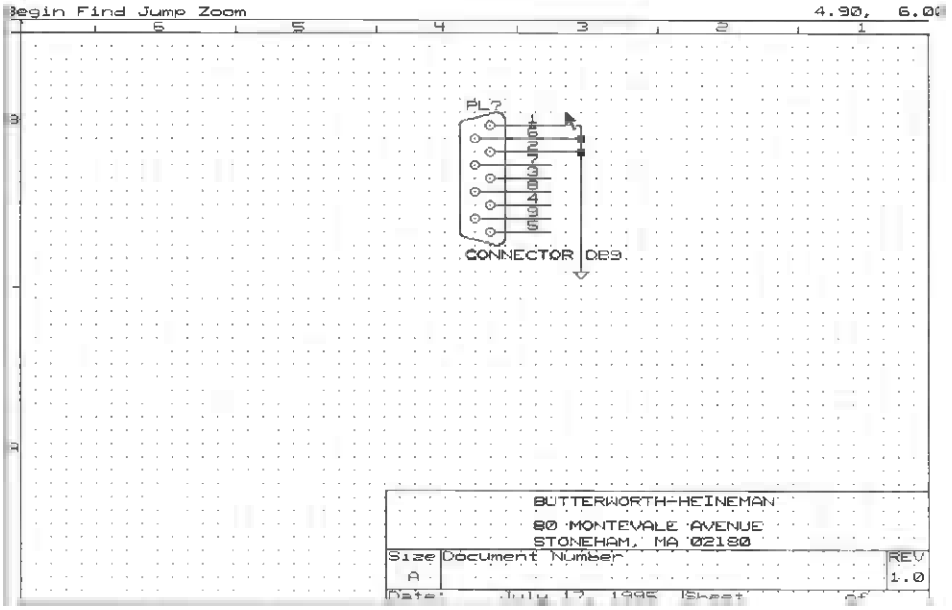


Figure 3-27 Defining the First Corner Point

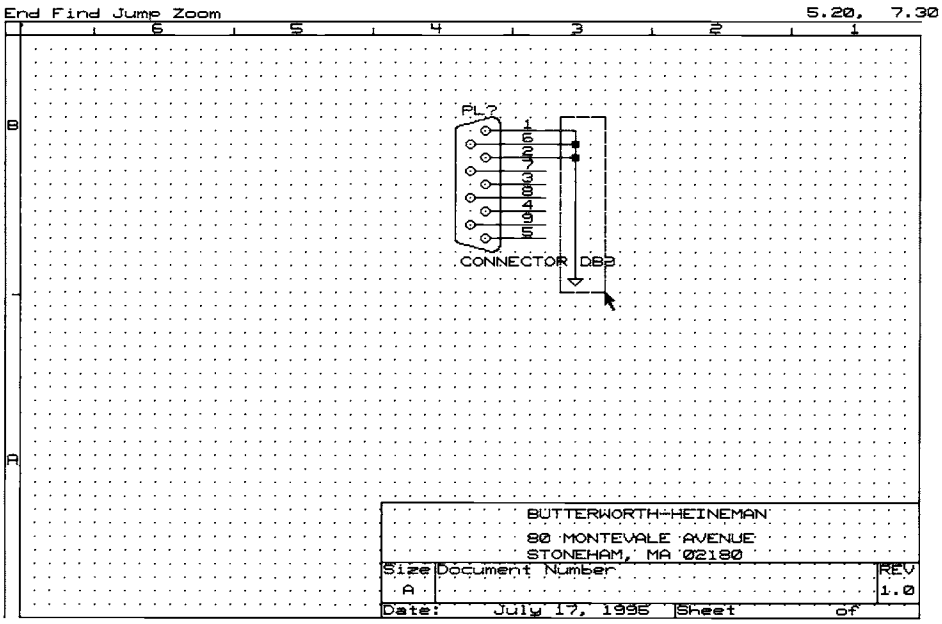


Figure 3-28 Defining the Second Corner Point

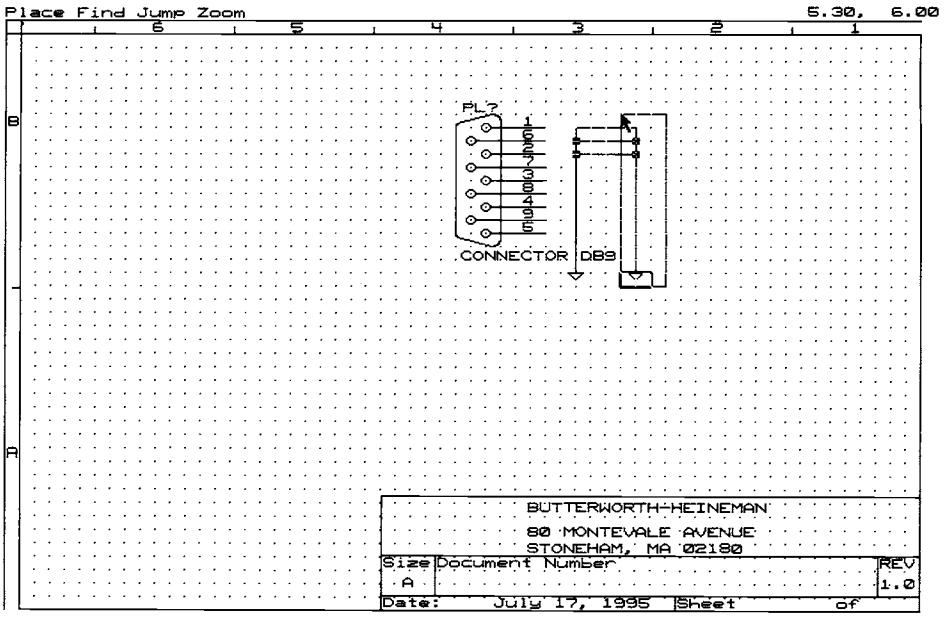


Figure 3-29 Dragging the Objects to a New Location

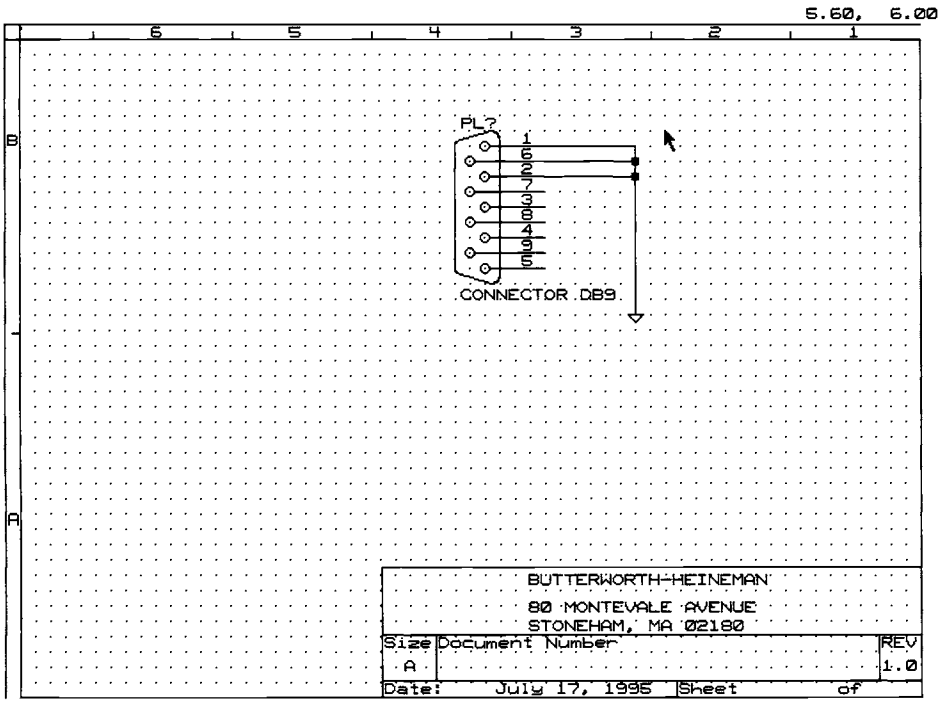


Figure 3-30 Completed Block Drag Command

Completing the Schematic

You have now learned how to get parts from the library and place them on the sheet, draw wires and junctions, place ground objects, and perform basic editing operations including delete, move, and drag. Continue on and complete the schematic. First, use the Drag command to move the ground connections back towards PL1. Then use Drag to move all the objects you have drawn so far back towards the left side of the sheet.

Next you will learn how to place resistors, capacitors, and power objects.

Placing Resistors, Capacitors, and Power Objects

Place an electrolytic capacitor by using macro function F8. Press **F8** to make the electrolytic capacitor shown in Figure 3-31 appear. The symbol moves with the mouse cursor. Note the menu at the top of the Figure. If you press the left mouse button you can highlight and select one of the menu options. When you release the left mouse button, the highlighted menu selection is executed. This technique would be used to select the Rotate or Mirror options for orienting parts prior to

placing them on the sheet. If you just click the left mouse button, the first menu selection, Place, is executed and the capacitor is placed at the cursor location. The command remains modal until you hit the right mouse button to escape.

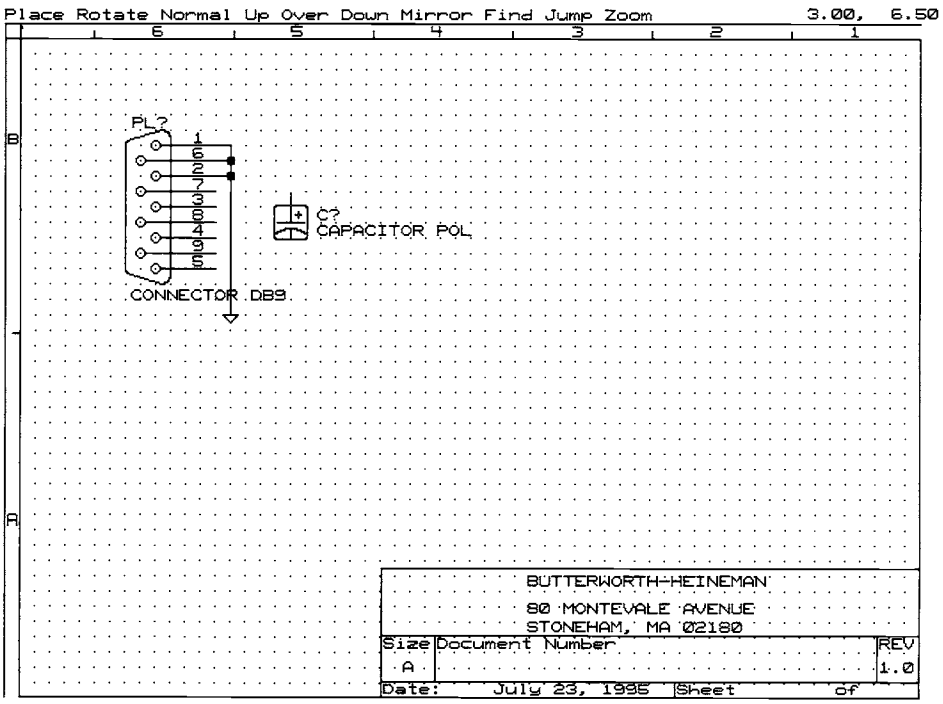


Figure 3-31 Placing a Capacitor

Figure 3-32 shows the sheet after the capacitor has been placed. Recall that macro F8 is used for electrolytic capacitors and F7 for nonpolarized capacitors.

The next step is to place the +24V power object. The CUSTOM.MAC macro commands include +5V and +12V power objects. You can use either one of these and later edit the description. Figure 3-32 shows the +5V power object, which will be edited to become +24V later on in the tutorial. Position the cursor where you want to place the power object and press **F3**. Use macro command F3 again to place the +5V power object. Note that F3 and F4 (for a +12V power object) automatically exit after the object is placed.

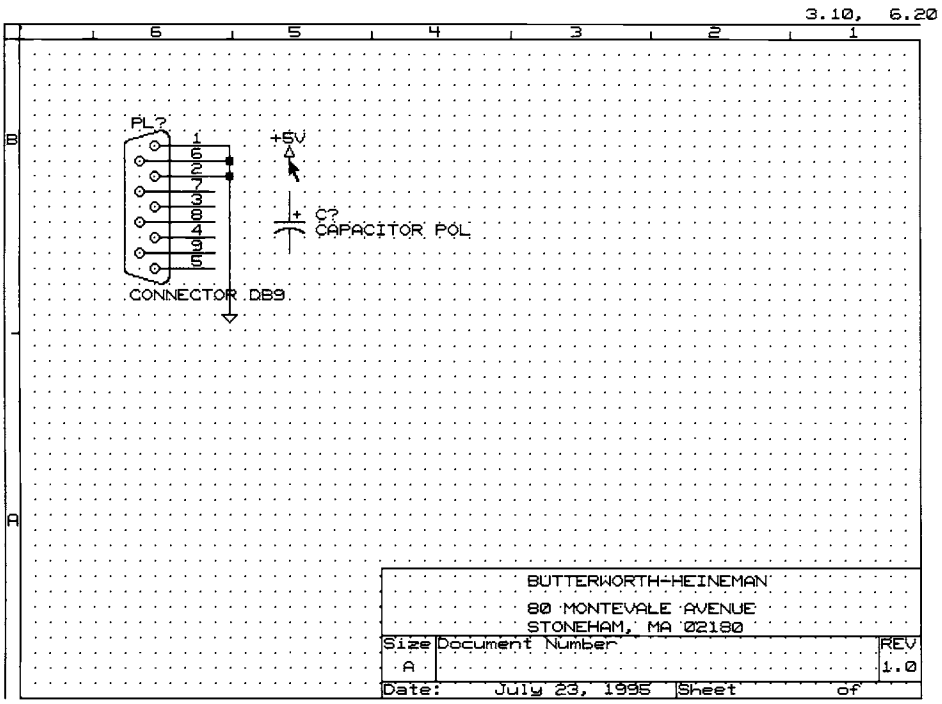


Figure 3-32 Placing +5V Power Object Using F3

Next, use macro command F6 to place a resistor as shown in Figure 3-33. Then place the second resistor as shown in figures 3-34 and 3-35. Rotate the resistor before placing it on the sheet. Hold the left mouse button down, highlight the Rotate option, and release the mouse button to rotate the resistor. Then position the resistor and click the left mouse button again to place it.

Finally use macros F1 and F2 to place wires and junctions and complete the section of the schematic as shown in Figure 3-36.

Placing and Interconnecting the Remaining Parts

By this point, you should be relatively proficient at getting and placing parts, wires, junctions, power and ground objects. Your next task is to complete the placement and interconnection of the remaining parts. After you are done, the following sections will show you how to edit part descriptions, place text, and print out a hard copy of your first OrCAD schematic.

Here are some hints on placing the remaining parts:

- **Resistors.** Use macro F6. Some of the resistors will have to be rotated.
- **Capacitors**. Use macro F7 for the nonpolarized part,C3; and macro F8 for C4. Rotate C3 before placing it.
- **Operational amplifier U1**. The LM324 is in the parts library and OrCAD will automatically find it if you use the Get command and type in LM324.
- **Molex connector PL2**. Use the Get command and type in CON5.
- **Transistor Q1**. Use macro F10 to place the NPN transistor.
- **+24V power object**. Use macro F3 and place a +5V power object. Later, you will edit the description to make it +24V.
- **Wires and junctions**. Use macros F1 and F2.

If you make a mistake, just use the Delete command and start over. If you place a part in a location that is not quite right, use Block Drag or Block Move. Don't worry about precisely locating every part or wire in exactly the same place as the model in Figure 1-17. Use the model as a guideline and follow the general flow.

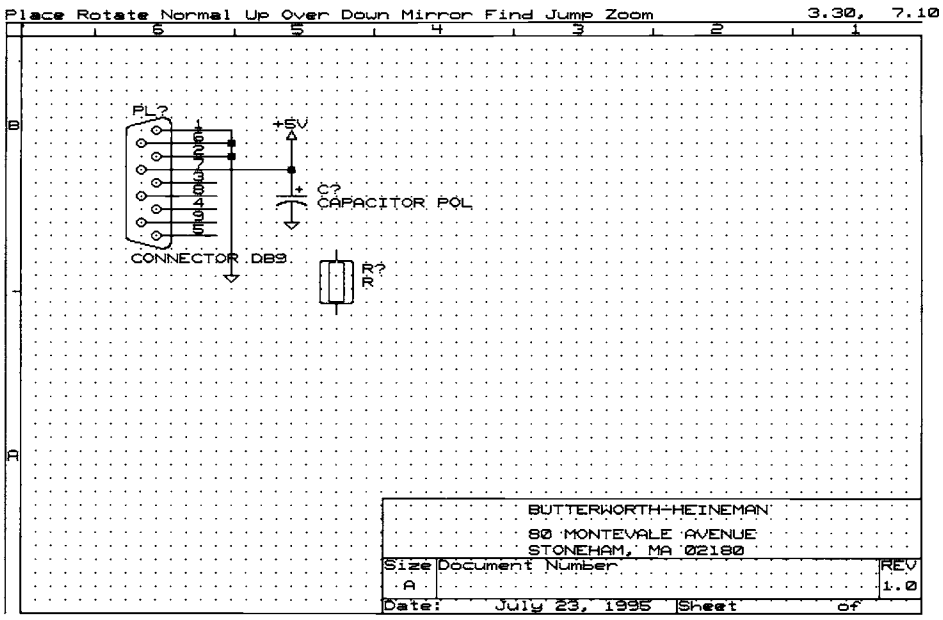


Figure 3-33 Placing the First Resistor Using F6

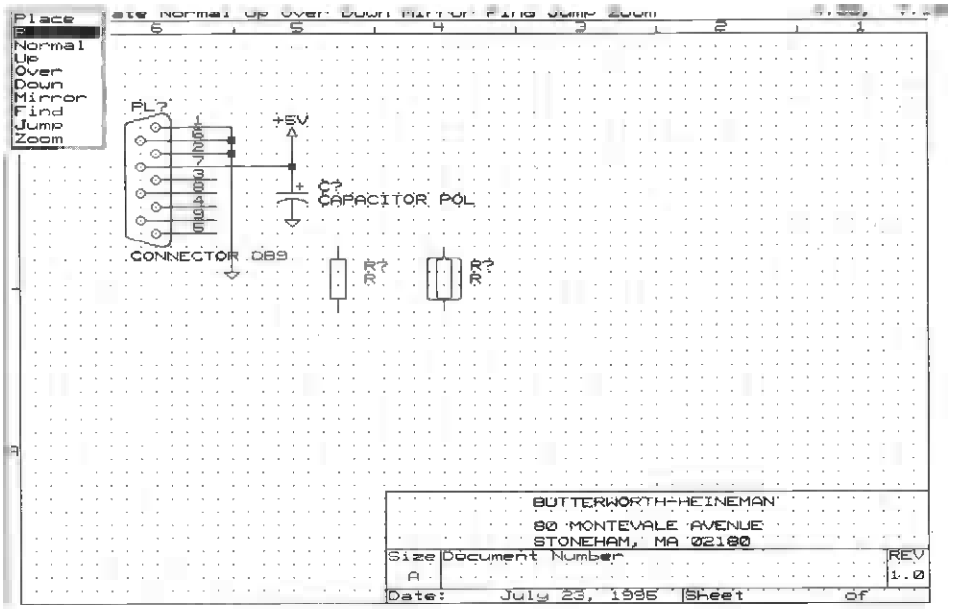


Figure 3-34 Rotating the Second Resistor

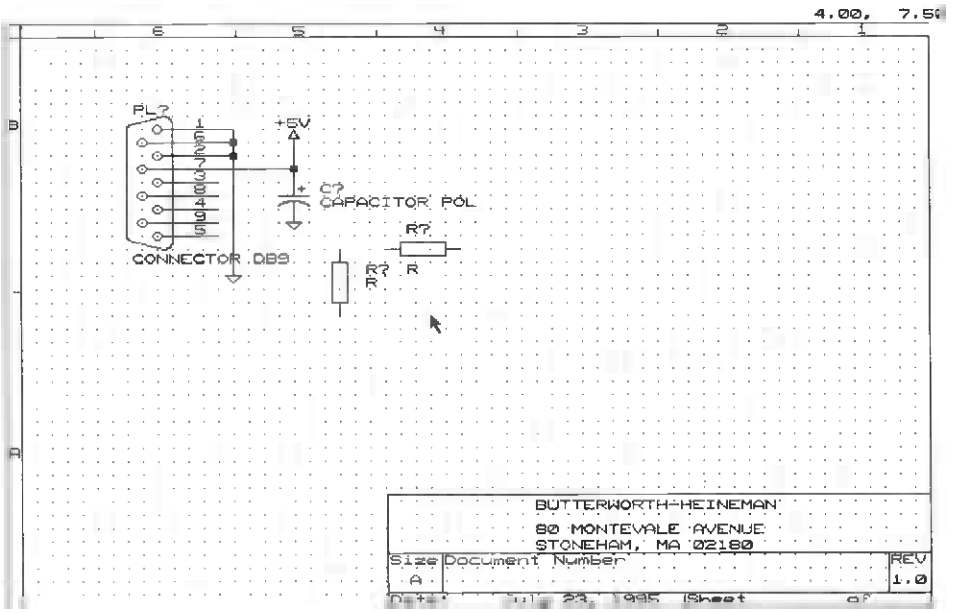


Figure 3-35 Placing the Second Resistor

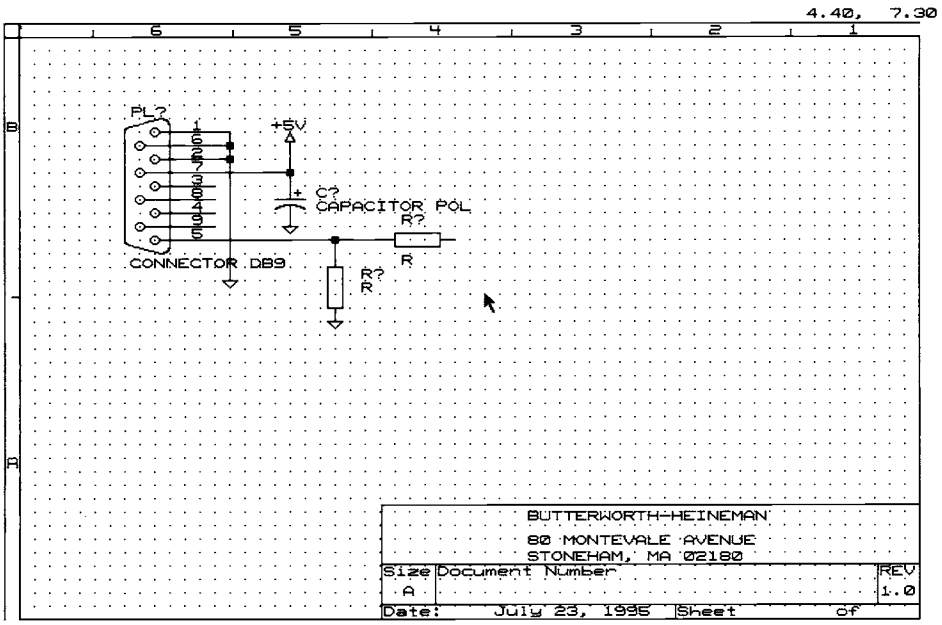


Figure 3-36 Completing Wires to the Resistors

Editing Reference Designators and Part Values

At this point you have placed and interconnected all the parts on the schematic. Your screen should look like Figure 3-37. The screen appears somewhat messy because reference designator and part value text overlaps. The next step will be to annotate the schematic. Annotation refers to editing reference designators and part values (descriptions). The annotation process involves four steps for each part:

- **Edit the reference designator**. In most cases the alphanumeric prefix, such as R for a resistor, stays the same and only the numeric suffix is changed. OrCAD reference designators default to a "?" suffix when the part is initially placed. Note that the 5 pin Molex connector at the right of the sheet will require editing the prefix from J to PL. OrCAD also provides a postprocessing tool that automatically numbers reference designator suffixes, based on the order that the parts were placed on the sheet. This tool will be reviewed in the next tutorial.
- **Edit the parts value**. This involves entering a part number or other descriptive data. Refer back to chapter 1 for further details. In this first tutorial, only the part value field is used, which limits the part description to

a single line. OrCAD allows multiple part description fields. You will learn to use this feature in the next tutorial.

- **Locate the reference designator**. Sometimes the reference designator can be left in the default location, but in most cases you will have to move it to improve legibility of the schematic. Group the reference designator with the part value near the part.
- **Locate the part value**. The general rule is that the reference designator should be located on top or to the left of the part value. For narrow parts such as resistors, an acceptable practice is to locate the reference designator directly above the part and the value directly below the part.

The reason for following the sequence given above is that the length and height occupied by part descriptions varies. You will find it much easier to locate the parts description last, when you know how much space is required. Neat and carefully placed annotations improve the readability of the schematic and reduce the possibility of confusion. This becomes an important issue on complex and crowded schematics.

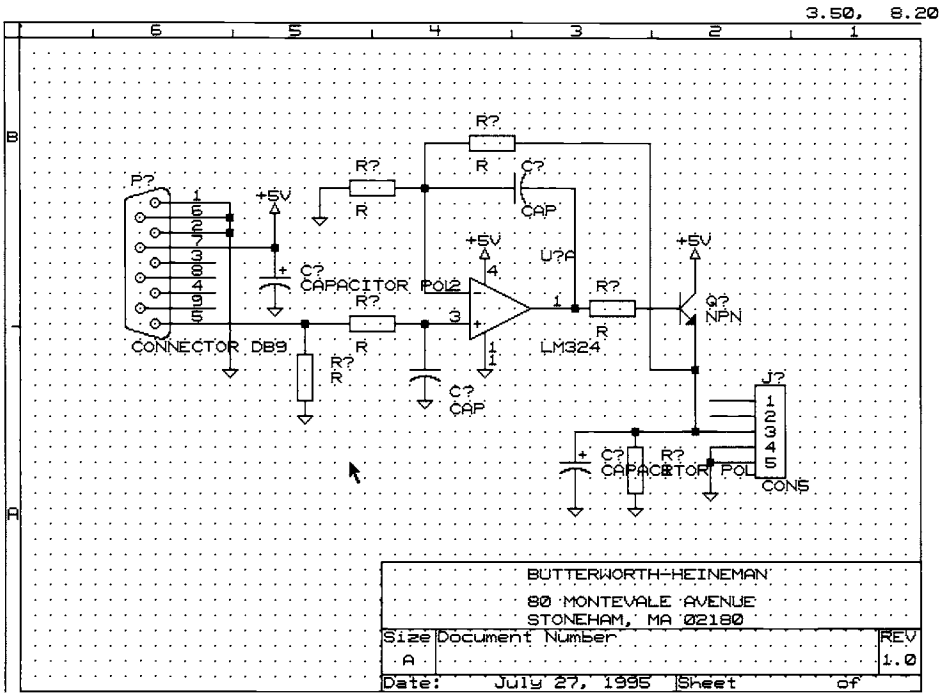


Figure 3-37 Schematic Prior to Annotation

Practice on R3. Start by editing the reference designator. Click the left mouse button to bring up the main menu, then highlight and click on Edit as shown in Figure 3-38. Next use the mouse cursor to select the part to be edited. Move the cursor to R3 as shown in Figure 3-39 and click the left button. This action brings up the parts edit submenu shown in Figure 3-40. You will use this same sequence whenever you are editing any part or object. Note that the submenu provides access to all properties that can be edited including the reference designator, part value, optional part fields, and the part orientation. Highlight and click on reference designator. This brings up another submenu shown in Figure 3-41.

The submenu presents a choice of actions: edit the name (text), change the location, or make the reference designator visible or invisible. Identical submenus appear for editing the part value or optional part fields. In most cases the reference designator and part value are kept visible. Optional part fields could be used for data such as manufacturer's part number. This type of data is useful for generating bill of materials, but would tend to clutter a crowded schematic and might be left invisible.

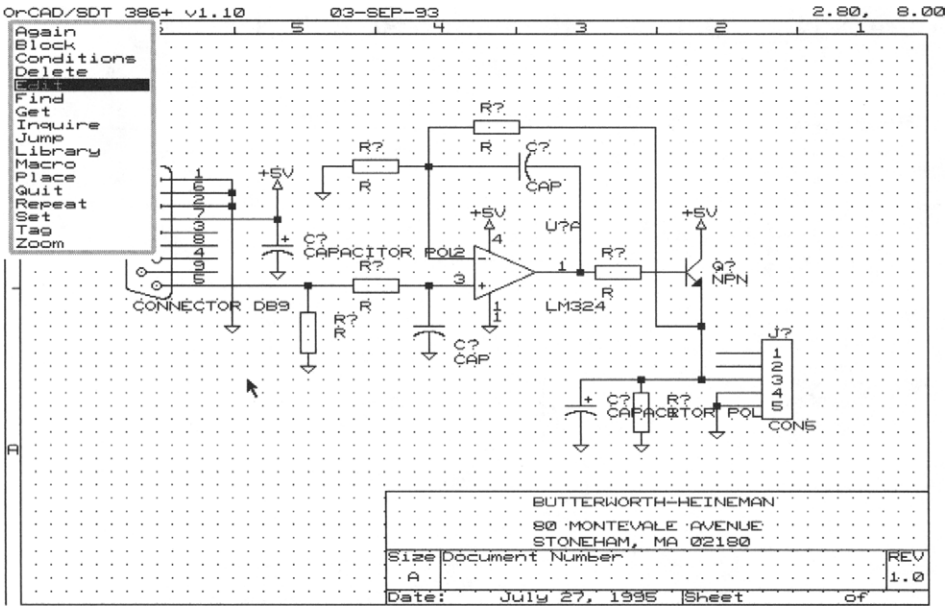


Figure 3-38 Selecting the Edit Command

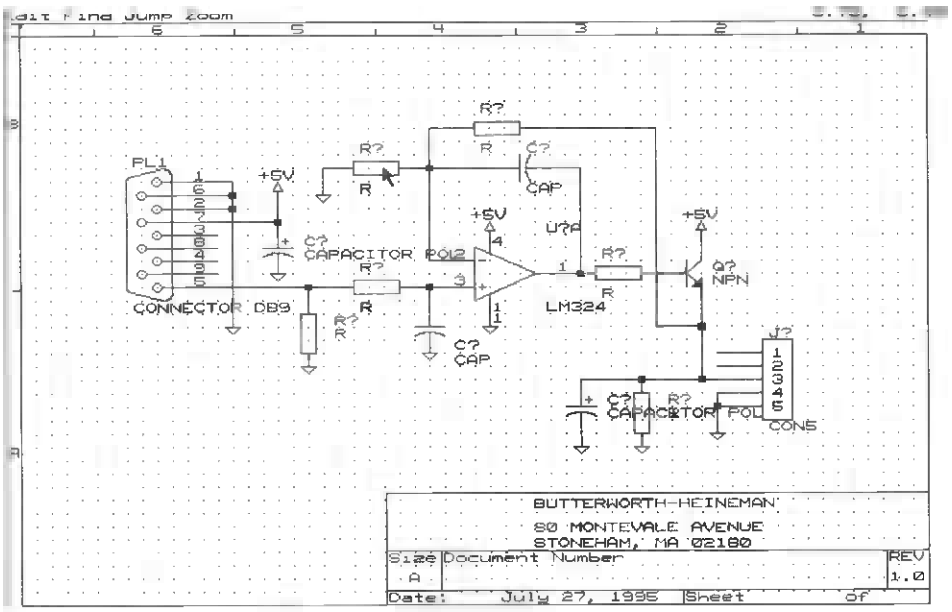


Figure 3-39 Selecting the Part to Be Edited

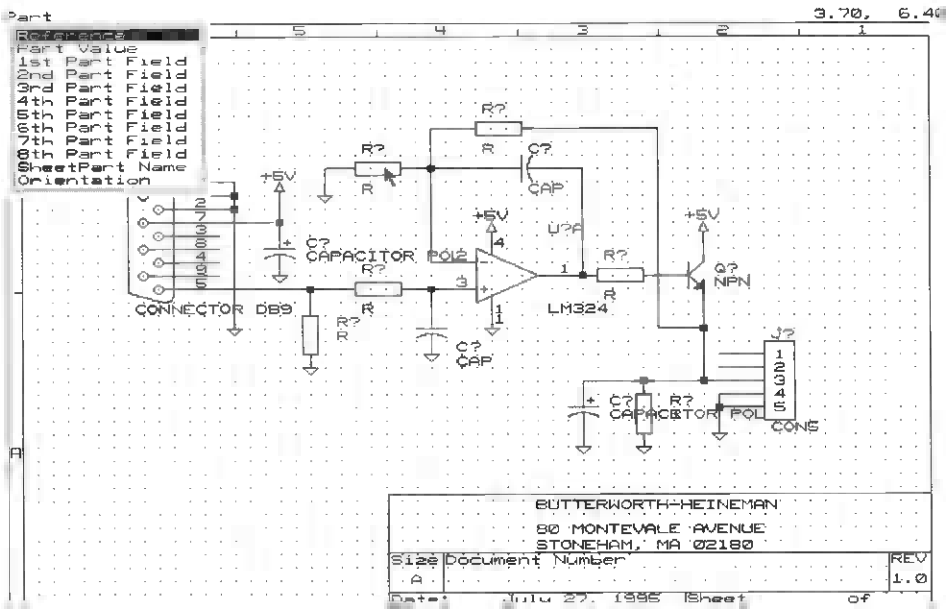


Figure 3-40 Selecting Edit Reference Designator

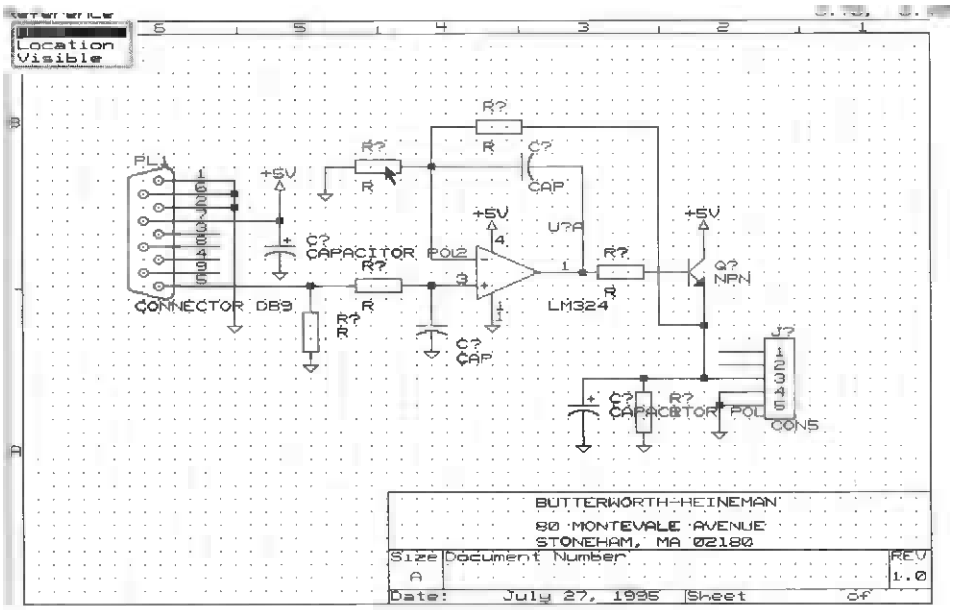


Figure 3-41 Selecting Edit Reference Designator Name

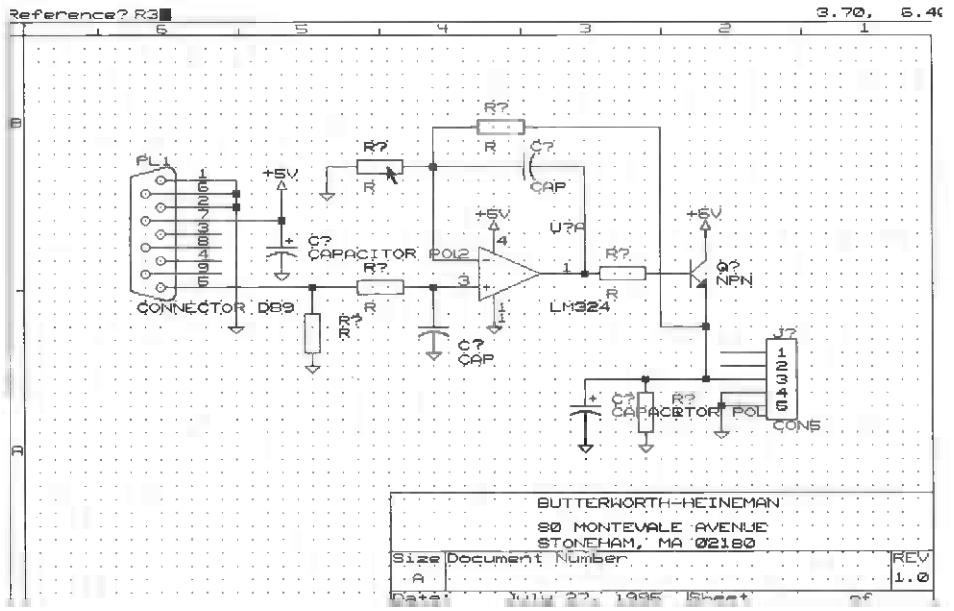


Figure 3-42 Entering the Reference Designator

Highlight and click on Name as shown Figure 3-41 to edit the reference designator text. The default reference designator appears as "R?". Refer to Figure 3-42. Use the backspace key to erase the **?**, then type in a **3** and press **<ENTER>.**

Repeat the same sequence to edit the part value. Highlight and click on Edit from the main menu and click on R3 again. This time you will edit the part value, so highlight and click on Part Value as shown in Figure 3-43. Then highlight and click on Name as shown in Figure 3-44 to edit the part value text. Refer to Figure 3-45. Use the backspace key to erase the existing text, type in the new description **10.0K 1%** and press **<ENTER>**.

This completes the text editing for R3. The last step is to move the reference designator and part value to the location used with the model. Since the reference designator is on top, move it first. Repeat the basic sequence for editing, but highlight and click on Reference Location as shown in Figure 3-46. Similar to Block Move, OrCAD creates a ghost copy of the reference designator text. Use the mouse cursor to position it in the desired location as shown in Figure 3-47 and then click the left mouse button. Now repeat this same process for the part value and locate the part value just below the reference designator.

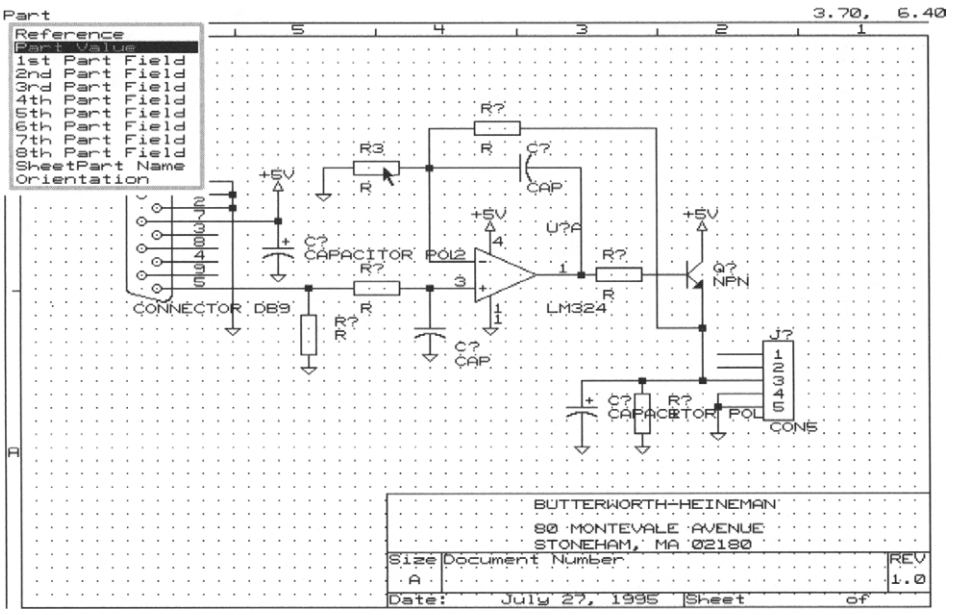


Figure 3-43 Selecting Edit Part Value

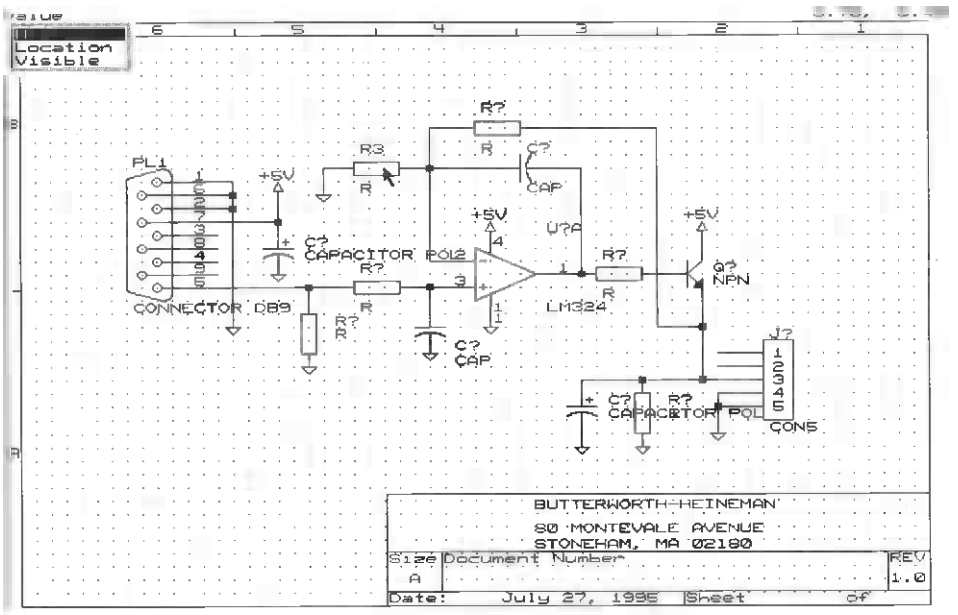


Figure 3-44 Selecting Edit Part Value Name

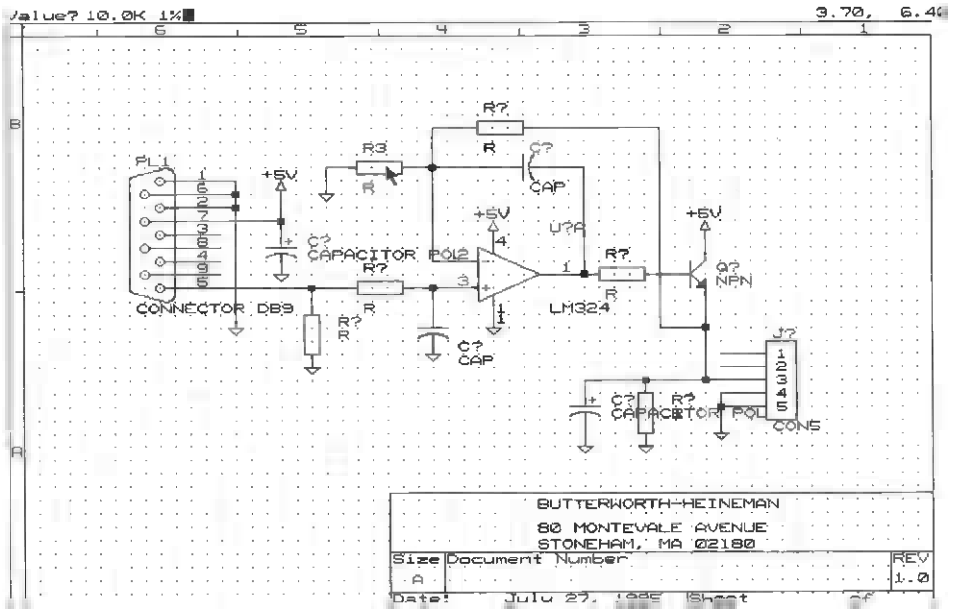


Figure 3-45 Entering the Part Value

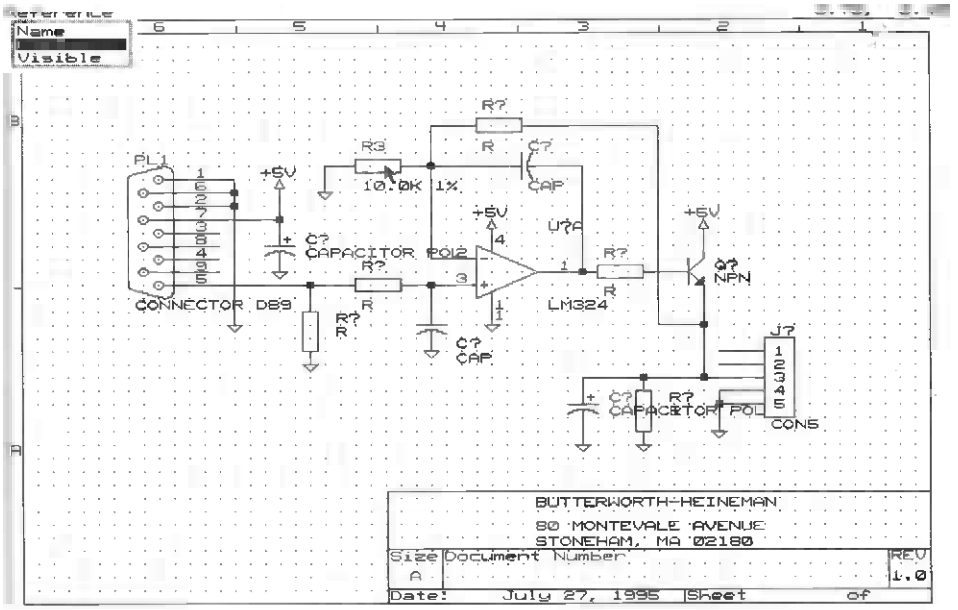


Figure 3-46 Selecting Edit Reference Location

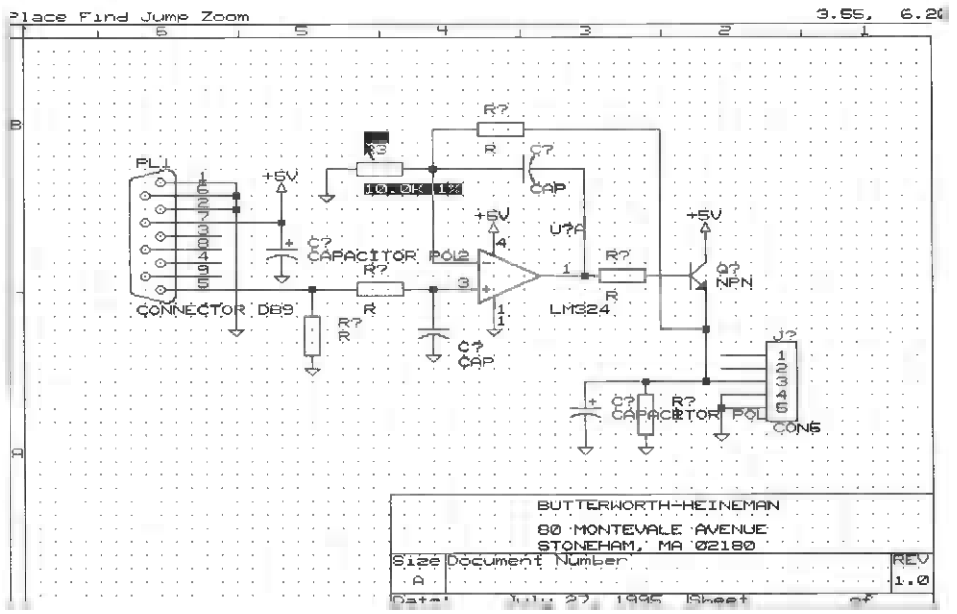


Figure 3-47 Changing the Reference Location

Editing Power Object Descriptions

Unlike ground objects, OrCAD does not provide any predefined power objects, only a set of standard power symbols, such as the arrow used on the tutorial model. Macros F4 and F5 place predefined +5V and +12V power objects. When other power levels are required, the descriptive text (name) associated with the power object can be edited. Caution! You must use precise names. For example, if a design has a +12V power plane, power objects named +12, 12V, +12VOLTS, or +12 V would be not be associated with the +12V plane.

Edit the three power objects on your sheet to make them +24V as on the tutorial model. Bring up the main menu, highlight and click on Edit, and then click on one of the power objects. The Edit Power menu shown in Figure 3-48 appears. Highlight and click on Name.

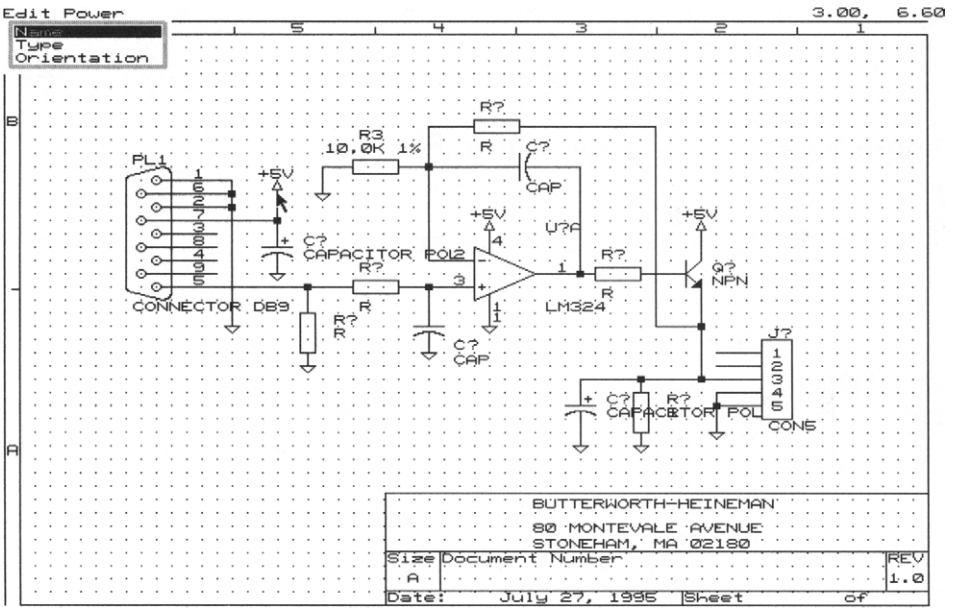


Figure 3-48 Selecting Edit Power Name

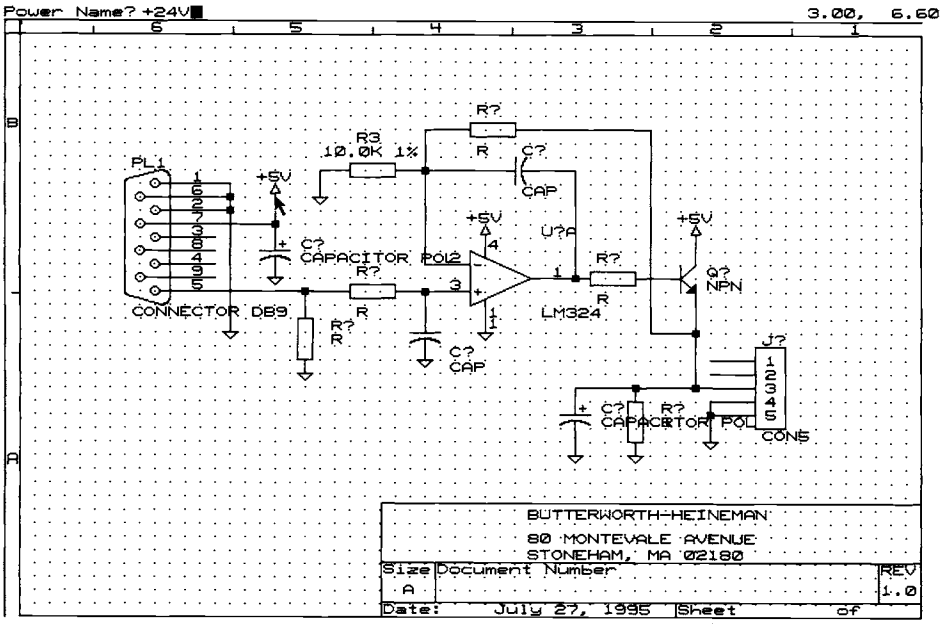


Figure 3-49 Power Object Name Entry

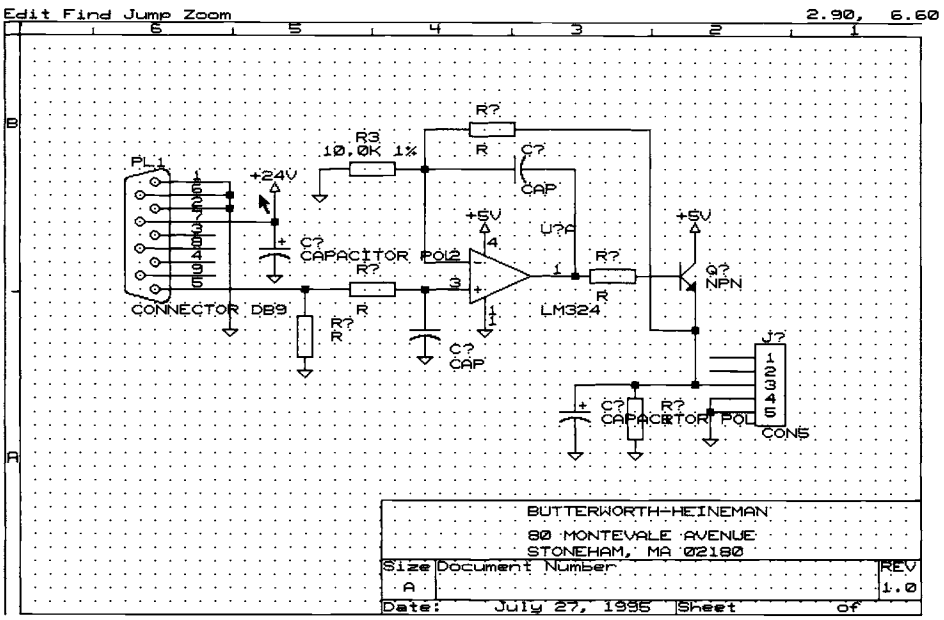


Figure 3-50 Completed Power Object after Edits

Backspace to delete the existing +5V description. Type in **+24V** and press **<ENTER>,** as shown in Figure 3-49. Figure 3-50 shows the +24V power object after completion of the edit. Edit the other two power objects on the sheet in a similar manner. Then edit the remaining parts using the model as a guideline.

Editing the Title Block

You can use the Edit command to edit the title block, just as any other OrCAD part or object. When a new design is created the title block area is initially filled with whatever default information was defined during the configuration process. This is a time saving feature that eliminates having to enter information that usually stays the same such as the company name and address.

To complete the title block for the tutorial exercise, add the name of the schematic, the document number, and sheet number information. Use the Edit command and then click on the title block area. This action brings up the title block edit submenu shown in Figure 3-51. Click on the Title of sheet option. Then enter the title text as shown in Figure 3-52. Use the same process to enter the document number, sheet number, and number of sheets as in the tutorial model.

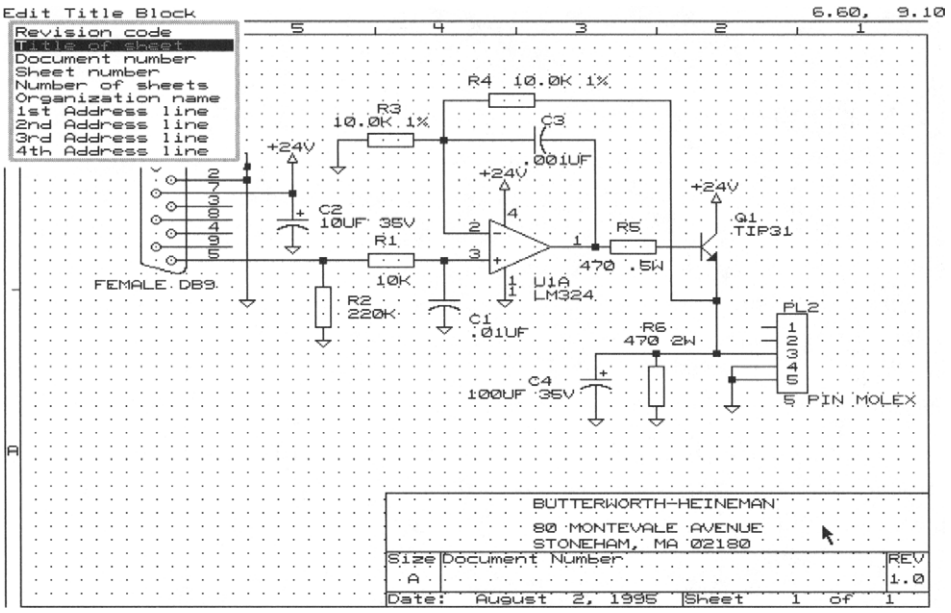


Figure 3-51 Editing The Title Block

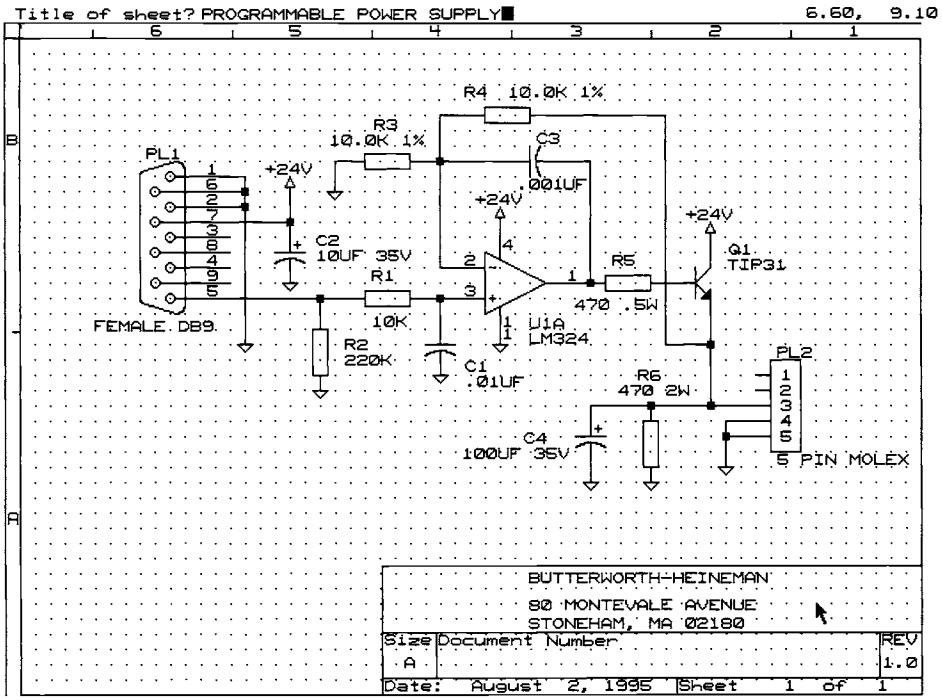


Figure 3-52 Entering the Title Text

Placing Text Notes

Your first schematic is now almost complete. Only a few text notes remain to be added. OrCAD provides for convenient entry of text anywhere on the sheet. Once text has been entered, it can be edited using the Edit command. New OrCAD users are often confused by the difference between text and labels. Both appear similar on the sheet. Labels have only one specific function: identifying wires for signal routing. The next tutorial will cover the use of labels. Text is used for general purpose descriptions and notes. In the tutorial model, text is used to annotate the signal names at the two connectors.

Start by entering the descriptive text "PROGRAMMABLE POWER SUPPLY" at the top of the sheet. Bring up the main menu, then highlight and click on the Place command as shown in Figure 3-53

Highlight and click on the Text command as shown in Figure 3-54. Then type in the text string and press **<ENTER>** as shown in Figure 3-55. The text string now appears on the sheet and moves with the mouse cursor. Move the text string

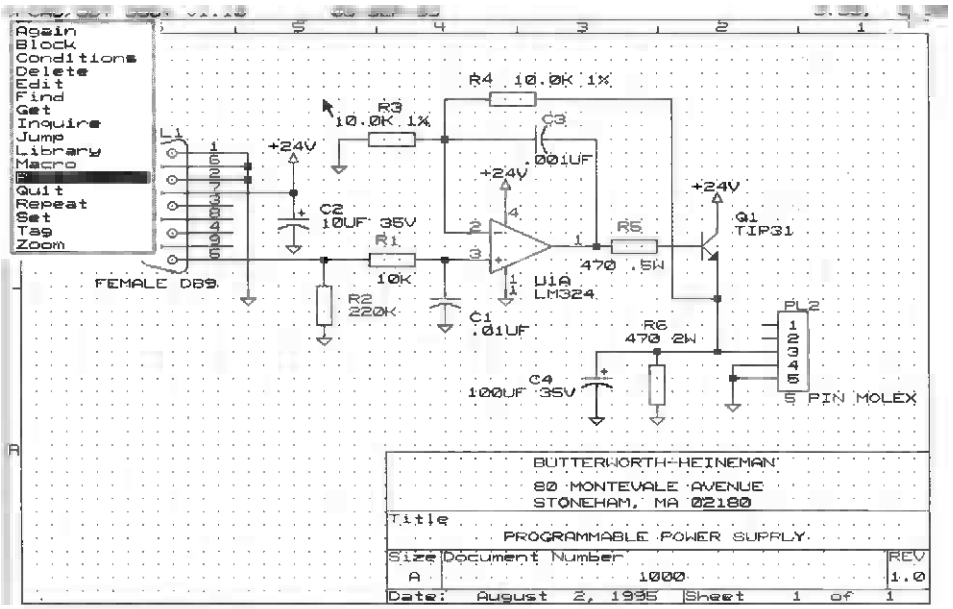


Figure 3-53 Selecting the Place Command

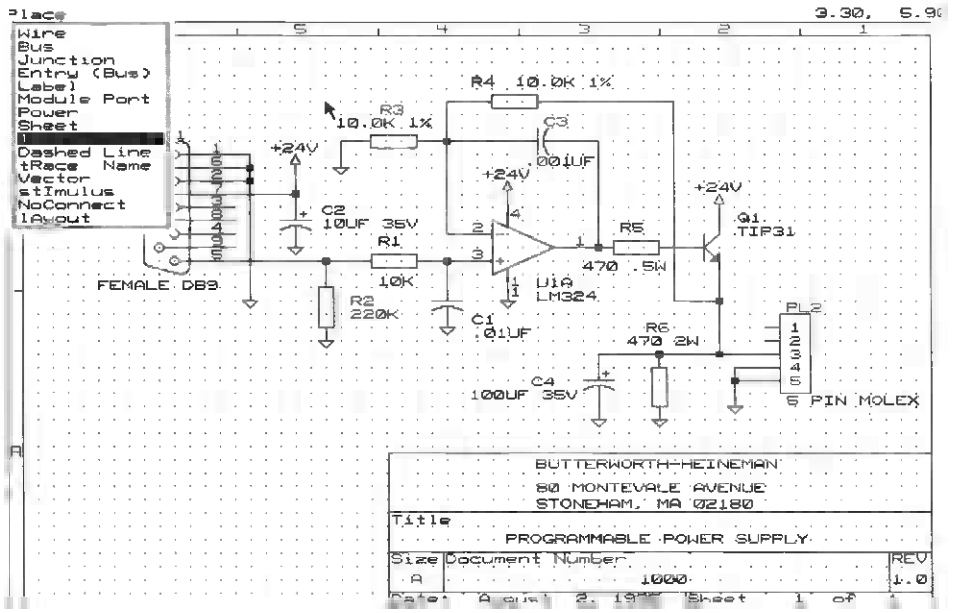


Figure 3-54 Selecting the Text Command

into the desired location as shown in Figure 3-56 and click the left mouse button to place it on the sheet.

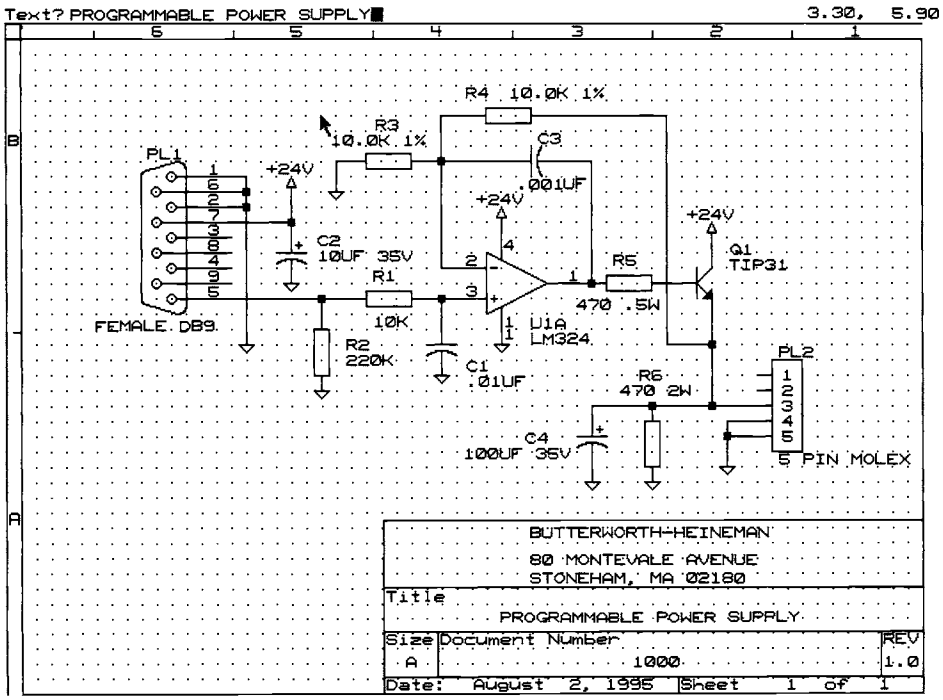


Figure 3-55 Entering the Text String

Completing the Tutorial Exercise

Use the techniques you have learned to place the remaining text on the schematic. This completes the drafting portion of the first tutorial exercise. You have now learned how to use the most common drafting commands that account for about 80 percent of every day schematic capture. To conclude the tutorial exercise, you will save the design to disk and print out a hard copy of the schematic.

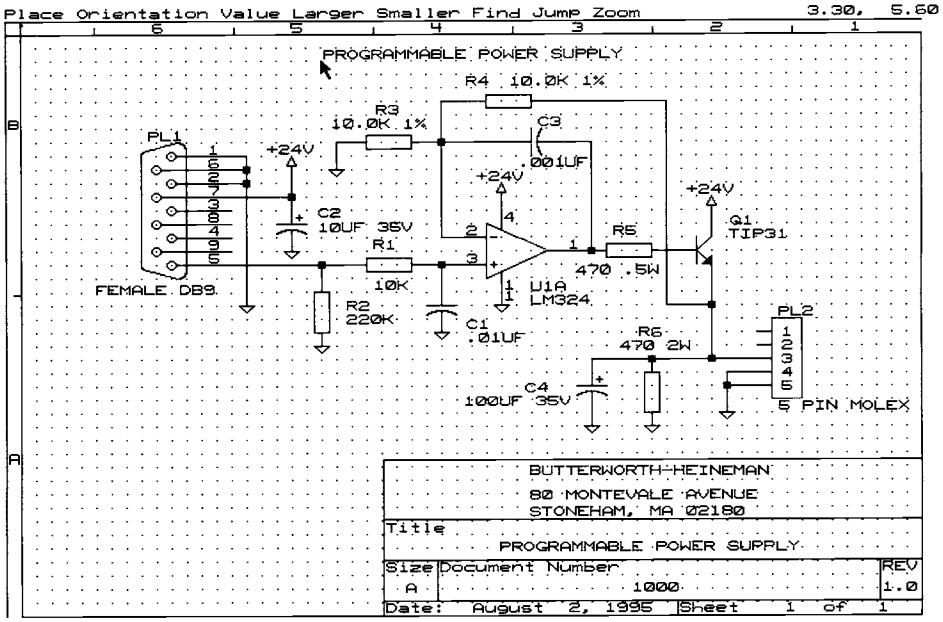


Figure 3-56 Placing the Text on the Sheet

Saving the Schematic

OrCAD does not automatically save the design when you exit the program. In the real world, when you are working on a complex schematic, good practice dictates periodically saving to disk. This reduces the likelihood that work will be lost due to operator errors, power failures, or computer crashes.

Save your tutorial exercise to disk. Bring up the main menu and then click on the Quit command as shown in Figure 3-57. Then click on Update as shown in Figure 3-58. This updates all schematic files in the design directory TUTOR1 on the hard disk.

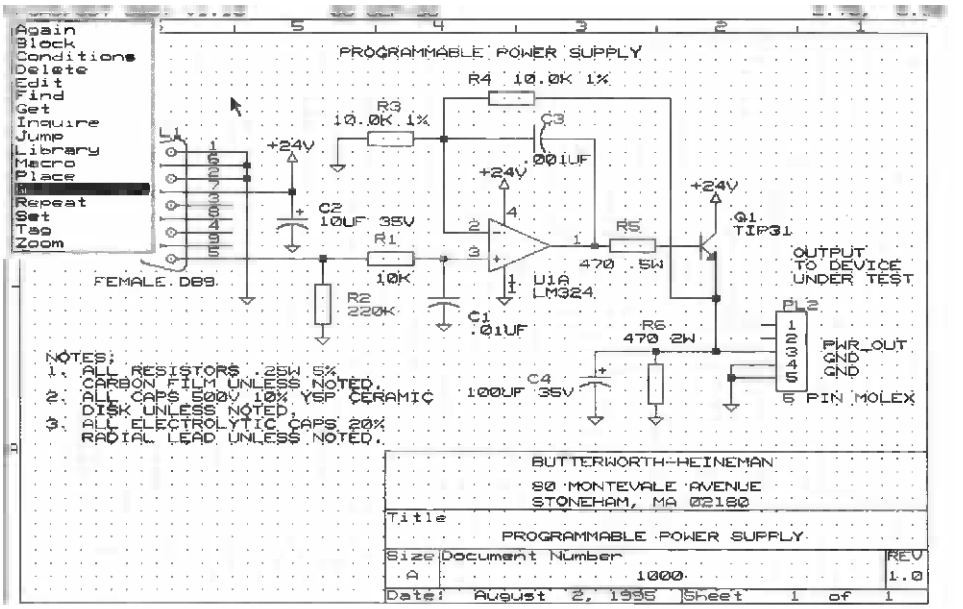


Figure 3-57 The Quit Menu

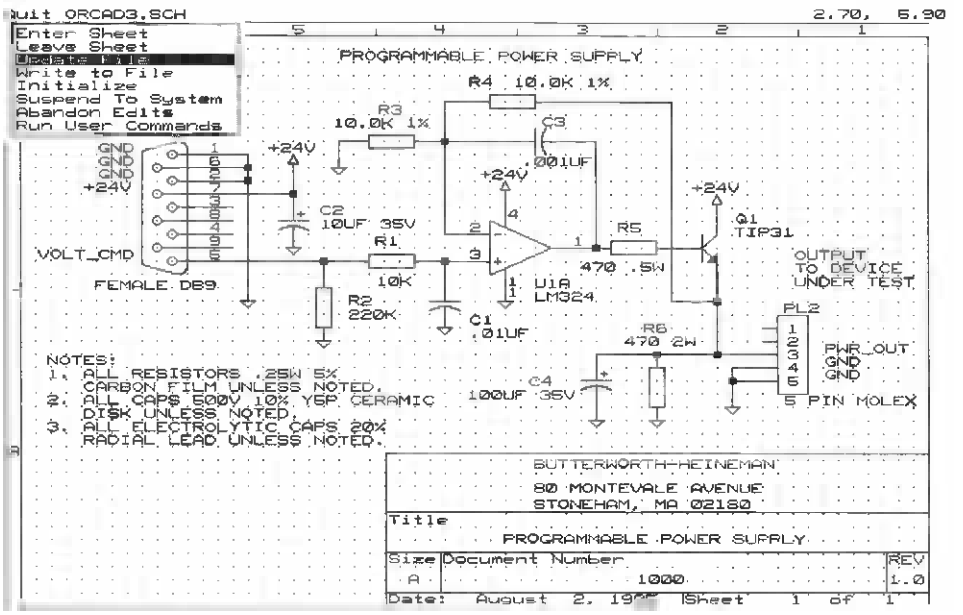


Figure 3-58 Using Update File to Save the Design

Quitting OrCAD SDT

After saving your design to disk, quit Draft (the schematic drafting program) by clicking on the Abandon Edits command as shown in Figure 3-59. This action will return you to the main OrCAD SDT screen shown in Figure 3-6.

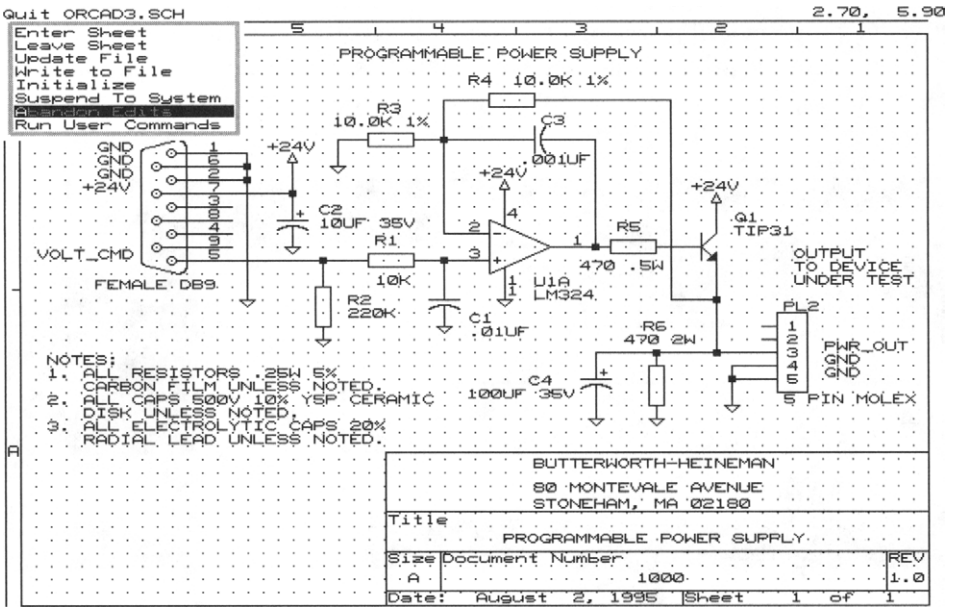


Figure 3-59 Quitting Draft Via Abandon Edits

Printing out the Schematic

The final step in this tutorial exercise is to print out your first schematic. As part of the installation and configuration process in chapter 2, OrCAD was configured to use a Hewlett-Packard compatible laser printer. The assumption is that most users will have access to a laser printer for local hard copy. If not, at least familiarize yourself with the Print Schematic configuration process.

The Print Schematic program (refer to main SDT screen in Figure 3-6) is used for local hard copy from a laser printer or similar device. Before launching Print Schematic the first time, verify the local configuration. Click the left mouse button once on Print Schematic to bring up the menu shown in Figure 3-60. Then click on Local Configuration. Verify that the options are the same as shown in Figure 3-61. Note that the program is set up to treat your design as a single sheet

and to print grid references around the border of the sheet. After you have verified the configuration, click on OK.

Next, print your schematic. From the main SDT screen (Figure 3-60), double click on Print Schematic. OrCAD will automatically format and print out the sheet.

OrCAD is not very forgiving when the printer is off-line or runs out of paper. Often the program will just hang up. Printing to a network device can be problematic. If the printer is not setup correctly or OrCAD hangs due to an off-line or out of paper condition, you may have to reboot the computer.

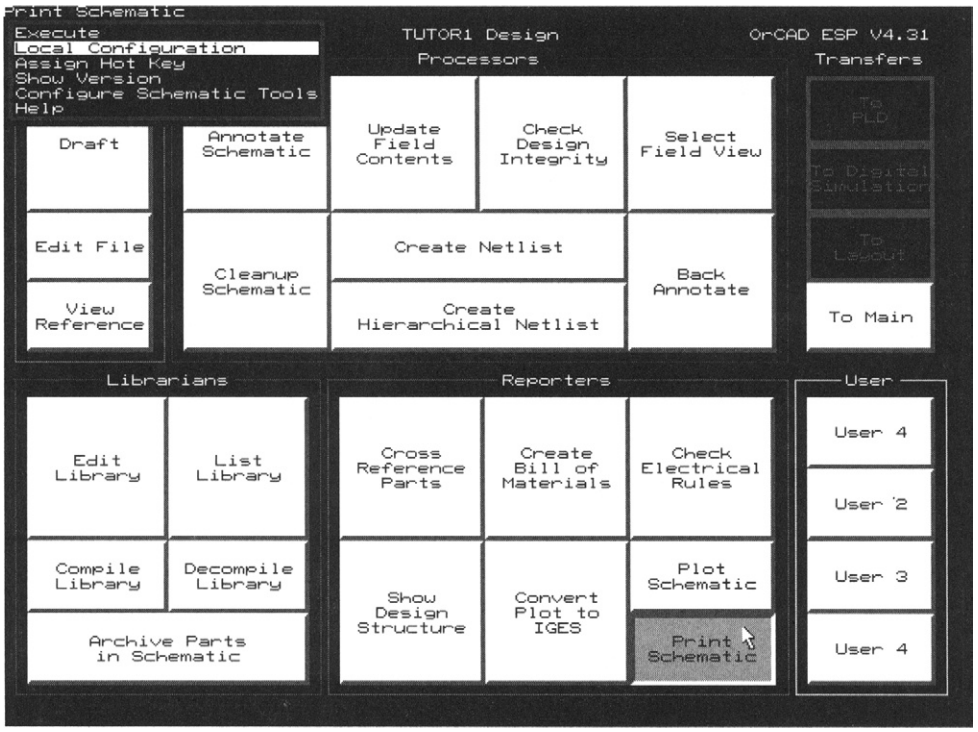


Figure 3-60 Selecting Print Schematic Configuration

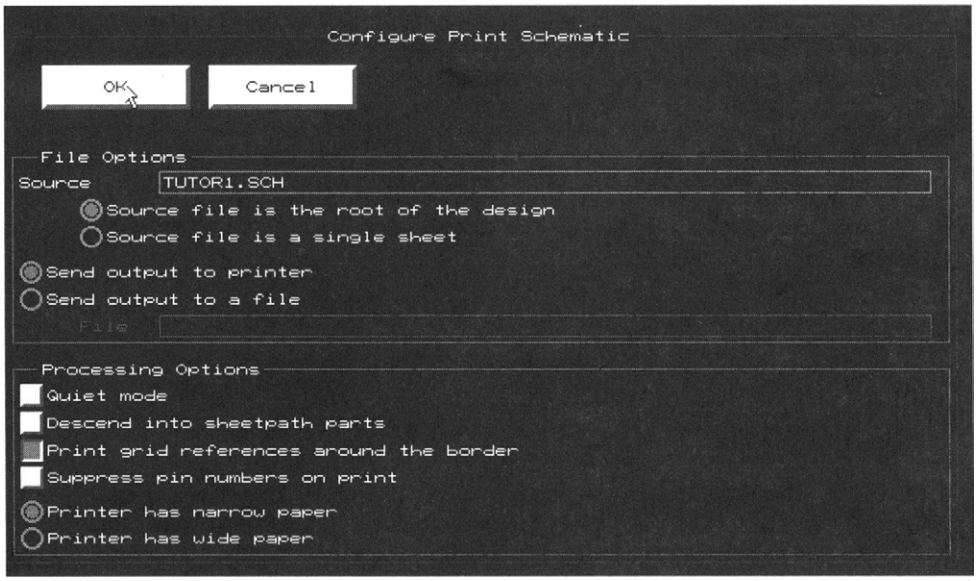


Figure 3-61 Print Schematic Configuration Options

Conclusion

You have now completed the first tutorial and are on your way to learning the ins and outs of OrCAD. In this tutorial exercise you have learned to use all the basic tools for placing and interconnecting parts and power objects, editing descriptions, and adding text notes.

4

Tutorial 2 - Hierarchical Design

This chapter is the second tutorial exercise. The intention is to build your skill level beyond the basics. In the previous tutorial, a simple schematic was created and then plotted out. The mechanics of creating this schematic were not unlike manual drafting using templates or even the use of a general purpose CAD system. In the second tutorial exercise, you will start to unlock the power of schematic capture. This will entail creating a complex hierarchical schematic and then using postprocessing tools to check for errors and generate a bill of materials.

Starting the Second Tutorial

Figure 1-18 is used as the model for this tutorial. Again, you should make a copy of the figure parts and keep them handy for reference. Prior to starting, review the material in chapter 1 on hierarchical schematics (pages 31-33), especially the OrCAD terminology. Also, you may want to briefly review the material in chapter 3. The assumption is made that you are familiar with the basics of creating designs, placing and editing parts and objects, and saving the design. The tutorial is divided into two sessions. Plan on spending several hours on each session. Be sure to save your work in between sessions.

First Session

Start the tutorial by creating a new design called TUTOR2, based on the standard template. Use the OrCAD Design Management Tools as explained in chapter 3. Then select TUTOR2 and launch OrCAD SDT and Draft.

Creating the Hierarchy

The first (top) sheet of a hierarchical design is a block diagram that shows interconnections between additional sheets (circuit blocks) in the next level down. The model for this tutorial is a single level hierarchy. The top sheet shown in Figure 1-18A on page 32 shows three circuit blocks.

The first task is to create the top sheet. Start by editing the title block using the title and document number shown in Figure 4-1.

The next step is to create the hierarchy. This entails placing three sheet objects, adding sheet nets for the signals, and interconnecting the sheet nets with wires.

Use the model as a guideline for object locations. The best technique is to place objects in approximate locations and then make any required adjustments by using the Block Drag command.

Place the Microcontroller sheet object near the center of the screen. Start by bringing up the main menu and click on Place as shown in Figure 4-1. Then click on Sheet as shown in Figure 4-2. The Place Sheet submenu shown at the top of Figure 4-3 will appear. Move the mouse cursor to the desired upper left corner of the sheet and click the left mouse button. Next move the mouse to the lower right corner and click again as shown in Figure 4-4.

The sheet object appears as shown in Figure 4-5 and the sheet edit submenu automatically appears at the top of the screen. This is the same submenu that would appear if you placed the cursor on the sheet object and then clicked on the Edit command.

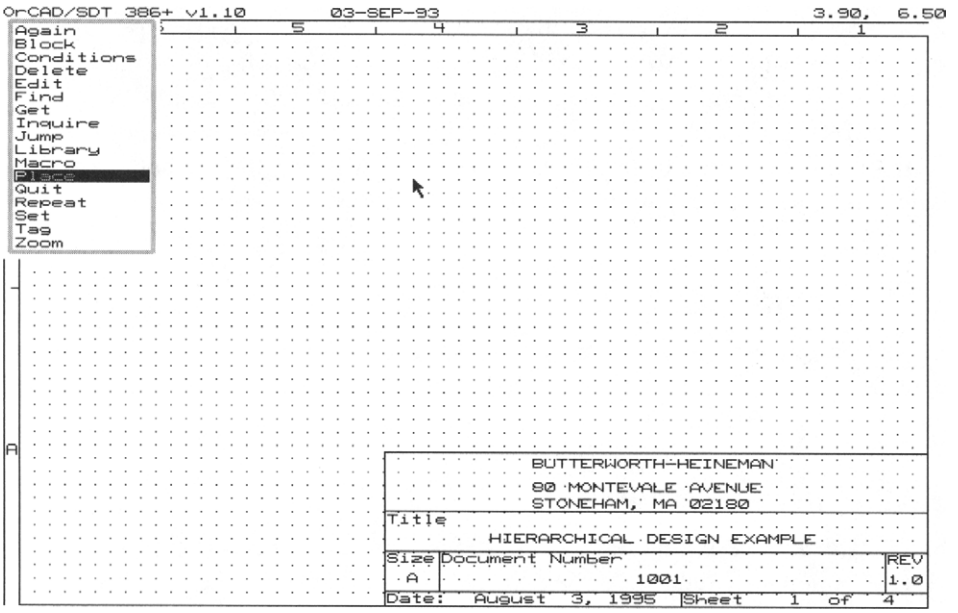


Figure 4-1 Selecting the Place Command

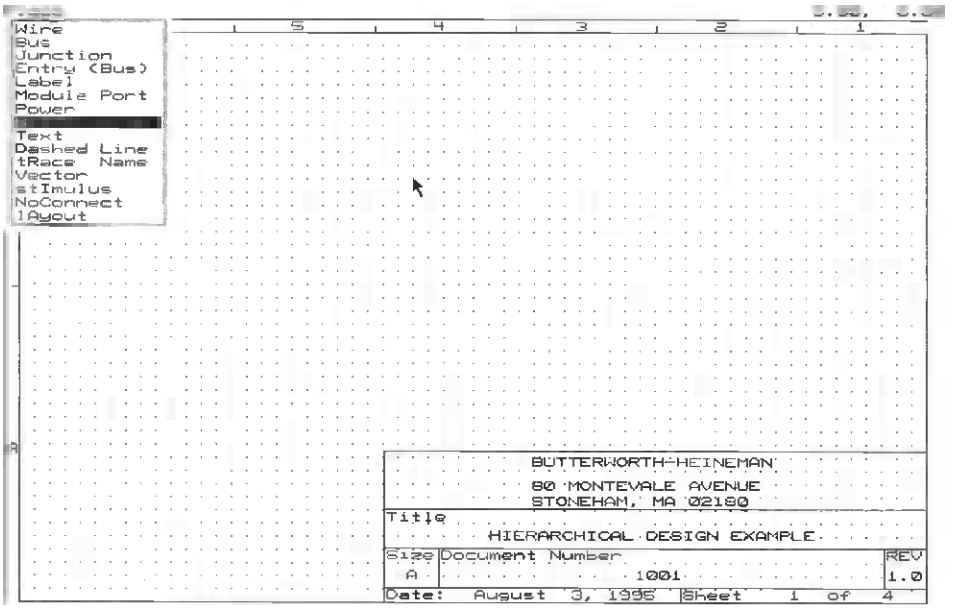


Figure 4-2 Selecting the Place Sheet Command

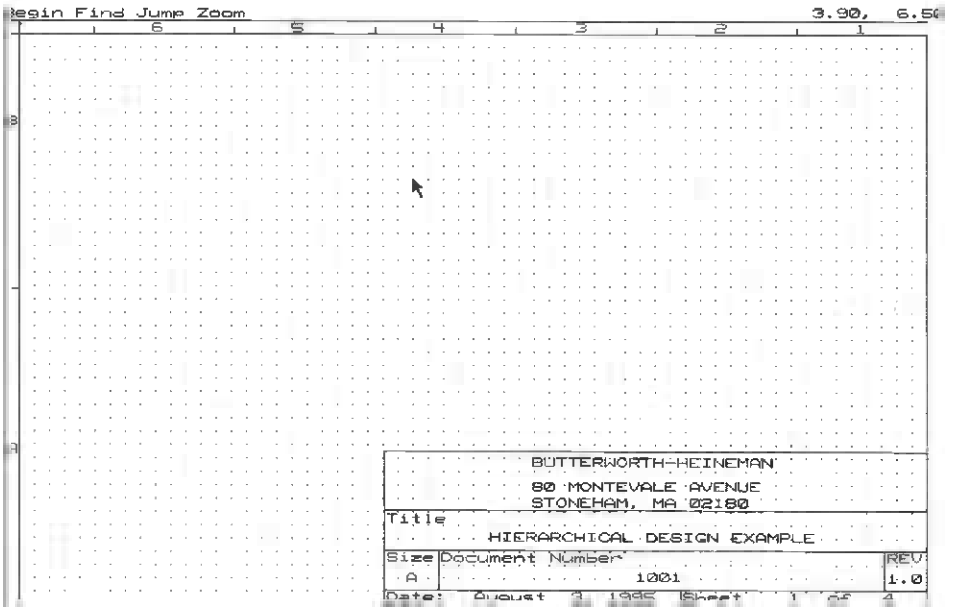


Figure 4-3 Placing the First Corner of the Sheet

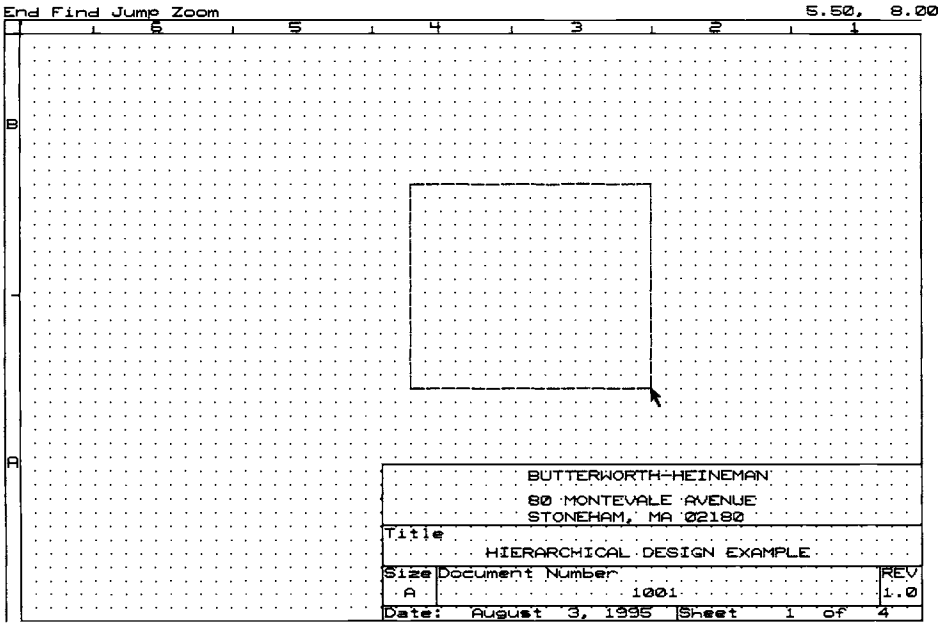


Figure 4-4 Placing the Second Corner of the Sheet

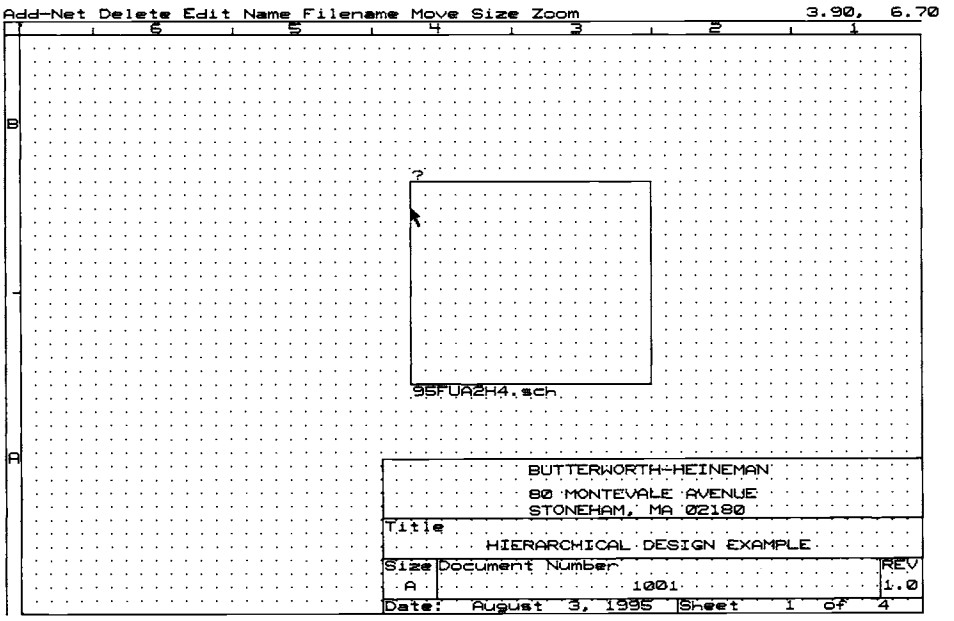


Figure 4-5 The Edit Sheet Submenu

Editing the Sheet Name and Filename

Each sheet object has a name that appears at the top and a filename that appears at the bottom. The name defaults to a "?" symbol. Good practice is to enter a descriptive name for the sheet object that matches the title that will appear in the title block area for the corresponding sheet. OrCAD generates a random alphanumeric prefix for the filename. The extension for schematic files is always .SCH. You can retain the OrCAD generated filename, but the random characters appear strange on a finished schematic and make life difficult if DOS file copy operations are used at some later point. Suggested practice is to use a filename prefix that matches part of the sheet name. Remember that DOS only allows eight alphanumeric characters for the filename prefix.

Edit the sheet name. Hold the left mouse button down to bring up the pull-down menu and then highlight the Name option as shown in Figure 4-6. Enter the name text as shown in Figure 4-7. Then repeat the process for the filename, using the Filename option as shown in Figure 4-8. Use the same name and filename as in the model. When you are done, the sheet object should appear as in Figure 4-9. Click the right mouse button to escape.

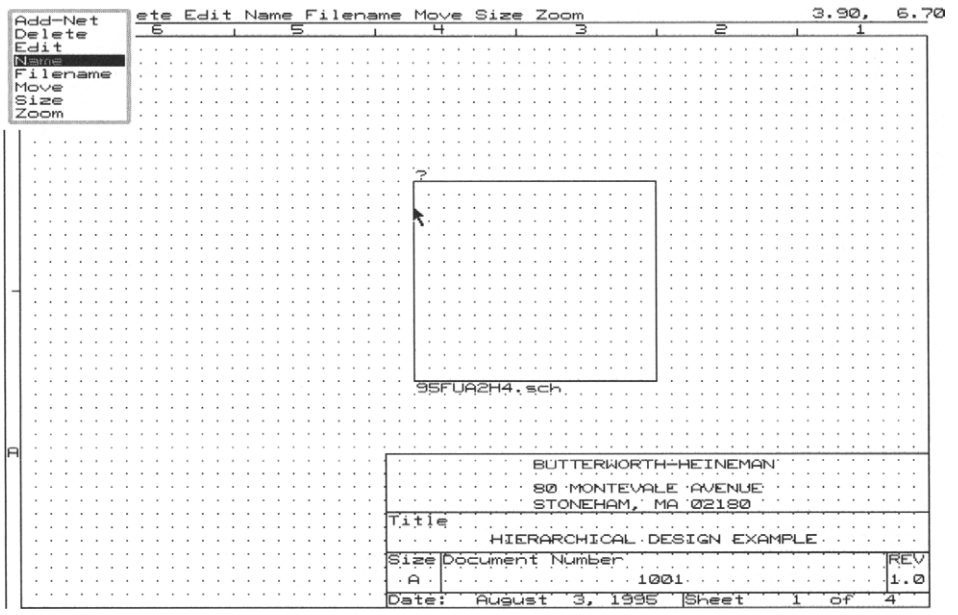


Figure 4-6 Selecting the Edit Sheet Name Command

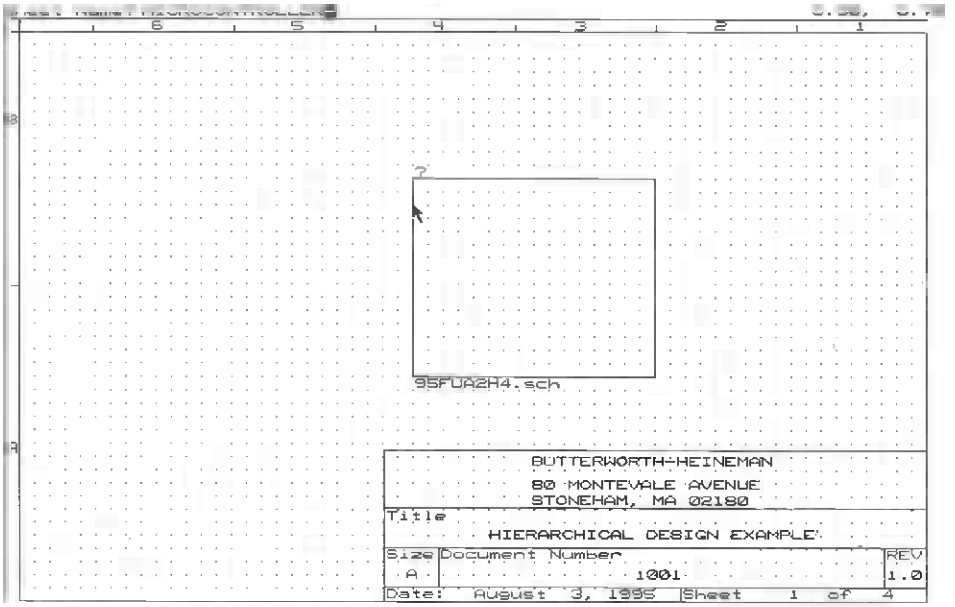


Figure 4-7 Entering the Sheet Name

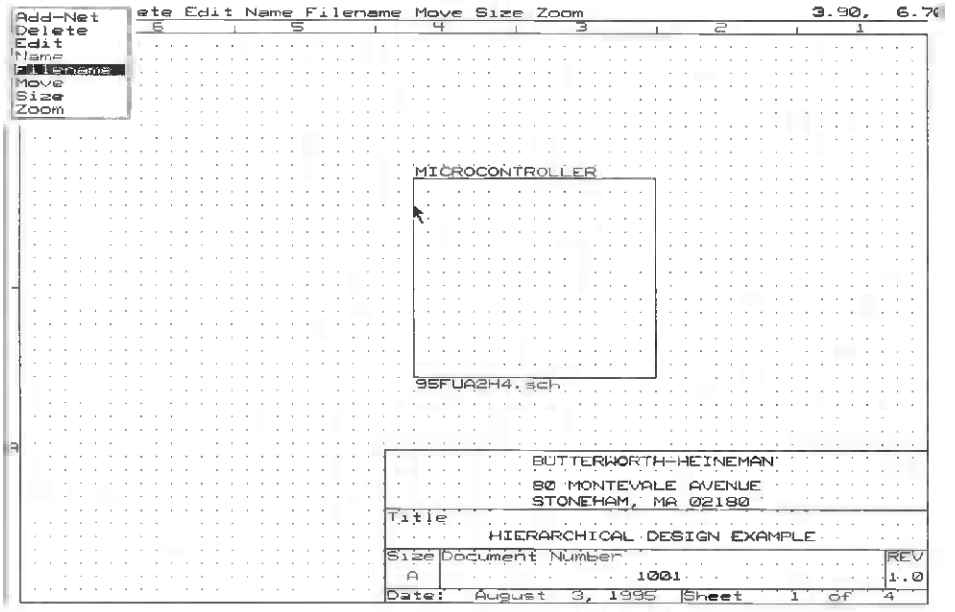


Figure 4-8 Selecting the Edit Sheet Filename Command

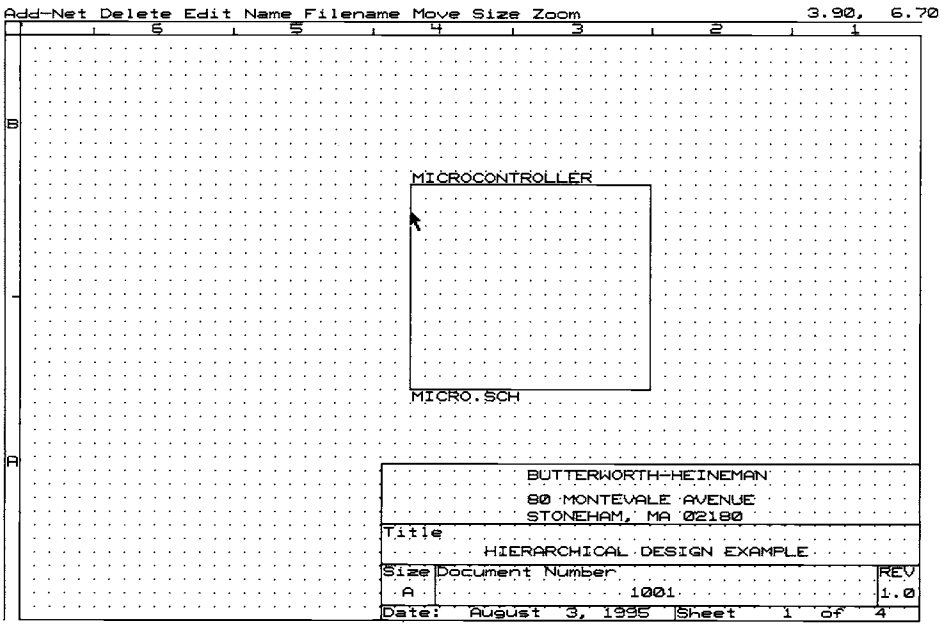


Figure 4-9 Sheet Object with Name and Filename

Completing the Sheets

Use the process you just learned to place the remaining two sheet objects as shown in Figure 4-10. Edit the names and filenames to be the same as the model. Note that one of the edit commands is Size. You can use this command to move the lower right hand corner and resize a sheet object. Block Drag is convenient for moving sheet objects around. If you make a mistake or want to change the name or filename, just use the Edit command.

Adding Sheet Nets

Sheet nets are the entry and exit points for signals that run between sheets. Remember that power and ground are automatically interconnected between all sheets in a hierarchy, unless measures are taken to isolate them.

Sheet nets are added to sheet objects by using the Edit command. Start by locating the cursor at the point where the sheet net is to be added, as shown in Figure 4-10. Click the left button to bring up the main menu, then highlight and click on Edit. The first selection on the edit submenu is Add-Net, as shown in Figure 4-11. You can just click the mouse button to execute this first option. Then enter the sheet net name as shown in Figure 4-12.

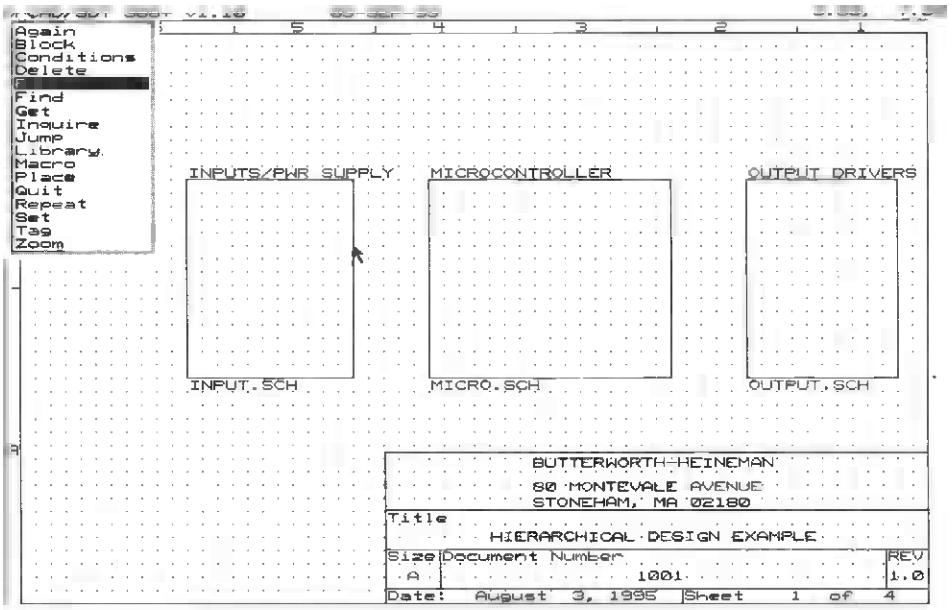


Figure 4-10 Using Edit Command to Add Sheet Nets

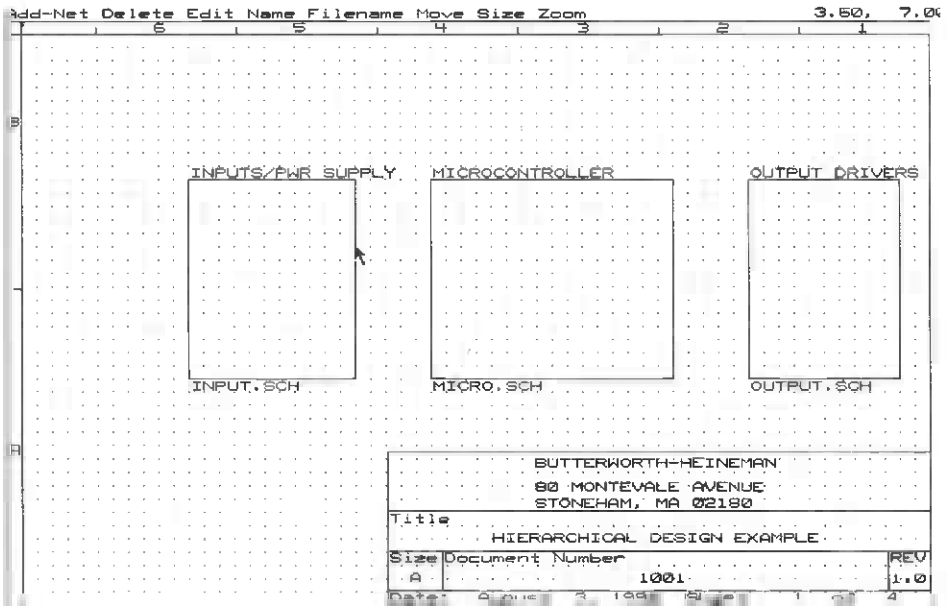


Figure 4-11 Adding a Sheet Net

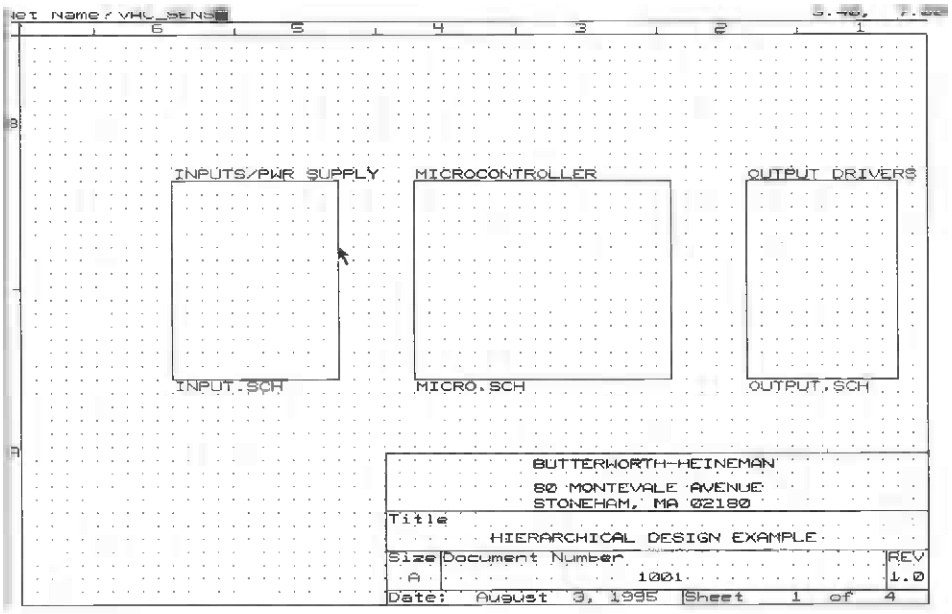


Figure 4-12 Entering the Sheet Net Name

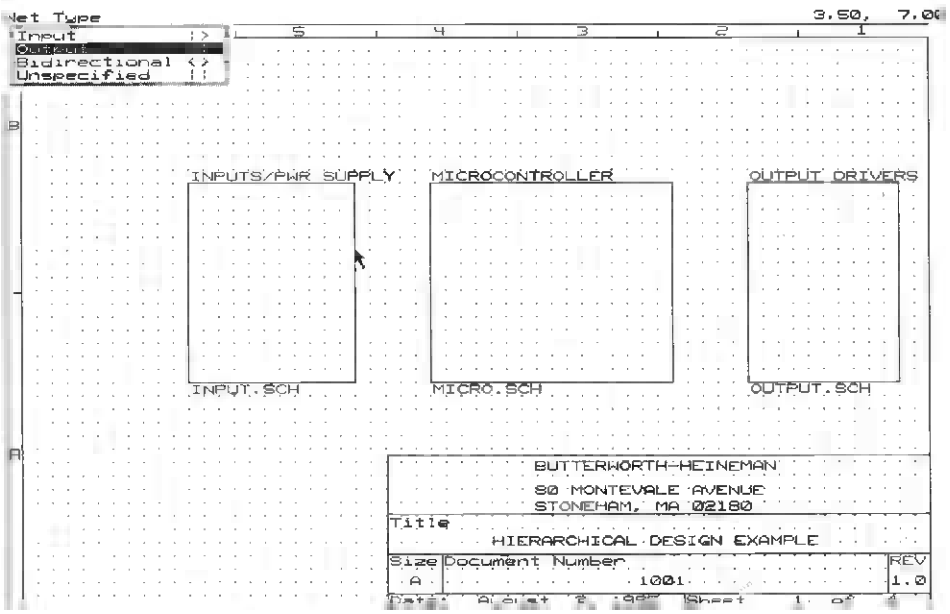


Figure 4-13 Selecting the Sheet Net Type

After entering the name of the first sheet net, the menu shown in Figure 4-13 appears. This menu asks for the type of sheet net. The signal in question, VAC_SENS, is an output from the INPUTS/POWER SUPPLY circuit block, so highlight and click on the Output option. The first sheet net is now completed and the screen appears as shown in Figure 4-14. The Edit Sheet command, as with other Edit commands, remains modal until you escape by clicking the right mouse button. You can continue and add the second sheet net, CAM_POS, by moving the cursor down to the desired location and clicking the left button again.

The Edit Sheet command is restricted to a given sheet object, so you will have to escape before you can add sheet nets to the other sheet objects. The Edit Sheet command works on whatever sheet object the cursor is located at when the command is launched. Use the Edit Sheet command to place the remaining sheet nets shown in Figure 4-15.

Interconnecting Sheet Nets

To complete the interconnection of signals between sheets, the sheet nets must be wired together. An important fact to remember is that the sheet net name defines a signal only within a given sheet. Two sheet nets with the same name on different sheets are not automatically interconnected. Conversely, you can interconnect sheet nets with different names. The only exceptions are power and ground, which are not considered signals. Power and ground are always common to an entire schematic unless special measures are taken to isolate a section of the circuitry.

In this tutorial, all interconnections are between sheet nets with the same name. Use macro F1 to place wires and interconnect the sheet nets as shown in Figure 4-16.

You have now completed the top level of the hierarchy. The next step is to complete the three sheets that make up the lower level of the hierarchy: INPUTS/PWR SUPPLY, MICROCONTROLLER, and OUTPUT DRIVERS.

Navigating between Sheets

Working on a hierarchical schematic design usually requires a considerable amount of navigation between the various sheets. OrCAD provides convenient commands for this purpose. Going from a higher level sheet, such as sheet one, into a lower level sheet, such as sheet two, is called "entering" the sheet. Going the opposite way to return to a higher level is called "leaving the sheet." Each sheet is stored in a separate file (with .SCH suffix) in the design subdirectory.

Only one file can be open at a time, which means that you can only view and work on one sheet at a time.

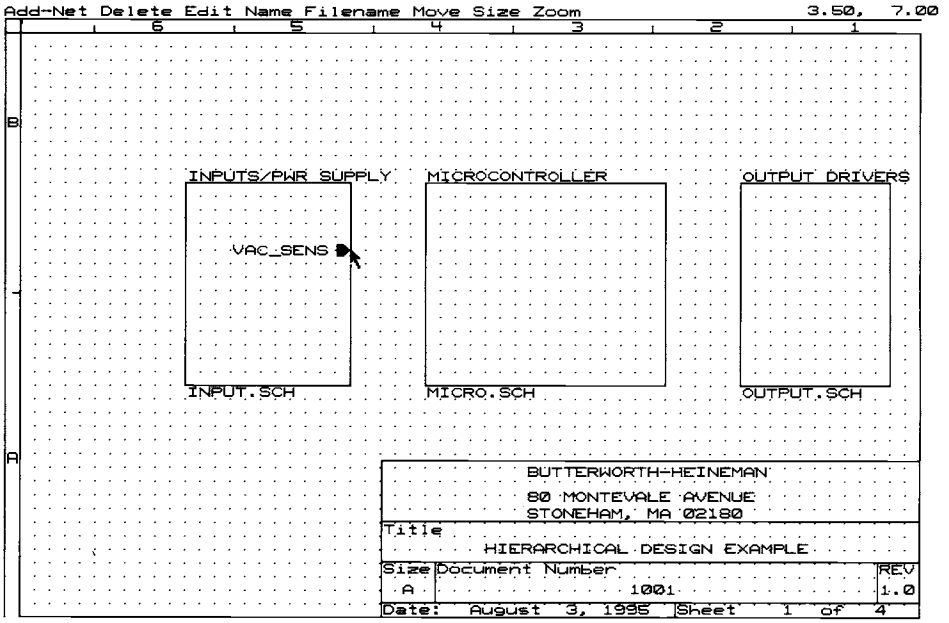


Figure 4-14 Completed First Sheet Net

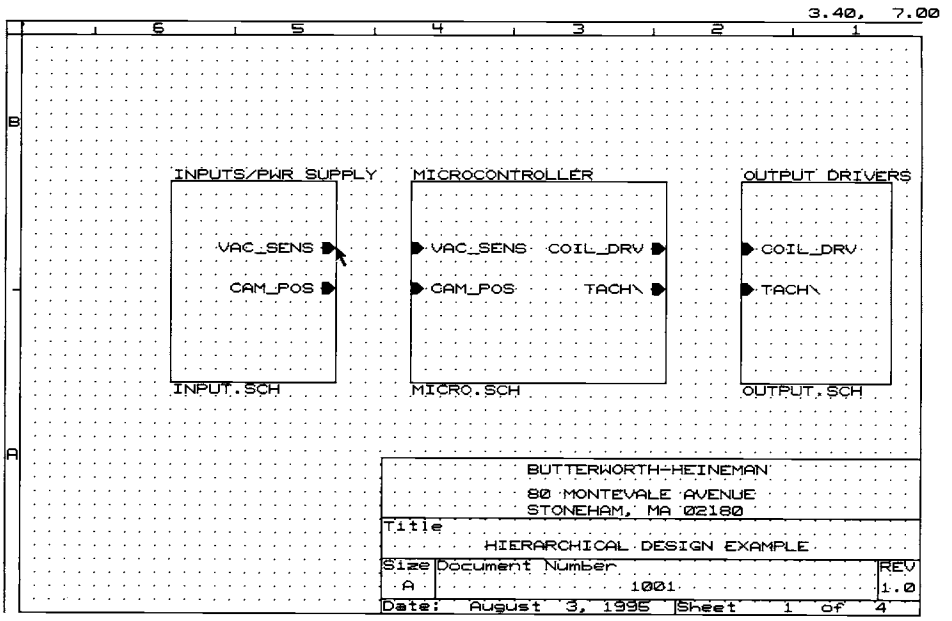


Figure 4-15 All Sheet Nets Completed

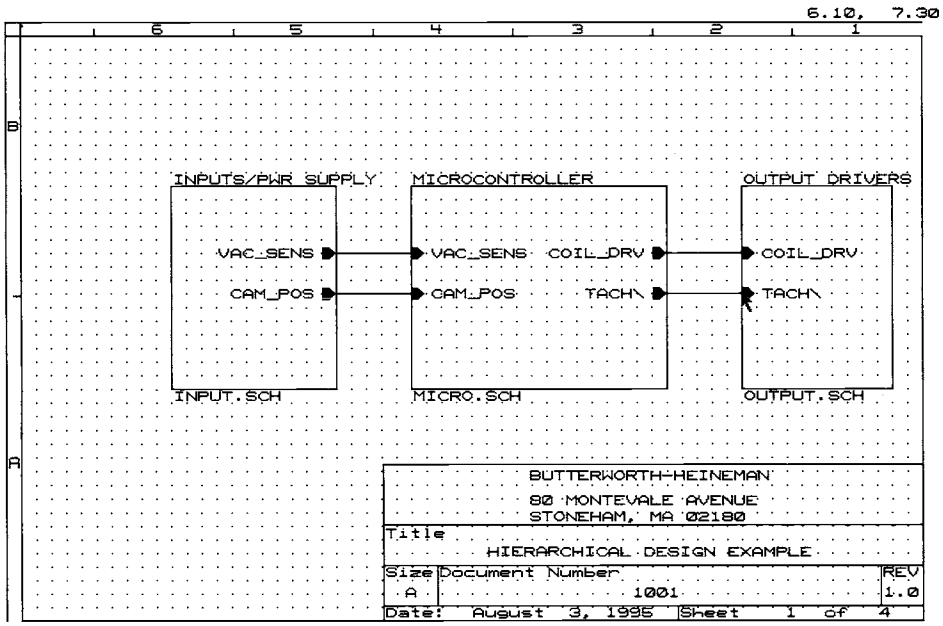


Figure 4-16 Completed Sheet Net Interconnections

No provision exists in OrCAD for multiple windows into a design. This is a significant limitation to users accustomed to Windows. You must write the active sheet to a disk file using the Update command before leaving for a higher level or entering a lower level. Otherwise you will lose whatever changes have been made. OrCAD does not automatically save the file for you, even though a warning message will appear.

When you create a sheet object, an empty sheet is automatically created and associated with the sheet object. You can easily enter this sheet via the Quit menu. Open the main menu and click on Quit as shown in Figure 4-17. Before you proceed, remember to save the current sheet to disk by clicking on Update File. Next, click on Enter Sheet as shown in Figure 4-18. Then position the mouse cursor on the sheet you want to enter and click the left button as shown in Figure 4-19.

The selected sheet will appear as shown in Figure 4-20. Initially this sheet will be empty, with the exception of default title block information. Refer to the following section for more details about completing the second sheet. When the second sheet has been completed, access Quit from the main menu. Remember to save the data to disk by using the Update command. Click on Leave Sheet to return to the top sheet (sheet one).

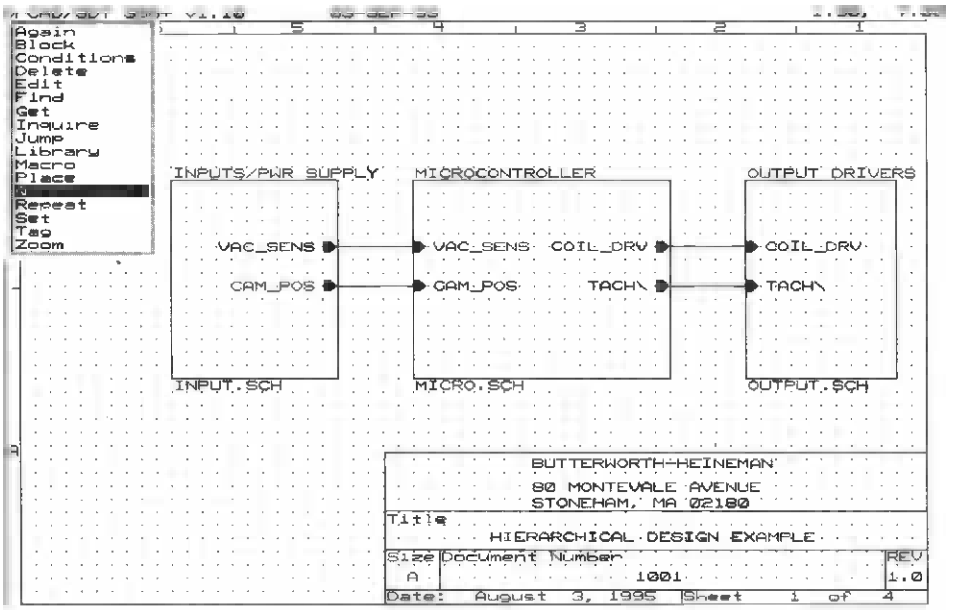


Figure 4-17 Selecting the Quit Command

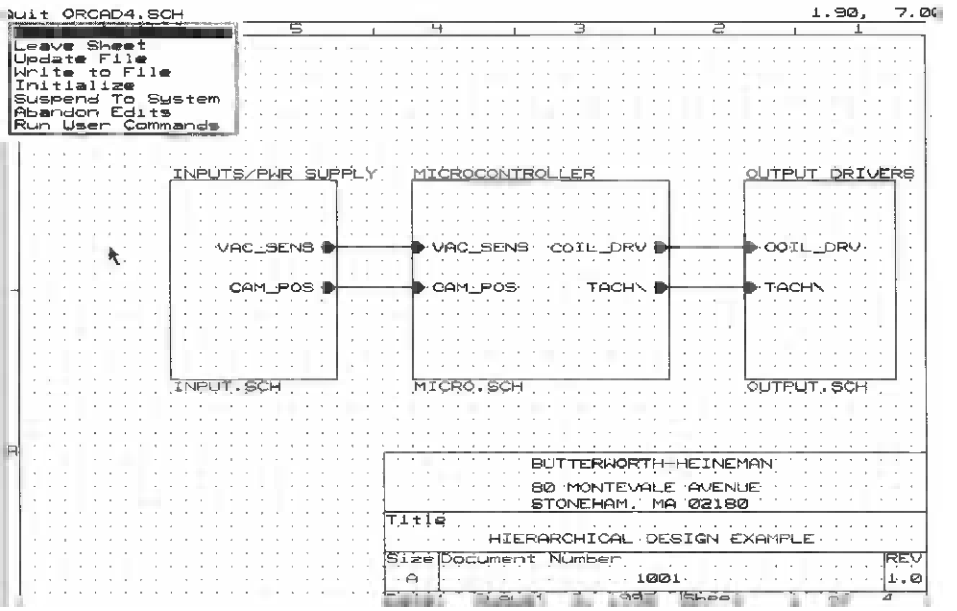


Figure 4-18 Selecting the Enter Sheet Option

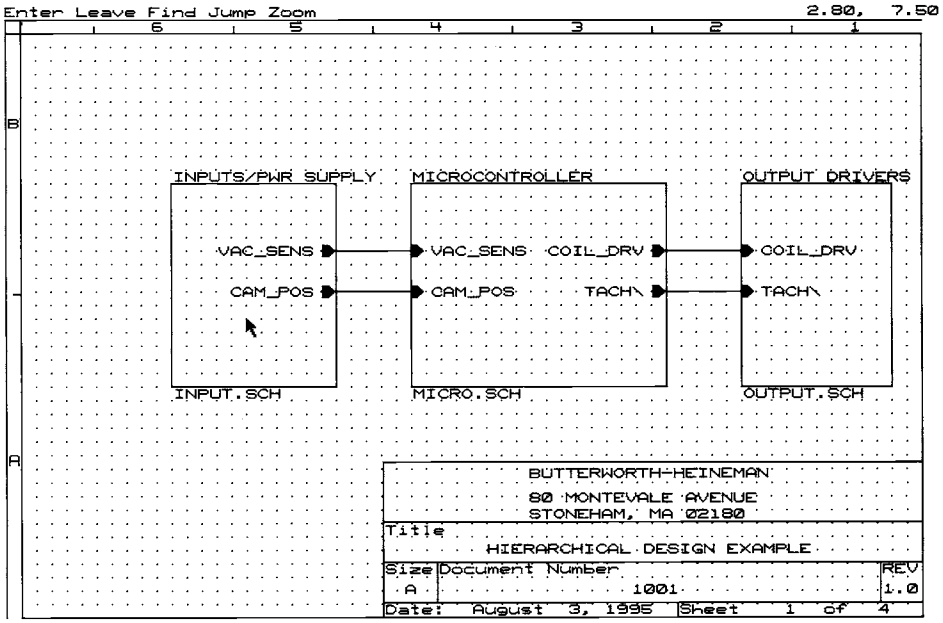


Figure 4-19 Clicking on the Sheet to be Entered

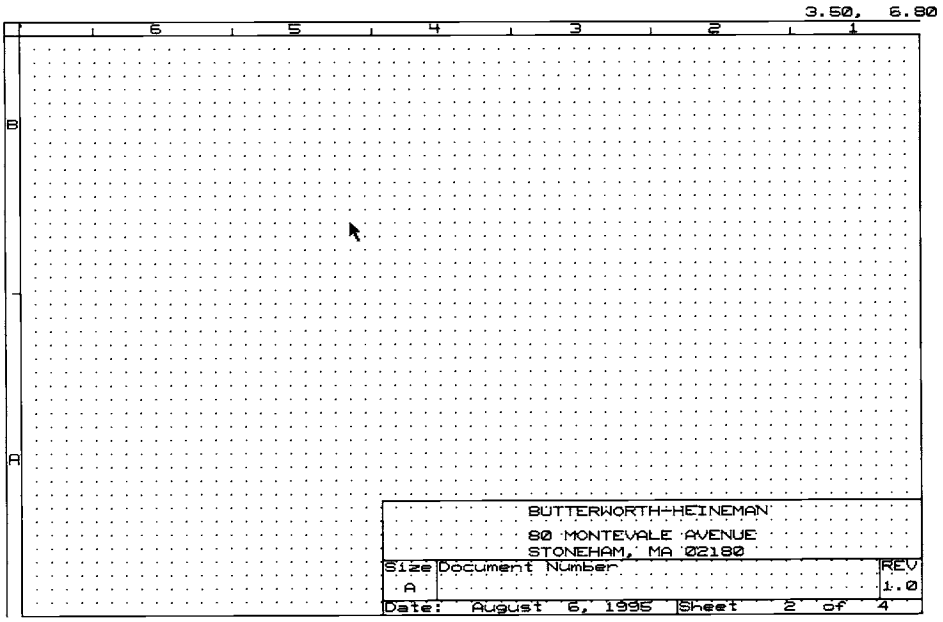


Figure 4-20 Entry into the Second Sheet

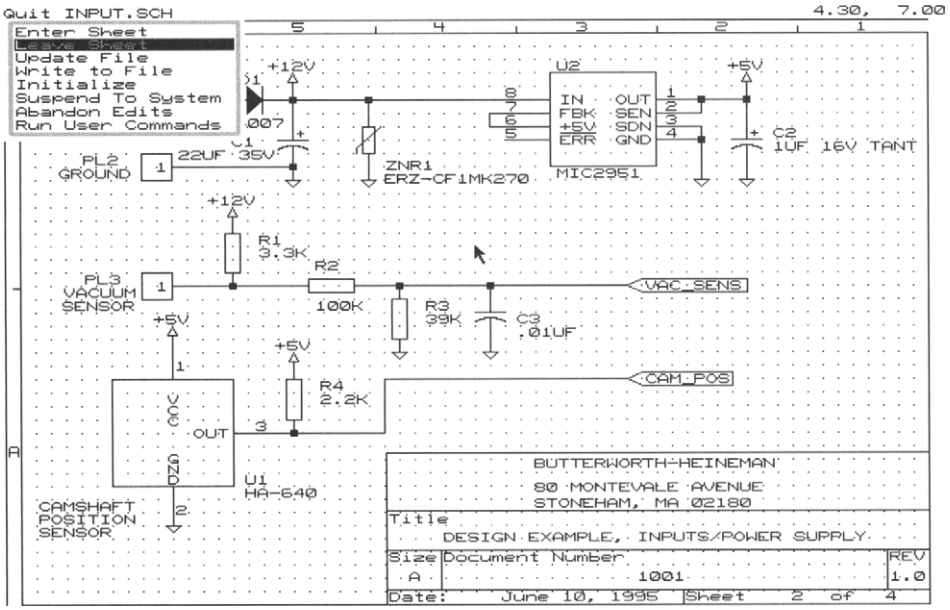


Figure 4-21 Navigating with the Leave Sheet Command

Completing the Second Sheet

The completed second sheet is shown in Figure 1-18B (and above in Figure 4-21, with a section obscured by the menu). By now you should be able to complete this sheet on your own using the skills learned in chapter 3. Most of the parts are common resistors, capacitors, and diodes that can be placed using the macro commands (refer back to pages 69-70 for details).

The design example used for this tutorial is based on an ignition system for a Harley-Davidson motorcycle. The upper section of sheet one is the power supply that uses an automotive type voltage regulator, the Micrel MIC2951. U1 is a Senisys HA-640 Hall Effect position sensor that generates a timing reference pulse based on camshaft position. A vacuum sensor signal is input on PL3. The second sheet shows a Microchip PIC16C71 RISC microcontroller with on-chip four channel A/D converter. The two trimpots R5 and R6 are used to set the Rpm limit and adjust the timing advance curve. The last sheet shows two output drivers. Q1 drives an electronic tach with a +12V square wave. Q3 is a special Darlington transistor, manufactured by Fuji, that can drive an ignition coil and withstand +350V pulses and directly sink upwards of 5 amps.

This is a real world circuit. Some of the parts are fairly unique and may not be familiar. Listed below are some hints on getting these parts from the OrCAD libraries. In each case, the part name and library is given. Because the Get command will automatically search all libraries for the part name, the library name is for reference only.

- **Connectors PL1 - PL3**: HEADER 1 from CUSTOM.LIB. Note the space between HEADER and the character 1.
- **Hall Effect Sensor U1**: HA-640 from CUSTOM.LIB.
- **Voltage regulator U2**: MIC2951 from CUSTOM.LIB. Mirror before placing.
- **Varistor ZNR1**: VARISTOR from CUSTOM.LIB

Placing the Module Ports

Module ports are used to route signals onto and off of sheets. A one-to-one correspondence must exist between the module ports shown on the sheet and the sheet nets shown on the sheet object. For each sheet net shown on the INPUTS/PWR SUPPLY sheet object on sheet one of the hierarchy, a module port with the same name and type (input, output, and so forth) must be shown on sheet two. The module ports can be located anywhere on the sheet, location and order are irrelevant. Only the name and type must match the sheet net. General practice is to place input ports on the left and output ports on the right. Maintaining some spatial relationship between the order of module ports on the sheet and sheet nets on the sheet object improves schematic readability, but may not be possible on complex and crowded schematics.

Place the VAC_SENS module port. Bring up the Place menu as shown in Figure 4-22 and then click on Module Port. Then enter the name as shown in Figure 4-23. Next highlight and click on the appropriate type, in this case OUTPUT, as shown in Figure 4-24. The module port appears on the sheet and moves with the mouse cursor. Position the module port as shown in Figure 4-25. Then click the left mouse button to place the module port onto the sheet.

Because the length of a module port varies with the number of characters in the name, the suggested approach is to place the module port first and then connect it with a wire. This will avoid having the end of the wire overlap into the module port.

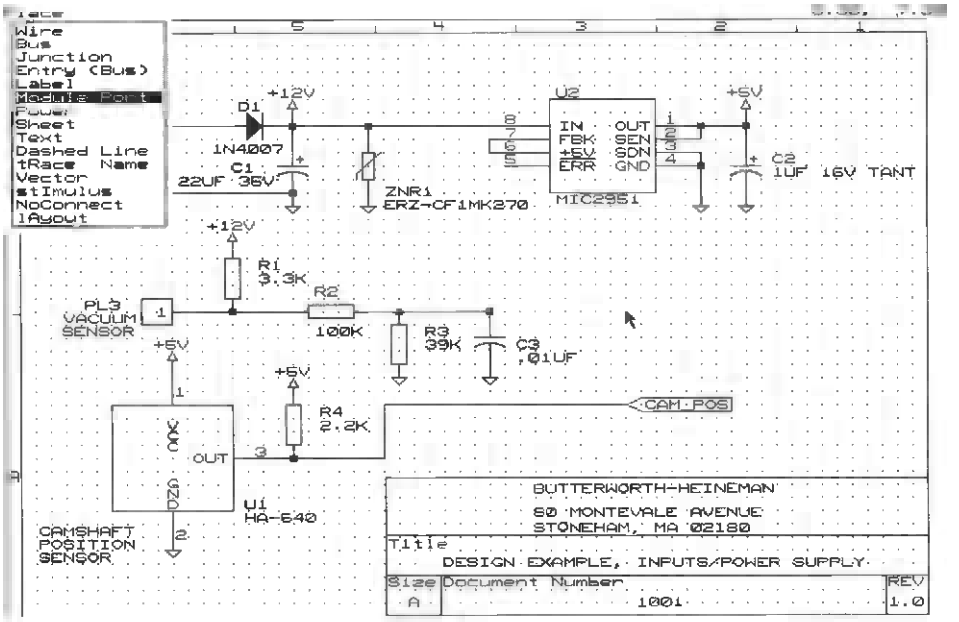


Figure 4-22 Selecting the Place Module Port Command

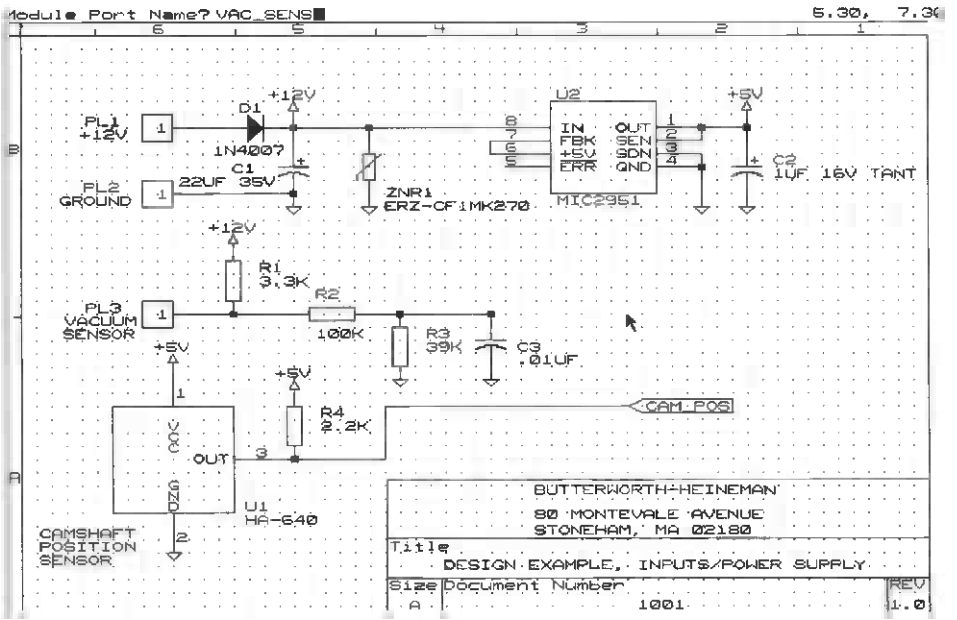


Figure 4-23 Entering the Module Port Name

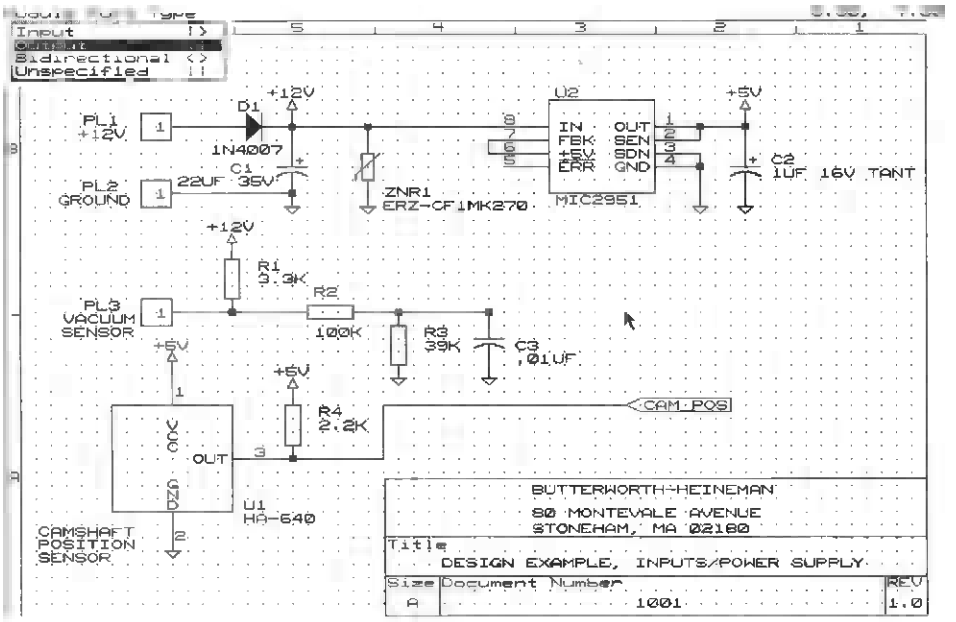


Figure 4-24 Selecting the Module Port Type

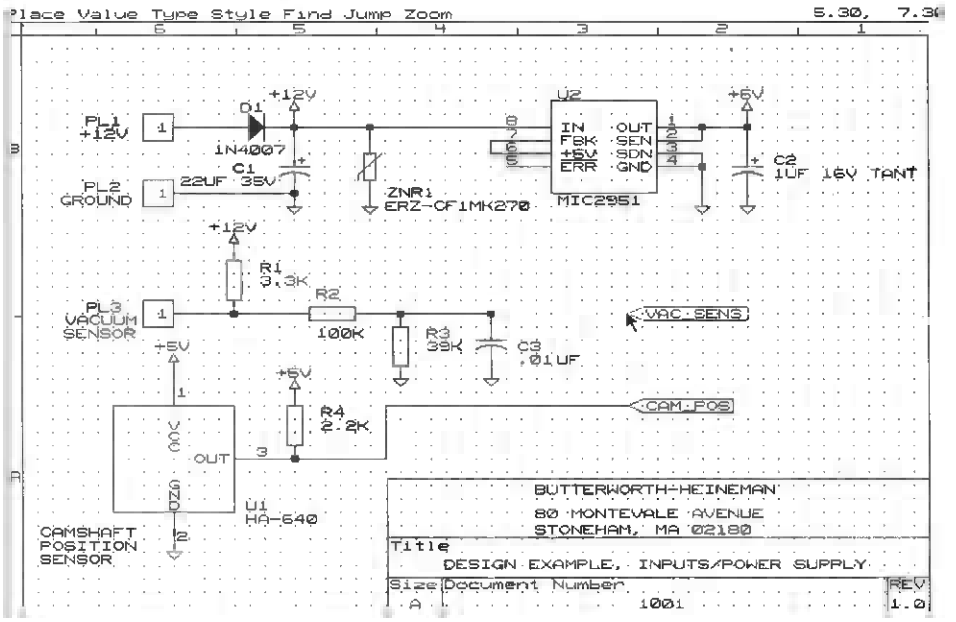


Figure 4-25 Placing the Module Port on the Sheet

Completing the Third Sheet

The third sheet involves many of the same considerations as the second sheet. Again, module ports are placed as explained in the previous section. Note that both input and output type module ports are used on sheet three. Listed below are the special parts:

- **Trimpots R5 and R6**: POT from CUSTOM.LIB. Mirror and then rotate three times before placing.
- **Microcontroller U3**: PIC16C71_ALT from CUSTOM.LIB. Mirror before placing.
- **Resonator Y1**: RESONATOR from CUSTOM.LIB. Rotate three times before placing.

Placing Labels on Wires

Often wires must be routed from one end to another on a crowded sheet. The resulting maze of wires can reduce clarity and cause confusion and errors in tracing signal paths. On sheet three the CAM_POS signal is an input and the corresponding module port appears on the lower left side. The signal must run to pin 6 on U3, which is on the other side of the sheet. To illustrate the use of labels, the signal connection is made with two wire segments which are labeled. The OrCAD connectivity database considers this the same as a continuous wire.

In order for a label to become associated with a wire, the lower left corner of the label must touch the same grid point as the wire. Imagine the character "C" enclosed in a box. The lower left corner of this box must be on the wire. When placing labels, both the label and wire must be drawn on grid.

Place the CAM_POS label. Bring up the Place menu as shown in Figure 4-26 and click on Label. Enter the name as shown in Figure 4-27. The label appears and moves with the mouse cursor. Position the label on top of the wire as shown in Figure 4-28 and then click the left mouse button to place it onto the sheet.

Completing the Fourth Sheet

After having completed the previous three sheets, you should find that drawing sheet four is routine. There are only two parts that require mention. NPN Darlington transistor Q3 is found in the DEVICE.LIB under the name NPN DAR. Note that a space must be entered between the characters NPN and DAR. Mirror HEADER 1 when placing PL4 and PL5.

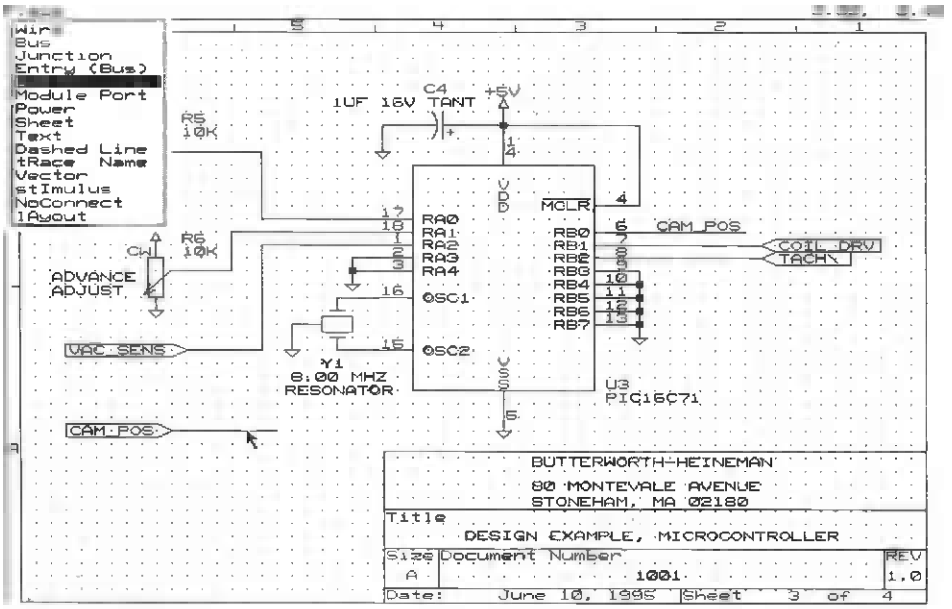


Figure 4-26 Selecting the Place Label Command

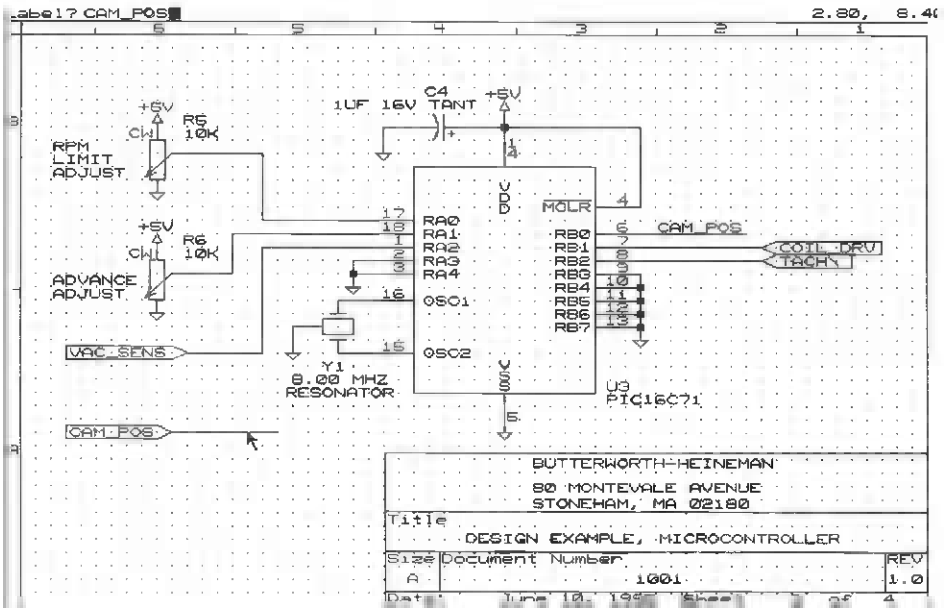


Figure 4-27 Entering the Label Name

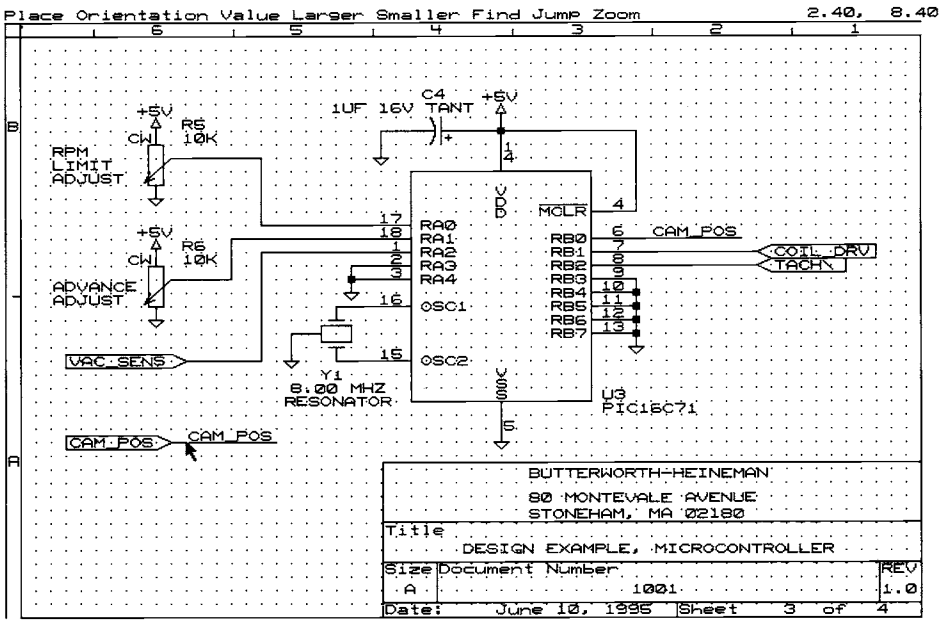


Figure 4-28 Placing the Label on Top of the Wire

Backing up Design Data

OrCAD stores all data and configuration files related to a given design in a separate directory. The OrCAD directory structure is given on page 58. All the files for the tutorial exercise are in a subdirectory named TUTOR2. The full path to this subdirectory is C:\ORCAD\TUTOR2.

When working on a design, you should backup all files in the design directory onto floppy disk at regular intervals. Generally, a backup should be done at the end of every editing session and at the end of every major stage in the design process. Failure to do so will lead to the inevitable consequence of lost data.

To backup your design data, exit OrCAD and go back to DOS. Obtain a blank floppy disk, format the disk, and label the disk TUTOR2. OrCAD design files are relatively compact and multiple designs will easily fit on a 1.4MB floppy. Use the same subdirectory name on the floppy backup as on your PC hard drive. To create the subdirectory on the floppy and copy the design files, use the following DOS commands (assume the floppy disk is the A: drive) :

A: <ENTER>

MD :\TUTOR2<ENTER>

C:<ENTER>

CD\ORCAD\TUTOR2<ENTER>

COPY *.* A:\TUTOR2\<ENTER>

This sequence of DOS commands changes to the A: drive, creates the subdirectory on the A: drive, changes back to the C: drive and then copies all the design files.

Second Session

The second session of the tutorial introduces OrCAD postprocessing tools. The two tools that will be utilized are Cleanup Schematic and Create Bill of Materials. You will learn to examine the output of these tools using a text display utility. However, before starting with the postprocessing tools, good practice dictates backing up all the design data.

Running Cleanup Schematic

Cleanup Schematic is a very simple yet useful postprocessing tool that be run whenever any drawing or editing has been done. You should also run Cleanup Schematic prior to printing, plotting, or running other postprocessing tools. Cleanup Schematic, as the name implies, catches minor drafting errors (such as overlapped wires, parts off of the grid, and improper labels) and corrects them. Another important use of cleanup is to remove error objects. Error objects are generated by the electrical designs rules checking routine. They appear as small dots. Electrical rules checking will be discussed in detail in the next chapter.

Start OrCAD, select the TUTOR2 design, and launch SDT. At the main SDT screen, click on Cleanup Schematic, as shown in Figure 4-29. Then click on Local Configuration as shown in Figure 4-30. This brings up the configuration screen for Cleanup Schematic. Verify that your configuration matches that shown in Figure 4-31.

Note that the file source is listed as "Source file is the root of the design." This option must be selected for hierarchical schematics. A similar selection box appears in most OrCAD postprocessing routines. If this option is not selected, OrCAD will only process the file for sheet one, TUTOR2.SCH. The lower level sheets will be ignored.

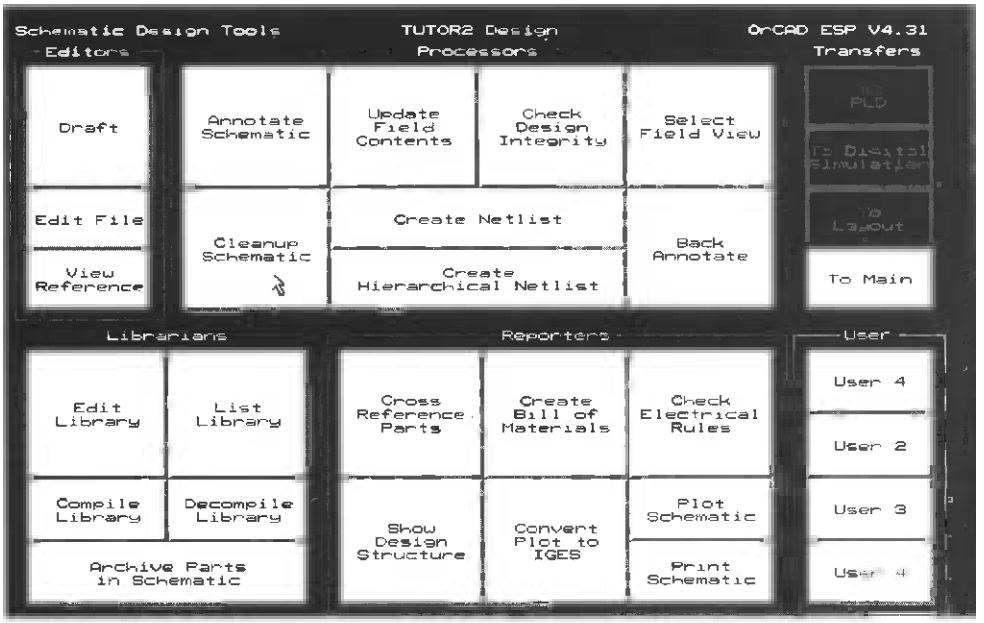


Figure 4-29 Launching Cleanup Schematic

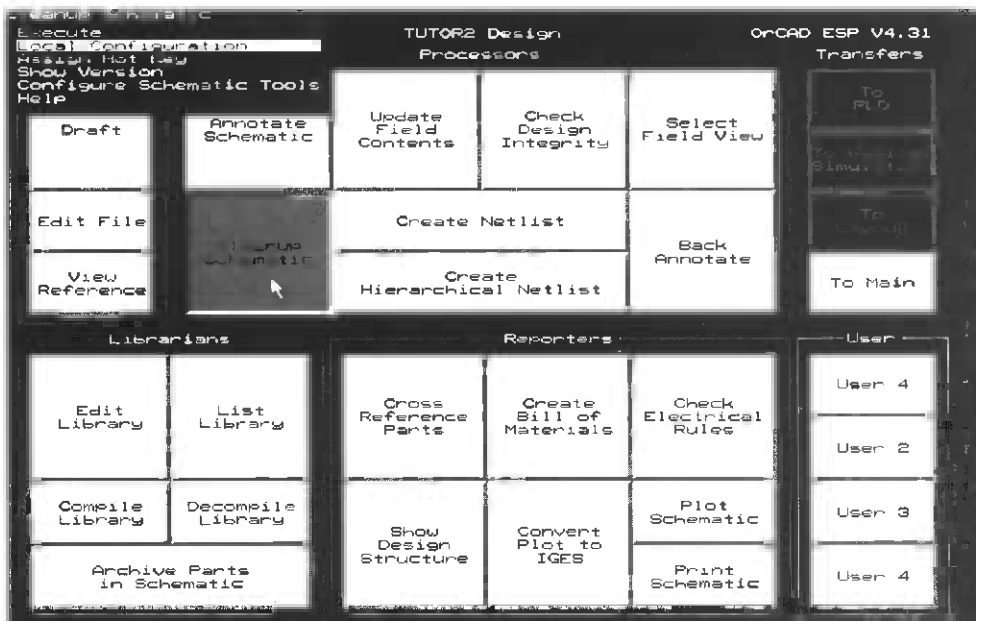


Figure 4-30 Selecting Cleanup Schematic Configuration

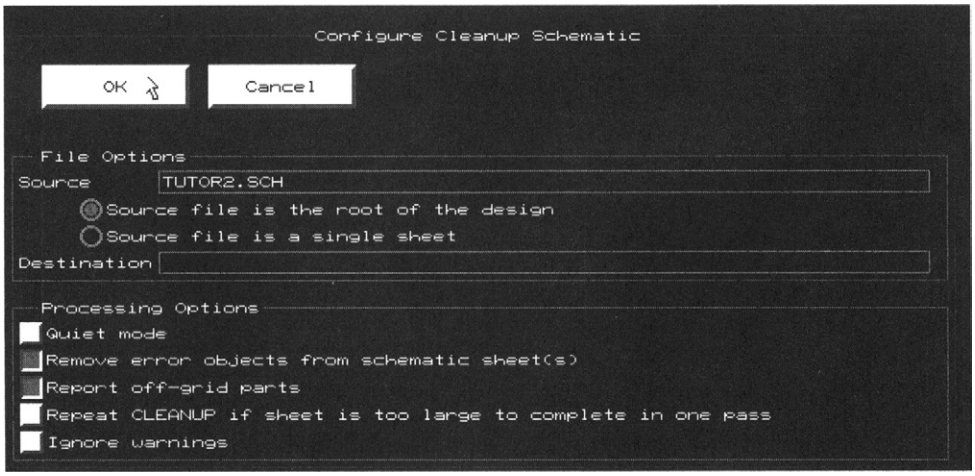


Figure 4-31 Cleanup Schematic Configuration Options

Cleanup Schematic allows several processing options explained below:

Quiet mode. This refers to text status information that appears at the bottom of the screen when Cleanup Schematic runs. Except on old 386 PCs, the text scrolls so quickly that it can't be read, but it gives some indication of program progress. Selecting quiet mode turns off the status text display. A similar option appears on most other OrCAD postprocessing routines.

Remove error objects. Error objects will be discussed in the next chapter. This option provides a convenient way to remove error objects.

Report off-grid parts. Off grid parts are a serious error condition, since it is almost impossible to connect a wire to an off grid part pin. This option should always be enabled.

Repeat Cleanup if sheet too large. This options dates back to the days of 640KB RAM. You are unlikely to ever create a sheet large enough to overflow available memory.

Ignore warnings. This allows Cleanup to bypass certain conditions, such as failure to find a library part, which would otherwise cause the program to fail. Avoid using this option, since the output may be incorrect .

After you have verified the configuration options, click on OK to exit. Then double click on Cleanup Schematic to launch the tool.

Most OrCAD tools write status messages along with information about any error conditions to an output file named #ESP_OUT.TXT in the design directory. Use

an ASCII text editor or text display utility to examine this file whenever you have run Cleanup Schematic. The disk supplied with this book includes a popular shareware text display utility, LIST.COM, that you can use to examine the #ESP_OUT.TXT file.

Exit to DOS and copy LIST.COM into your root or DOS directory, or other directory included in your PC's PATH statement. Refer to appendix B for more details, including shareware license and usage restrictions. Use LIST.COM to examine the output file by typing:

CD\ORCAD\TUTOR2<ENTER>

LIST #ESP_OUT.TXT<ENTER>

This changes to the TUTOR2 subdirectory and runs LIST.COM. The resulting screen display is shown in Figure 4-32. The first and last lines showing the file name and function key options are part of LIST.COM. Comprehensive on-line help is available. Press **F1** and explore the features of LIST.COM.

```
LIST        1              08-09-95 18:27 ♦ C:\ORCAD\TUTOR2\#ESP_OUT.TXT
-------------------------------------
Loading: C:\ORCADEXE\CLEANUP.EXE
With Parameters:  TUTOR2.SCH          /E /G
-------------------------------------
Clean Up
"OrCAD/SDT 386+ v1.10    03-SEP-93"
(C)Copyright 1991,1992,1993 OrCAD, Inc. ALL RIGHTS RESERVED.
Reading Configuration Data File:  sdt.bcf
Initial memory allocation: 2000K.
Reading Library:  C:\ORCADESP\SDT\LIBRARY\CUSTOM.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\TTL.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\DEVICE.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\ANALOG.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\CMOS.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\INTEL.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\MEMORY.LIB
Reading Library:  C:\ORCADESP\SDT\LIBRARY\POWER.LIB
Opening "TUTOR2.SCH"
Opening "INPUT.SCH"
Opening "MICRO.SCH"
Opening "OUTPUT.SCH"
Opening "TUTOR2.SCH"
Cleaning Up      "TUTOR2.SCH"
Checking Header
Checking Sheets
Checking Library Parts
Checking Module Ports
Checking Labels
Checking Power
Checking Bus Entries
Checking Junctions
Checking Wires and Buses
Checking Non-Orthogonal Wires and Buses
Checking Horizontal Wires
Checking Horizontal Buses
Checking Vertical Wires
Checking Vertical Buses
Cleaning Up      "INPUT.SCH"
Checking Header
Checking Sheets
Checking Library Parts
Checking Module Ports
Checking Labels
Checking Power
Checking Bus Entries
Checking Junctions
Checking Wires and Buses
Checking Non-Orthogonal Wires and Buses
Command►    *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 4-32 #ESP_OUT.TXT File after Running Cleanup

Note that #ESP_OUT.TXT reports which program was run (CLEANUP.EXE), the name of the design (TUTOR2.SCH), parts libraries in the SDT search path, and the progress made checking various aspects of each sheet. The file continues

past the bottom of the screen. Use the cursor keys to scroll through the entire file. Always look for any evidence of warnings or improper program termination.

Creating a Bill of Materials

OrCAD will automatically create a bill of materials. The bill of materials is an ASCII text file that includes an identification header and a parts list. The identification header contains much of the same information as in the title block area: the name of the design, revision level, drawing number, date, and time. The parts used in the design are sorted and organized into a parts list. Parts with the same description and reference designator prefix are grouped together such as all 10K resistors or .1UF capacitors. Each line in the parts listing is given a line number and contains quantity, reference designators, and part value.

The parts list is alphabetically sorted by reference designator prefix. For example, capacitors with prefix C are followed by diodes with prefix D. Unfortunately, OrCAD does not effectively sort by part value. The information in the header at the top of the parts list also appears somewhat jumbled. Some editing and cleanup with an ASCII text editor is usually required. Chapter 11 covers the use of a special sort utility that correctly sorts OrCAD bill of materials by part value. This utility, named BOMSORT.EXE, is included on the disk supplied with the book.

Start OrCAD, select the TUTOR2 design, and launch SDT. At the main SDT screen click on Create Bill of Materials. Then click on local configuration as shown in Figure 4-33. This brings up the configuration screen for Create Bill of Materials. Verify that your configuration matches that shown in Figure 4-34.

As with Cleanup Schematic, Create Bill of Materials has two source file options depending on whether the design is a single sheet or hierarchy. Select the "Source file is the root of the design" option. The file options section allow specifying a destination. This is the path and name of the ASCII bill of materials file the program creates. The file name defaults to the name of the design with extension .BOM. Unless a path is specified the file is written into the design subdirectory.

An advanced option allows merging an "include file" with the bill of materials report. The use of include files is discussed in detail in Tutorial 5. In brief, the include file can contain additional parts information, such as part numbers, which will automatically be merged into the bill of materials report. For now, just ignore this option.

Additional bill of materials processing options include:

Quiet mode. Same function as explained in the Cleanup section on page 136. This option appears on most OrCAD postprocessing tool configuration screens and is generally left deactivated, so that all status information is displayed as the program runs.

Descend into sheetpath parts. In brief, sheetpath parts are complex parts in which an entire sheet is treated as a single part. When this option is highlighted, the program will descend into a sheetpath part and explode it into the individual parts; otherwise the sheetpath part will appear as a single part. This option is not frequently used.

Place each part entry on a separate line. Another option that is not frequently used, because most companies format bill of materials so that parts with the same value are grouped together on a single line.

Verbose report. Includes the company name and address in the bill of materials header.

Header options. Two buttons select whether or not the program inserts a header on each page. A header on each page might appear useful at first glance. However, most users heavily edit OrCAD generated bill of materials including the header information and location of page breaks. A single header at the beginning of the report will save editing time later on.

Report spacing options. Two buttons select single or double spaced. While the final version of a bill of materials is usually double spaced between lines for enhanced readability, the comment made above about editing also applies here. Leaving the OrCAD generated report single spaced will generally save editing time later on.

Report unused match strings in include file. Advanced option that is grayed out (not available) unless an include file is specified under file options.

Ignore warnings. Same function as explained in the Cleanup section on page 136. This option appears on most OrCAD configuration screens and is generally left deactivated, so that the program will terminate rather than generating an incorrect report.

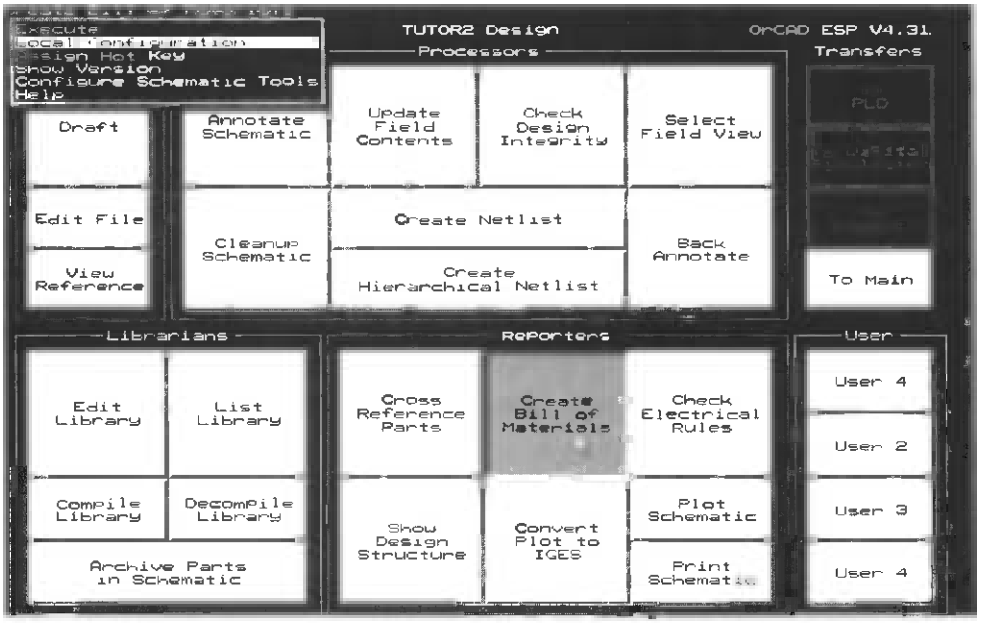


Figure 4-33 Selecting Create Bill of Materials Configuration

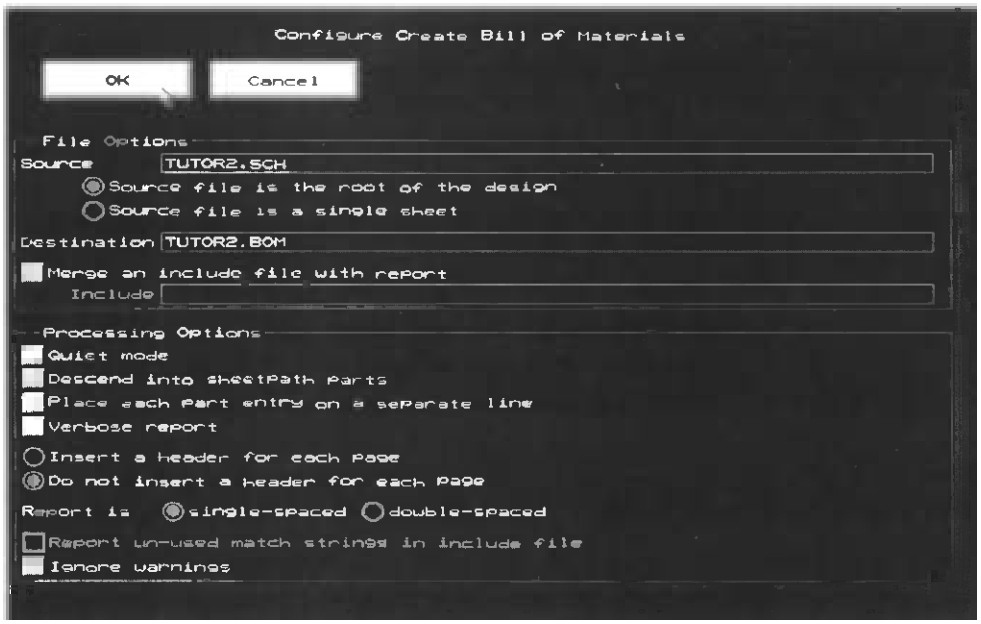
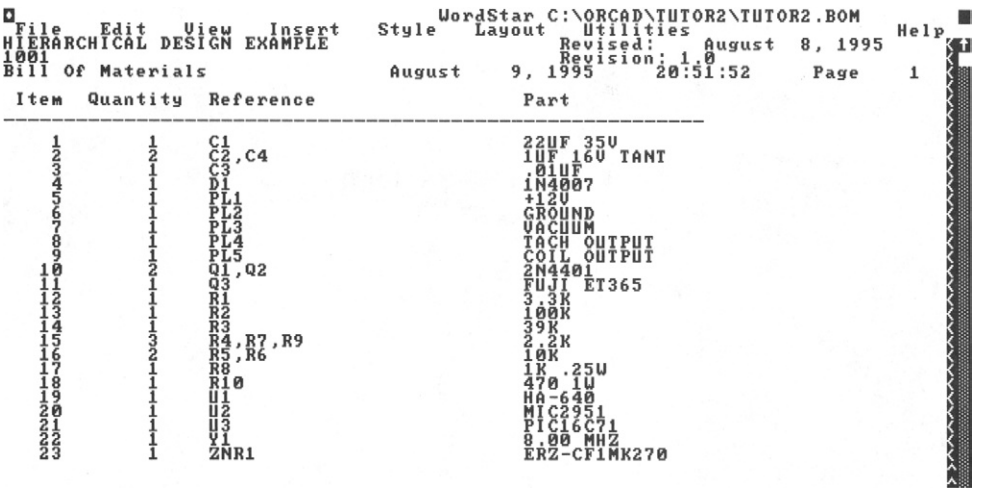


Figure 4-34 Bill of Materials Configuration Options

After you have verified the configuration options, click on OK to exit. Then double click on Create Bill of Materials to launch the tool.

As with other OrCAD tools, status messages along with information on any error conditions are written to the #ESP_OUT.TXT file. If an error condition is encountered, a screen message will appear to the effect "Program Did Not Terminate Normally." You can then use a text editor or list utility to examine #ESP_OUT.TXT for clues. Since generating a bill of materials report is relatively straightforward, error conditions are rarely encountered. Most users do not routinely examine the #ESP_OUT.TXT file after running Create Bill of Materials.

Completing the bill of materials requires editing the OrCAD output file TUTOR2.BOM using a text editor. The file is in ASCII format and most text editors and word processing programs can be used, including Microsoft Word, WordPerfect, WordStar, and many others. If you use the Windows versions of these programs, select a font that has fixed pitch (nonproportional). Otherwise, columns will not line up correctly. Figure 4-35 shows the TUTOR2.BOM bill of materials report file as it appears when loaded into WordStar for editing. Note that the top two lines and the scroll bar on the right are part of the WordStar screen.



```
WordStar C:\ORCAD\TUTOR2\TUTOR2.BOM
 File   Edit   View   Insert   Style   Layout   Utilities                Help
HIERARCHICAL DESIGN EXAMPLE                    Revised:    August  8, 1995
1001                                           Revision: 1.0
Bill Of Materials              August    9, 1995     20:51:52     Page    1

 Item  Quantity  Reference                Part
-------------------------------------------------------------

    1         1  C1                       22UF 35V
    2         2  C2,C4                    1UF 16V TANT
    3         1  C3                       .01UF
    4         1  D1                       1N4007
    5         1  PL1                      +12V
    6         1  PL2                      GROUND
    7         1  PL3                      VACUUM
    8         1  PL4                      TACH OUTPUT
    9         1  PL5                      COIL OUTPUT
   10         2  Q1,Q2                    2N4401
   11         1  Q3                       FUJI ET365
   12         1  R1                       3.3K
   13         1  R2                       100K
   14         1  R3                       39K
   15         3  R4,R7,R9                 2.2K
   16         2  R5,R6                    10K
   17         1  R8                       1K .25W
   18         1  R10                      470 1W
   19         1  U1                       HA-640
   20         1  U2                       MIC2951
   21         1  U3                       PIC16C71
   22         1  Y1                       8.00 MHZ
   23         1  ZNR1                     ERZ-CF1MK270
```

Figure 4-35 TUTOR2.BOM Bill of Materials Report

The bill of materials report generated by OrCAD includes all parts that appear on the schematic. In this tutorial exercise, the schematic represents a printed circuit board. The bill of materials should represent the printed circuit board assembly. Additional parts that do not appear on the schematic, such as the circuit board,

wire harness, IC sockets, and transistor mounting hardware must be added during the editing process. On the other hand, parts may appear on the OrCAD generated bill of materials that do not actually exist as physical components. In this example, connectors PL1-5 are not actual physical parts, but merely pads on the PCB to which the wire harness is soldered. PL1-5 should be deleted from the bill of materials.

Carefully examine the bill of materials report. Often simple errors involving part descriptions or reference designators can be spotted. Potential design improvements may also become apparent. For example, the bill of materials may show five 1.00K 1% resistors and a single 1K 5% resistor. Manufacturing costs might be reduced by making all the resistors 1% tolerance and thus eliminating the requirement to purchase, inspect, and inventory another part value.

Printing out the Schematic

Printing out hierarchical schematics is no different than printing out a single sheet schematic using the same techniques you learned in the previous tutorial. However, the local configuration for Print Schematic must be set up correctly. Go back into OrCAD SDT, and bring up the Print Schematic Local Configuration. Verify that the options are set up as shown in Figure 4-36. Note that file option is set up for a hierarchical schematic, the same as with all the other postprocessing routines in this chapter.

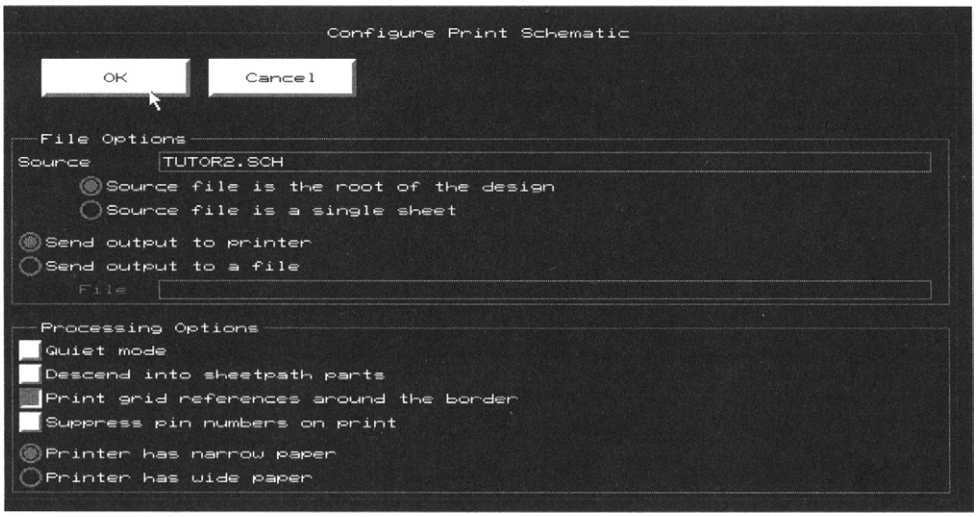


Figure 4-36 Print Schematic Configuration Options

The next two sections discuss additional postprocessing programs that relate to hierarchical designs: Cross Reference Parts and Show Design Structure. Neither

of these programs are required during normal operations and you can skip ahead without missing any important information.

The Cross Reference Parts Tool

Cross Reference Parts is a specialized postprocessing routine. Some users will find this tool helpful in locating certain errors in reference designator and module port assignments. The program can also report unused parts (gates) in multiple part packages.

Cross Reference Parts, like Create Bill of Materials, generates an ASCII text file. This file contains a listing of all parts in the design. Each part is on a separate line that includes the part description, reference designator, sheet name, sheet number, and sheet filename. A similar report can also be generated for module ports, listing the names and sheet locations.

Start OrCAD, select the TUTOR2 design, and launch SDT. At the main SDT screen click on Cross Reference Parts. Then click on local configuration to bring up the configuration screen shown in Figure 4-37.

File options function in the same manner as with other OrCAD postprocessing routines. The destination is the name of the output file. The file name defaults to the name of the design with the extension .XRF. Unless a path is specified, the file is written to the design subdirectory.

Processing options for Cross Reference Parts include:

Quiet Mode and **Descend into sheetpath parts**. Same function as with other OrCAD routines, refer to previous section.

Report only type mismatch parts and identical reference designators. This option suppresses listing every part. Only serious errors are reported. Type mismatch occurs when reference designators do not match the package type; for example, U1A and U1B appear for a part that has only one gate. An example of an identical reference designator error would be two resistors designated R1.

Report identical part reference designators. Generates a separate page listing any occurrences of identical reference designators found in the design.

Report type mismatch parts. Generates a separate page listing any occurrences of type mismatch parts found in the design.

Report unused parts in multiple part packages. This option can be used to locate unused gates. These can then be drawn on the sheet and shown with the inputs grounded if required

Report the X and Y grid coordinates of all parts. The option has limited usefulness.

List module ports. Treats module ports like parts and generates a separate page cross-referencing module ports. This option is useful for finding inconsistent module port names or type assignments. However, all errors that can be observed in the report are automatically detected by OrCAD's more comprehensive Electrical Rules Check routine.

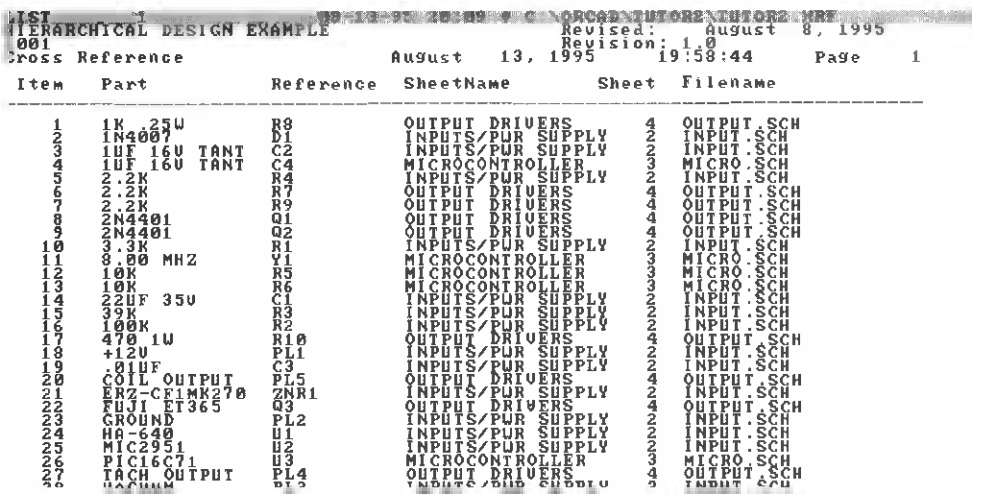
Sort options. Two buttons are provided to select the sort order, either by part value and then reference designator or by reference designator alone. The first option, sorting by part value and then reference designator provides more readable output.

Header and report spacing options and Ignore warnings . Same function as with other OrCAD routines, refer to previous section.

After you have verified the configuration options click on OK to exit. Then double click on Cross Reference parts to launch the tool. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless the program terminated abnormally, you do not need to examine this file. The report generated by Cross Reference Parts is written to the file TUTOR2.XRF.

Exit to DOS and then use the LIST.COM utility to examine TUTOR2.XRF. Figure 4-38 shows page one, the basic cross-referenced parts listing. Figure 4-39 shows pages two through four. Page two is the identical reference designator report, which shows that no errors were found. Page three is a cross-referenced listing of module ports. Page four is the reference designator type mismatch report, which shows that no errors were found.

Figure 4-37 Cross Reference Parts Configuration Options



HIERARCHICAL DESIGN EXAMPLE Revised: August 8, 1995
001 Revision: 1.0
Cross Reference August 13, 1995 19:58:44 Page 1

Item	Part	Reference	SheetName	Sheet	Filename
1	1K .25W	R8	OUTPUT DRIVERS	4	OUTPUT.SCH
2	1N4007	D1	INPUTS/PWR SUPPLY	2	INPUT.SCH
3	1UF 16V TANT	C2	INPUTS/PWR SUPPLY	2	INPUT.SCH
4	1UF 16V TANT	C4	MICROCONTROLLER	3	MICRO.SCH
5	2.2K	R4	INPUTS/PWR SUPPLY	2	INPUT.SCH
6	2.2K	R7	OUTPUT DRIVERS	4	OUTPUT.SCH
7	2.2K	R9	OUTPUT DRIVERS	4	OUTPUT.SCH
8	2N4401	Q1	OUTPUT DRIVERS	4	OUTPUT.SCH
9	2N4401	Q2	OUTPUT DRIVERS	4	OUTPUT.SCH
10	3.3K	R1	INPUTS/PWR SUPPLY	2	INPUT.SCH
11	8.00 MHZ	Y1	MICROCONTROLLER	3	MICRO.SCH
12	10K	R5	MICROCONTROLLER	3	MICRO.SCH
13	10K	R6	MICROCONTROLLER	3	MICRO.SCH
14	22UF 35V	C1	INPUTS/PWR SUPPLY	2	INPUT.SCH
15	39K	R3	INPUTS/PWR SUPPLY	2	INPUT.SCH
16	100K	R2	INPUTS/PWR SUPPLY	2	INPUT.SCH
17	470 1W	R10	OUTPUT DRIVERS	4	OUTPUT.SCH
18	+12V	PL1	INPUTS/PWR SUPPLY	2	INPUT.SCH
19	.01UF	C3	INPUTS/PWR SUPPLY	2	INPUT.SCH
20	COIL OUTPUT	PL5	OUTPUT DRIVERS	4	OUTPUT.SCH
21	ERZ-CF1MK270	ZNR1	INPUTS/PWR SUPPLY	2	INPUT.SCH
22	FUJI ET365	Q3	OUTPUT DRIVERS	4	OUTPUT.SCH
23	GROUND	PL2	INPUTS/PWR SUPPLY	2	INPUT.SCH
24	HA-640	U1	INPUTS/PWR SUPPLY	2	INPUT.SCH
25	MIC2951	U2	INPUTS/PWR SUPPLY	2	INPUT.SCH
26	PIC16C71	U3	MICROCONTROLLER	3	MICRO.SCH
27	TACH OUTPUT	PL4	OUTPUT DRIVERS	4	OUTPUT.SCH

Figure 4-38 Page One of the Cross Reference Parts Report

```
LIST      49      91      08-13-95 20:09 ♦ C:\ORCAD\TUTOR2\TUTOR2.XRF
HIERARCHICAL DESIGN EXAMPLE                    Revised:    August  8, 1995
1001                                           Revision: 1.0
Cross Reference                  August   13, 1995     19:58:44     Page    2

 Identical References
-------------------------------------------------------------------------------

 Total Duplicate Pairs Encountered :     0


HIERARCHICAL DESIGN EXAMPLE                    Revised:    August  8, 1995
1001                                           Revision: 1.0
Cross Reference: Module Ports    August   13, 1995     19:58:44     Page    3

 Item   Part          Reference  SheetName        Sheet  Filename
-------------------------------------------------------------------------------

    1   CAM_POS       Output     INPUTS/PWR SUPPLY   2  INPUT.SCH
    2   CAM_POS       Input      MICROCONTROLLER     3  MICRO.SCH
    3   COIL_DRV      Output     MICROCONTROLLER     3  MICRO.SCH
    4   COIL_DRV      Input      OUTPUT DRIVERS      4  OUTPUT.SCH
    5   TACH\         Output     MICROCONTROLLER     3  MICRO.SCH
    6   TACH\         Input      OUTPUT DRIVERS      4  OUTPUT.SCH
    7   VAC_SENS      Output     INPUTS/PWR SUPPLY   2  INPUT.SCH
    8   VAC_SENS      Input      MICROCONTROLLER     3  MICRO.SCH


HIERARCHICAL DESIGN EXAMPLE                    Revised:    August  8, 1995
1001                                           Revision: 1.0
Cross Reference: Module Ports    August   13, 1995     19:58:44     Page    4

 Type Mismatch
-------------------------------------------------------------------------------

 Total Mismatches Encountered :     0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 4-39 Pages 2-4 of the Cross Reference Parts Report

The Show Design Structure Tool

You should never get into a situation where you have to use the Show Design Structure tool. The purpose of the tool is to show the structure (levels) of a hierarchy. Bluntly stated, if your design is so complex that you have to run this tool to get an overview of the structure, you had better reorganize your design.

Show Design Structure can be launched from the main SDT screen. The routine has a minimal local configuration screen which is shown in Figure 4-40. The file and processing options function in the same manner as with other OrCAD routines. Note that the output file name defaults to the name of the design with the extension .TWG. Figure 4-41 shows the output file contents examined with the LIST.COM utility.

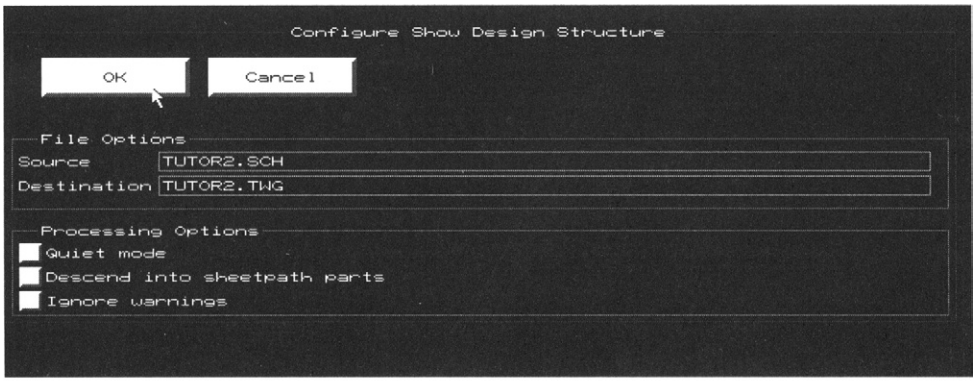


Figure 4-40 Show Design Structure Configuration Options

```
LIST      1       9      08-13-95 19:59 ♦ C:\ORCAD\TUTOR2\TUTOR2.TWG
  <<<root>>>
  [TUTOR2.SCH]     August  8, 1995
    INPUTS/PWR SUPPLY
    [INPUT.SCH]       June 10, 1995
    MICROCONTROLLER
    [MICRO.SCH]       June 10, 1995
    OUTPUT DRIVERS
    [OUTPUT.SCH]        May 23, 1995
------------------------------
```

Figure 4-41 TUTOR2 Design Structure Report

Conclusion

You have now completed the second tutorial. At this point you have learned how to create hierarchical schematics and how to use some of the postprocessing tools, including generating a bill of materials report.

5

Tutorial 3 - Postprocessing

At this point, you have learned most of the basic skills required to capture schematics using OrCAD. The third tutorial exercise continues to build on these skills. The focus shifts from merely drawing a schematic to using OrCAD's powerful postprocessing tools. You will learn how to use these tools to check for electrical errors and to generate a netlist for input to a printed circuit board design system.

Starting the Third Tutorial

Use Figures 5-1 through 5-3 as the model for this tutorial exercise. Make a copy of these figures and keep them handy for reference. The tutorial is divided into three sessions. Plan on spending several hours on each session.

In the first session, you will draw a three page hierarchical schematic for a PC bus interface card. The first session introduces data bus concepts and covers the Draft menu in greater detail, with special emphasis on the Place command.

The second session introduces the Annotate Schematic, Check Electrical Rules, and Plot Design routines. Annotate Schematic automatically assigns reference designators. Check Electrical Rules searches for design rule violations. Plot Design generates hard copy on large format plotters.

The third session covers Netlist postprocessing using the Create Netlist routine. The session includes an in-depth discussion of netlist concepts.

First Session

Start the third tutorial by creating a new design called TUTOR3, based on the standard template. Use the OrCAD Design Management tools as explained in chapter 3. Then select TUTOR3 and launch OrCAD SDT and Draft.

Draw the hierarchy (sheet 1) using Figure 5-1 as a model. Apply the techniques learned in chapter 4 to place the sheet objects, sheet nets, wires, and edit the title block. The sheet nets for the data bus must be named D[0..7] exactly as shown.

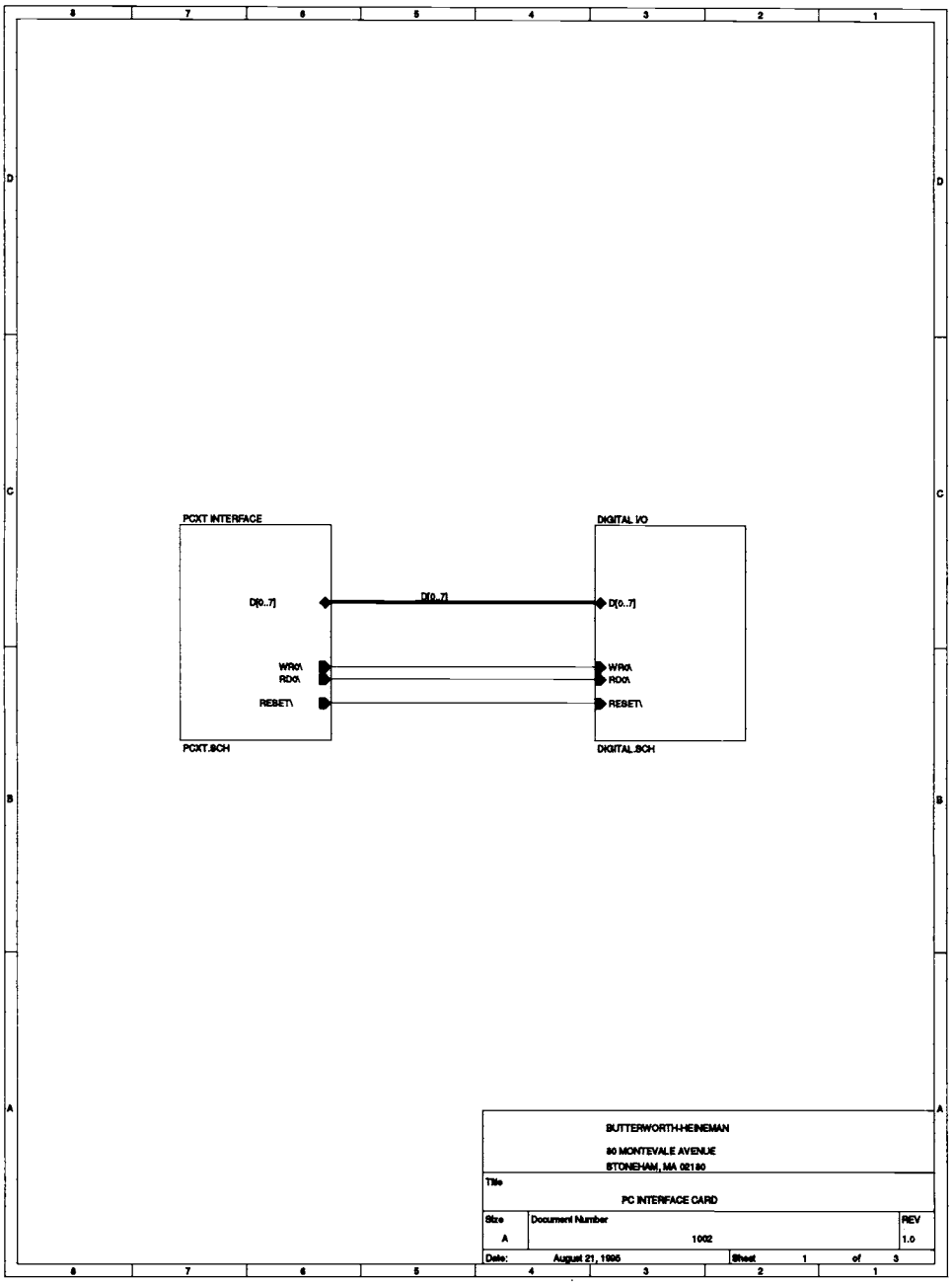


Figure 5-1 Third Tutorial Sheet One

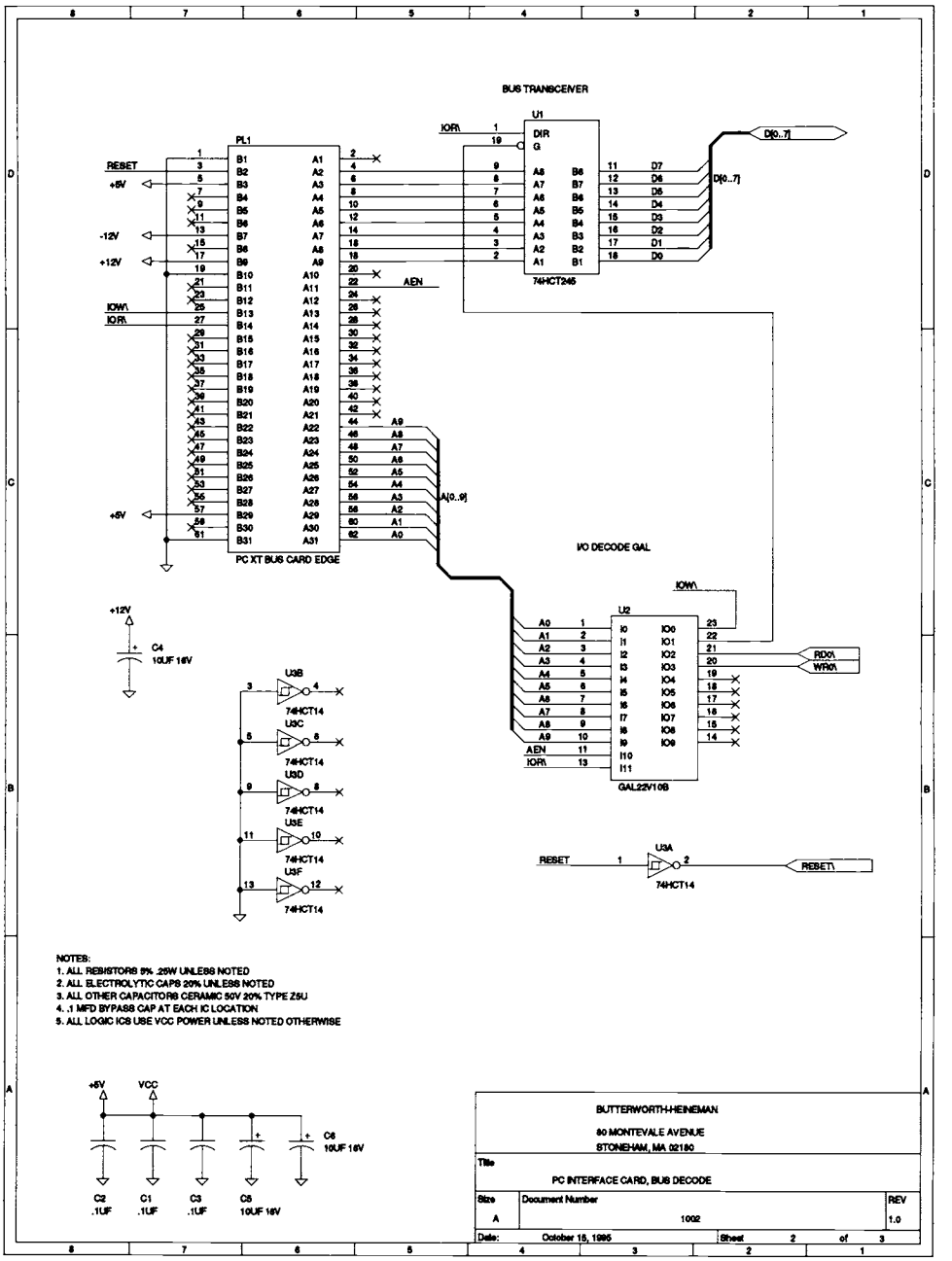


Figure 5-2 Third Tutorial Sheet Two

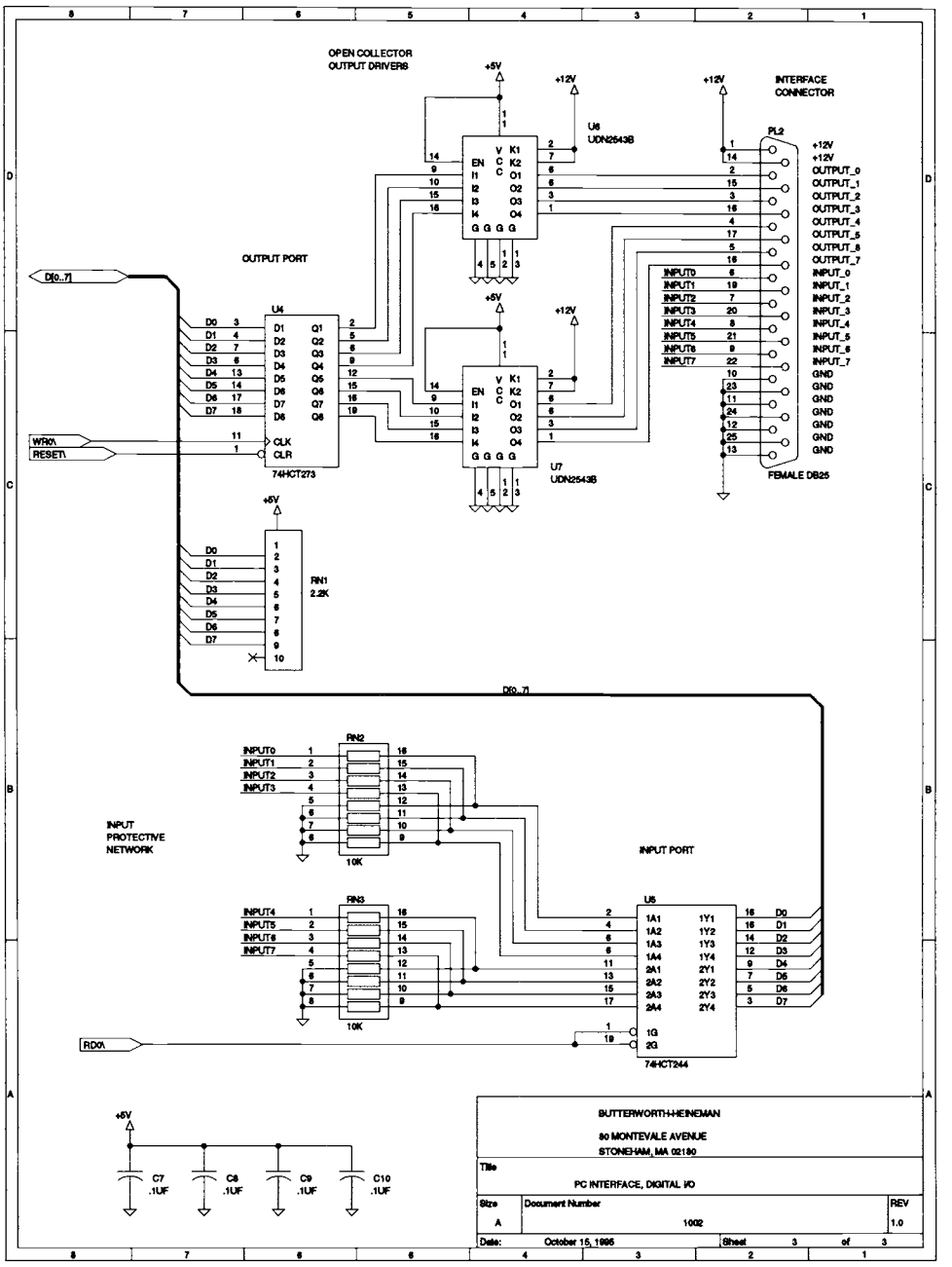


Figure 5-3 Third Tutorial Sheet Three

The next step is to draw the data bus. Bus objects are drawn similar to wires. Because no macro has been defined for drawing bus objects, you must use the Place command.

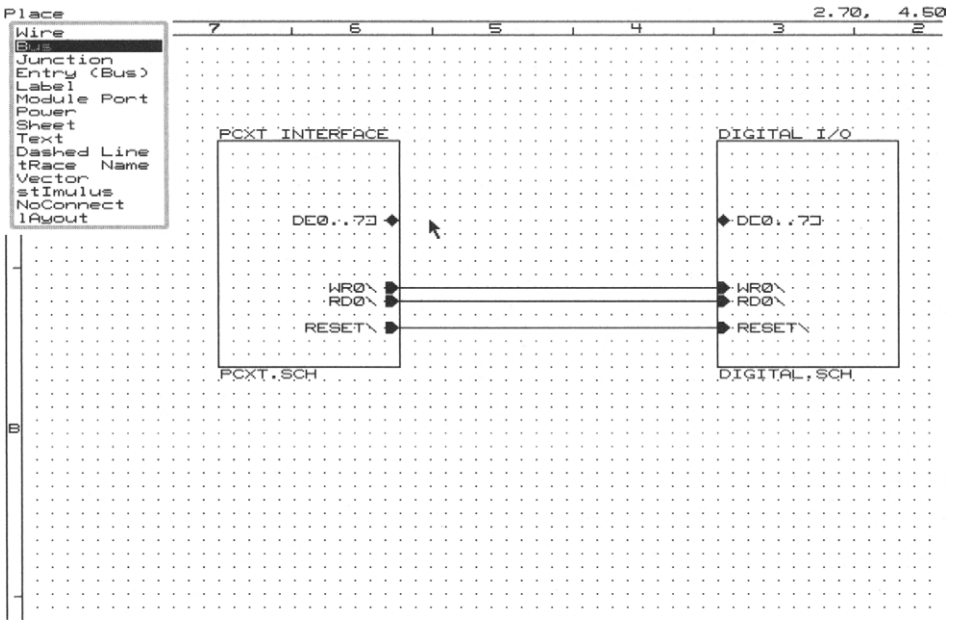


Figure 5-4 The Place Menu

Overview of the Place Command

So far you have drawn wires, junctions, sheet objects, text, labels, and power objects. All of these objects appear as options on the menu for the Place command. Wires, junctions, and power objects have been drawn using macro commands, which are more convenient for frequent tasks. Let's examine the Place menu and the other types of objects that appear on it in more detail. The Place menu is shown in Figure 5-4. The following objects appear on the Place menu:

Wire Electrical connections between pins and objects.

Bus A means of representing multiple signal wires. Buses should be used for all data and address lines and any other signals that can be logically grouped together. Using buses improves readability and reduces clutter. Refer back to the section in chapter 1 for an introduction to bus concepts. Bus objects appear as thick lines.

Junction	Electrical interconnection point for crossing wires.
Entry (Bus)	Used to connect wires to buses. Appear as a short 45 degree diagonal line. Two styles appearing as / or \ are available.
Label	A means of identifying signals. Can also be used to electrically join wire sections and route signals between different areas on a given sheet.
Module Port	Used to route signals between sheets in hierarchical designs. Must correspond to sheet nets.
Power	Used to connect wires and pins to power planes. Several styles and orientations at 90 degree increments are available. Power objects are named the same as the corresponding power plane.
Sheet	Used in hierarchical designs. Can be thought of as representing the circuit blocks in a block diagram.
Text	Used to annotate schematics with miscellaneous information such as notes. No electrical properties.
Dashed Line	Like text, dashed lines can be used to annotate schematics and do not have any electrical properties. Common uses are to outline functional circuit blocks within a sheet or to show hand wired connections used to implement engineering changes on a printed circuit board.
Trace Name	Advanced feature for digital timing simulation software. Identifies a node to be traced (timing analyzed).
Vector	Advanced feature for digital timing simulation software. Associates a test vector with a particular node.
Stimulus	Advanced feature for digital timing simulation software Associates a stimulus signal with a particular node.
NoConnect	Special object placed on unconnected pins or wires to identify them as deliberately unconnected and to prevent an error from being flagged by the Electrical Rules Check routine. Appears as a small X.

Layout Advanced feature used in conjunction with OrCAD PCB design software to associate a layout directive (such as trace width or via type) with a particular node.

In this tutorial, you will learn to use the Bus, Entry, and NoConnect objects.

Bus Considerations

Bus objects are used to represent multiple signals thereby eliminating clutter and improving readability. Good practice dictates always using bus objects to represent data and address signals that are routed between more than two ICs or that are routed between sheets.

Special rules apply to naming and labeling buses and bus signals, routing signals to and from buses, and routing buses between sheets:

- **Bus names**. Labels are used for naming buses. Every bus must have an associated label. The label name is of the form NAME[A..B] where NAME is any legal alphanumeric label name, and A and B are integers that represent the wire numbers branching to and from the bus. Note that square brackets [ ] must be used and the numbers must be separated by two periods. Examples of valid names include: D[0..7] and ADDR1[16..31]. A must be less than B. Negative numbers are not allowed. The label must be placed with the "hot point" (lower left hand corner) touching the bus.
- **Signal names**. The signals branching from a bus have names based on the bus name. For example, data bus D[0..7] has eight data signals D0, D1, through D7. A given signal can branch from a bus any number of times.
- **Bus entry**. Signals can only be routed to or from a bus via a special entry object which appears as a 45 degree diagonal slash. Wires that cross a bus or end at a bus are not electrically connected to the bus unless an entry object is used. Junctions cannot be used to connect pins or wires to a bus.
- **Bus routing**. Sections of the same bus can be joined together in a "Y" or can "cross" without junctions (junctions have no effect on bus objects). Different buses that cross one another do not join, just as with wires. Bus are routed between sheets by module ports, again just as with wires, except the name of the module port must correspond exactly with the bus name.
- **Drawing conventions**. To help differentiate buses from wires, buses are traditionally drawn with 45 degree beveled corners. A 45 degree split is generally preferred at the point where a bus splits into two separate sections. Two styles of bus entry appearing as / or \ are available. Pick an entry

style that flows with the direction of the bus. Use Figures 5-2 and 5-3 as examples of proper flow.

Completing the First Sheet

To complete the first sheet, draw the bus between the two sheet objects as shown in Figure 5-1. Bring up the Place menu as shown in Figure 5-4. Then click on Bus. The Bus menu appears as shown in Figure 5-5. Locate the cursor at the starting point on the edge of the left sheet object. Click the left mouse button to begin the bus. Move the cursor to the edge of the right sheet object and click the left mouse button again to place a corner point. Then click the right mouse button to escape. The last corner point becomes the end point of the bus as shown in Figure 5-6. As the last step, place the bus label using the name D[0..7] as shown in Figure 5-7. The label must touch the bus. Sheet one is now complete.

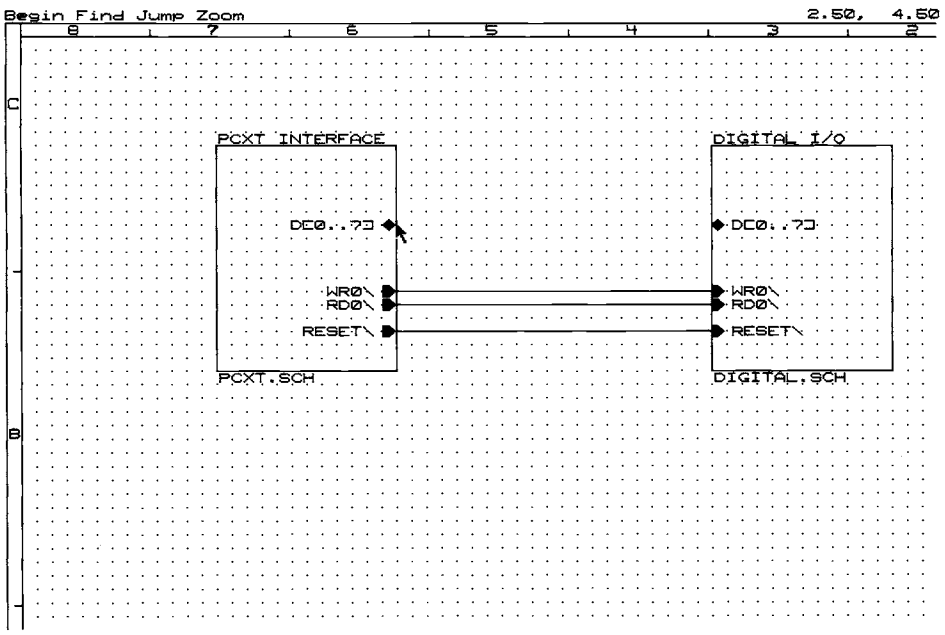


Figure 5-5 Starting the Bus

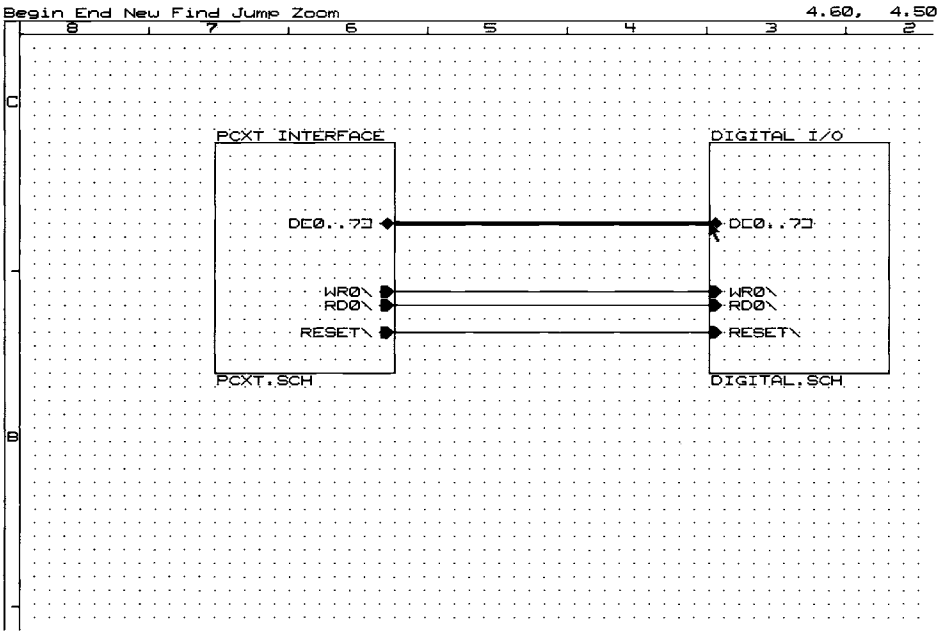


Figure 5-6 Ending the Bus

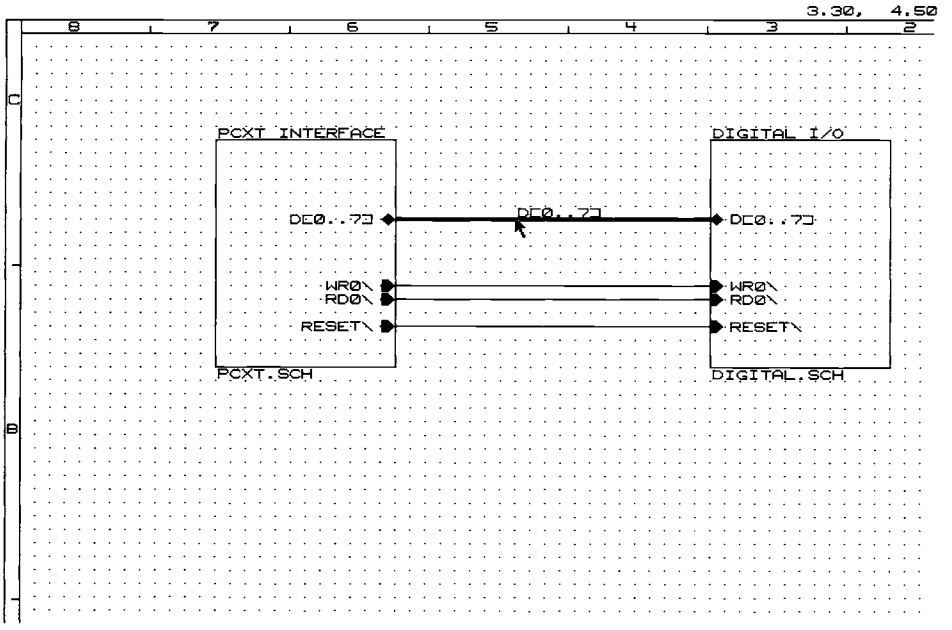


Figure 5-7 Labeling the Bus

Starting the Second and Third Sheets

Use the skills you learned in the previous tutorial exercise to start the remaining two sheets. At this point you should be able to enter the sheets, place parts, power and ground objects, wires, junctions, labels, and text, and edit the part descriptions and title block areas. Leave the reference designators alone for now, as you will be learning to use the Annotate Schematic postprocessing routine which automatically numbers the reference designators. Also leave the buses, bus entries, and no connects for now, as these objects will be covered in more detail later on.

Listed below are some hints on new parts:

- **IBM PC XT bus card edge connector**: CONNECTOR IBM PC XT from CUSTOM.LIB.
- **Female DB 25 pin connector**: CONNECTOR DB25 from DEVICE.LIB. Rotate twice and then mirror before placing.
- **74HCT series logic ICs**: these are in the TTL logic library TTL.LIB (since they have TTL family pinouts). You can use the Get command and enter the suffix number, that is 244 for 74HCT244. You will then get a menu of choices for different versions (such as 74LS244 or 74HCT44). Use the 74HCT14 from CUSTOM.LIB, which is a smaller part than that in TTL.LIB.
- **GAL22V10B generic array logic IC**: GAL22V10B from CUSTOM.LIB.
- **UDN2543B power driver IC:** UDN2543B from CUSTOM.LIB.
- **Resistor networks**: 10 pin SIP is RN10 and 16 pin DIP is RN16 DIP ISO, both from CUSTOM.LIB.

Power Pins

You will recall from chapter 1 that OrCAD does not show visible ground and power pins for many logic ICs. Most TTL logic ICs have ground pins defined as GND and power pins defined as VCC. CMOS ICs use VSS and VDD. The 74HCT series ICs in the tutorial are CMOS versions of old TTL parts and still use the original TTL pin names. The GAL22V10B also uses GND and VCC. In some cases, the only way to verify names for invisible ground and power pins is to examine the part with the library editor, a topic that will be examined in the next tutorial. With today's trend towards multiple voltage designs, invisible ground and power pins are sometimes a real nuisance.

Invisible ground and power pins are automatically tied to ground and power planes with the same name. All the parts in the tutorial with invisible pins use

GND and VCC. GND pins are no factor, because all the ground objects are also named GND and tied to the same plane. However the +5 volt power supply is a different situation. Visible +5 volt power objects on bypass capacitors and pull up resistors are named +5V, as is the incoming +5 volt supply on the card edge connector. The VCC plane must be connected this +5 volt plane. This can be done by placing a visible VCC power object on the sheet and connecting it to a +5V object. Refer to the lower left hand corner of Figure 5-2.

To place the VCC power object, just use macro F3 to place a +5V power object and then edit the name.

An alternate approach would have been to use power objects named VCC throughout the entire schematic and then add a note indicating that the VCC supply is +5V. This was once common practice. With today's trend towards multiple voltage levels, using the actual power supply voltage as the name for the power object reduces the chance of misinterpretation.

Introduction to the Set Menu

Before you start drawing the data buses on sheets two and three, certain SDT display and drafting options need to be set and verified. The Set command is used for this purpose. Select the Set menu by clicking on Set from the main menu as shown in Figure 5-8. This brings up the Set menu as shown in Figure 5-9. Note that most of the menu options are followed by a YES/NO indication which shows the status of the option. Options that do not include a status indication have an additional submenu. A detailed description of the various Set options follows:

Auto Pan Enables automatic panning (movement of the display window relative to the sheet) when the mouse is moved to the edge of the display window. Normally on. An archaic option from the days of slow 286 and 386 class PCs. Setting Auto Pan off slightly speeds up the display, but makes moving around the sheet very difficult. Always set Auto Pan to YES.

Backup File Enables automatic creation of a backup file whenever the Quit command is used to write to or update a schematic file that already exists. The original file is saved with the filename extension .BAK. Always set Backup File to YES. If a serious error, system crash or other unforeseen event occurs, the backup file allows recovery of data.

Drag Buses Enables rubberbanding of buses when using the Block Drag command. Another archaic option related to graphics performance. Always set Drag Buses to YES.

Error Bell Enables the error beep tone. OrCAD has an annoying tendency to flag even relatively innocuous actions as errors. For example, trying to select an object for editing and not having the mouse cursor within the object border. When an error occurs, everything is locked up for about one second and you must then click the right mouse button to escape. The error beep tone helps you recognize the occurrence of an error.

Left Button Enables automatic generation of an <ENTER> keystroke when the left mouse button is released. If this option is disabled, many commands will require double mouse clicks.

Macro Prompts Displays the sequence of commands contained in the macro when a particular macro is executed. This is another archaic option related to graphics performance. Always set Macro Prompts to YES. If Macro Prompts is set to NO, screen redraw during macro execution is disabled and any macros using pan or zoom will fail.

Orthogonal When Orthogonal mode is set to YES, wires and buses can only be drawn in the vertical or horizontal direction. This can be a significant time saver since OrCAD automatically places a corner if the mouse cursor is moved in an "L" motion. Generally accepted practice is to draw all wires in orthogonal mode. To help differentiate buses from wires, corners on buses are generally drawn with a 45 degree bevel. Temporarily set Orthogonal mode to NO while drawing buses.

Show Pins This option actually means "show pin numbers." Normally set to YES. Setting Show Pins to NO eliminates display (and hard copy) of pin numbers on library parts. Pin names are not affected. The only circumstance where display of pin numbers is not required would be an abstract schematic used for instructional or reference purposes and not intended to be built up as a functioning circuit.

Title Block Enables display and hard copy of the title block area. Generally accepted practice requires that all engineering drawings include a title block, so this option is normally set to YES.

Worksheet Size Sets the size of the worksheet area. Valid entries are A through E. The actual worksheet dimensions and associated parameters are defined in the template table during the configuration process (refer back to chapter 2).

X,Y Display Enables display of X,Y coordinates. The OrCAD coordinate system is somewhat of an oddity. Unlike normal Cartesian coordinates, OrCAD coordinates define the zero (origin) point at the upper left corner. X and Y coordinates increase going to the lower right corner. X,Y display is normally set to YES.

Grid Parameters This command has a sub-menu that is shown in Figure 5-7 and discussed in detail further on.

Repeat Parameters This command has a submenu with four options used to set parameters for the Repeat command on the main menu and to allow automatic incrementing of label and module port numeric suffixes, that is DATA0, DATA1. The intention is to save time when dealing with very wide address and data buses. The tradeoff point between time savings and the overhead of setting up this feature is probably around 16 bits. You can experiment with this feature and draw your own conclusions.

Visible Lettering This command only applies to zoom scale 2. Unless you are using a very large monitor and high resolution graphics board, zoom scale 2 is almost useless. Visible Lettering has a submenu that allows selecting what types of text, that is pin numbers, labels, and so forth, are still displayed at zoom scale 2.

Cursor Style Selects the cursor style: either an arrow or a crosshair.

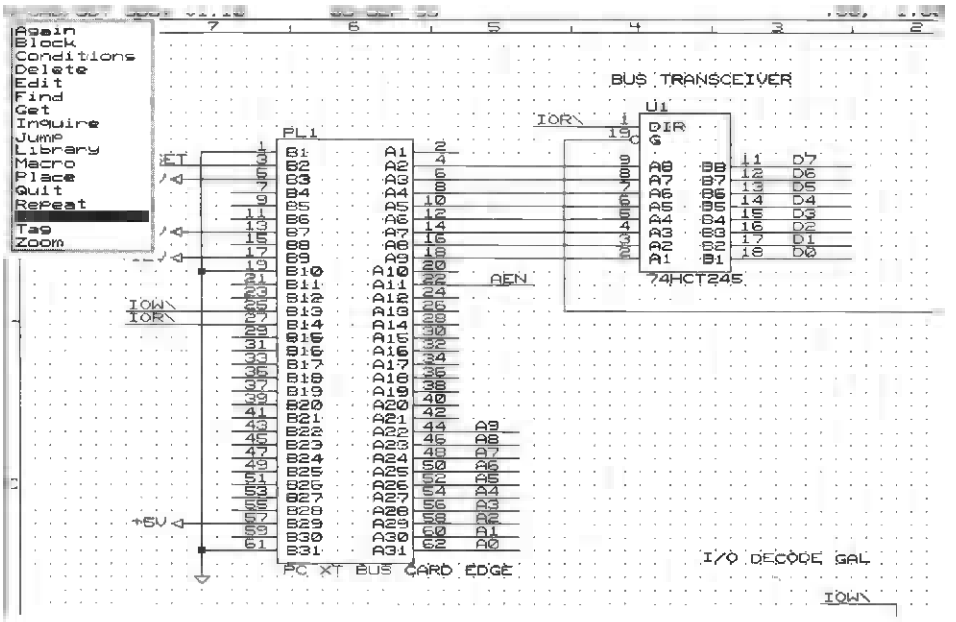


Figure 5-8 Selecting the Set Command

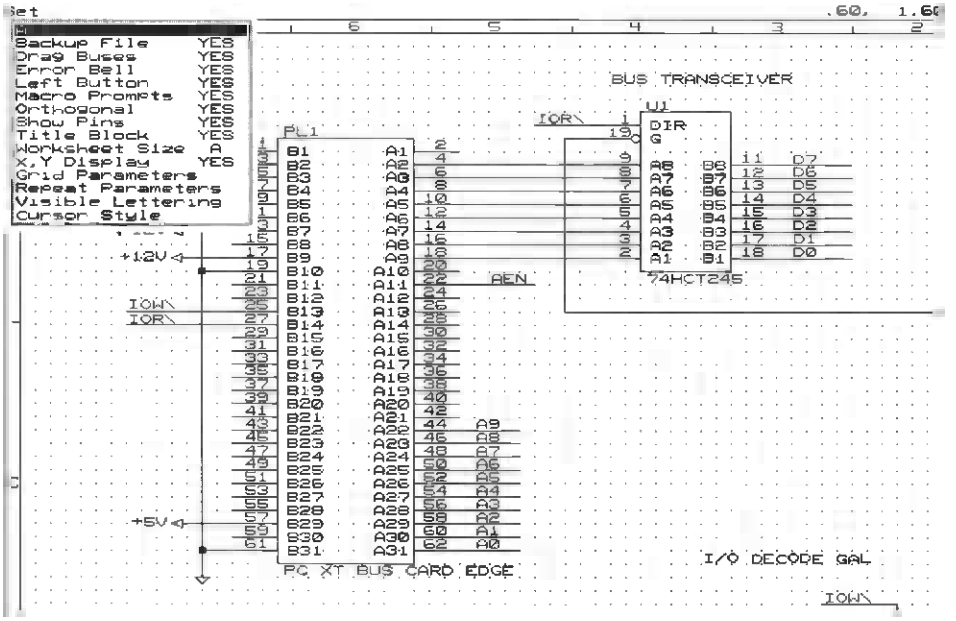


Figure 5-9 Set Menu Options

Setting Orthogonal Mode

Since the next step in the tutorial is to draw the data bus and data buses are normally drawn with 45 degree beveled corners, orthogonal mode must be disabled. Click on Orthogonal as shown in Figure 5-10 and then click on NO as shown in Figure 5-11. Use the same procedure for any other options on your system that do not agree with the settings shown in Figure 5-10.

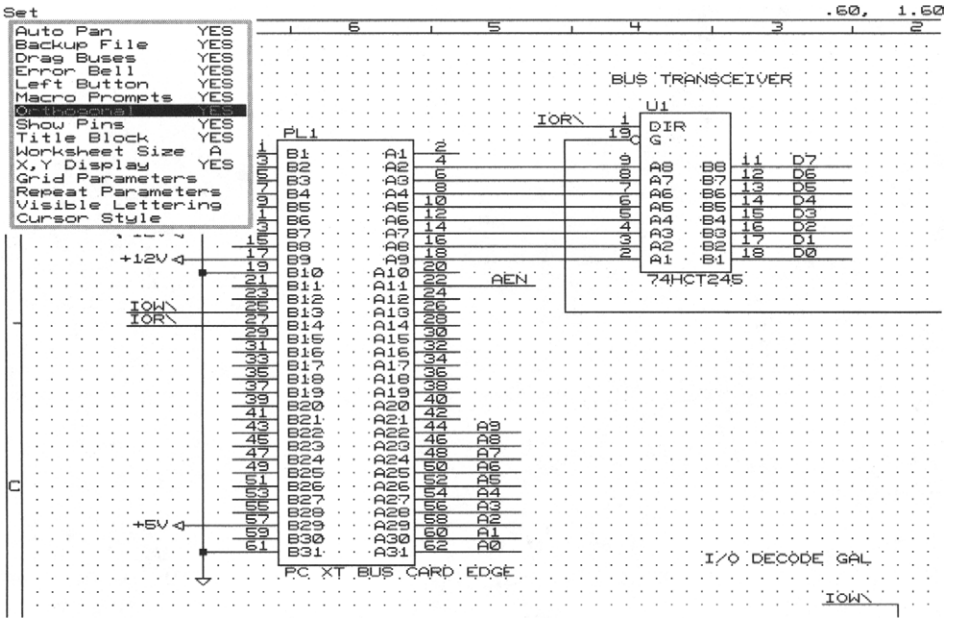


Figure 5-10 Selecting the Orthogonal Option

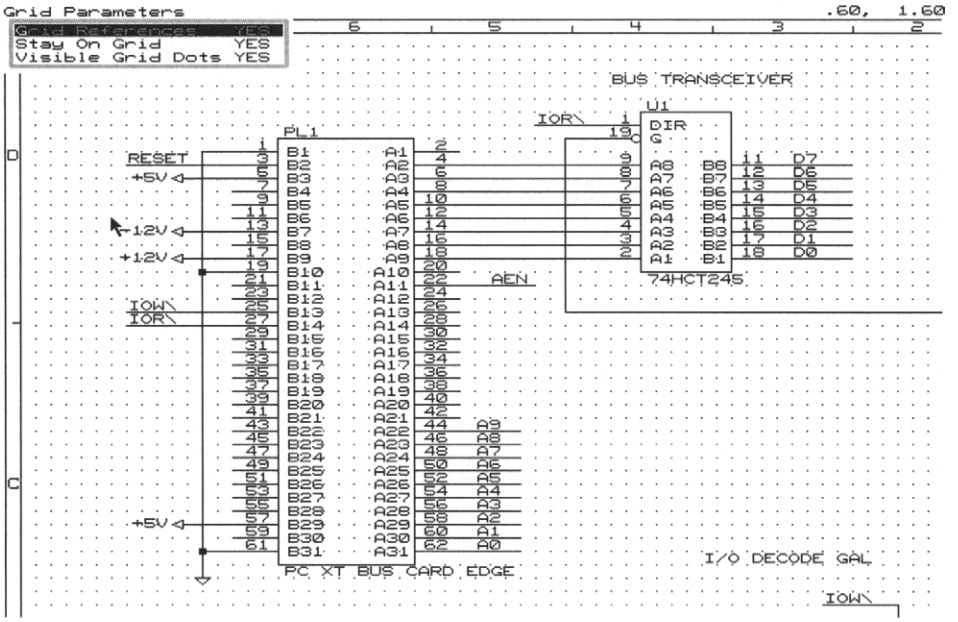


Figure 5-11 Setting Orthogonal Mode to Off

Setting Grid Parameters

Figure 5-12 shows the options for setting grid parameters. Click on Grid Parameters from the Set menu and verify that your options are set up as shown in the figure:

Grid References Enables display of alphanumeric grid references on the top and left borders of the display. Does not affect hard copy. Useful for maintaining your orientation on the sheet.

Stay on Grid Forces the cursor to stay on grid when placing objects. Caution! Never place any objects off grid. Always leave this option set to YES. Placing electrical objects off grid may cause serious errors. Connections between objects located off grid can appear intact, yet not be recognized by postprocessing routines such as Check Electrical Rules and Netlist. In some cases text (but never labels) might be located off grid to provide better alignment. However, if text or any other object is located off grid, locating the cursor on the object for editing or deletion

becomes very difficult. In most cases Delete Block or Block Move/Drag are the only commands that can find and manipulate off grid objects.

Visible Grid Dots Enables display of grid dots. Visible Grid Dots should always be set to YES. Note that this option has no effect on whether or not objects are placed on grid. At the normal zoom scale 1, the spacing between grid dots is 1/10 of X and Y units. For example, configuring SDT for inch units results in 10 grid dots per inch at zoom scale 1.

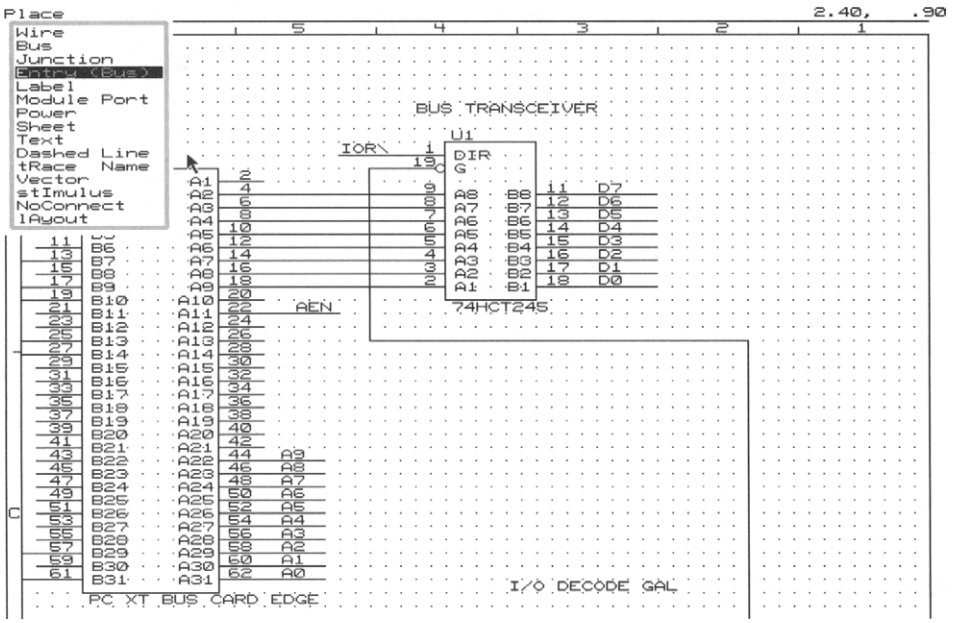


Figure 5-12 Grid Parameter Options

Completing the Buses

Your last task in the first session will be to complete the buses on sheets two and three. Orthogonal mode has been turned off to allow drawing the beveled corners on the buses.

Let's use the data bus at the top right side of sheet two as an example. First, place the bus entry objects. Refer to Figure 5-13. Click on the Entry (Bus) command on the Place menu. Holding down the left mouse button brings up the Entry sub-

menu shown in Figure 5-14. Highlight and select the forward slash (/) style entry object. The entry object appears on the sheet as a ghost image (looks like a dotted line) and can be positioned with the mouse cursor. Position the first entry as shown in Figure 5-15. Click the left mouse button to place the first entry object. Then place the remaining seven entry objects as shown in Figure 5-16.

The next step is to draw the data bus. Drawing buses is similar to drawing wires, except that no macro command has been defined for buses. Select the Bus command from the Place menu, locate the cursor at the starting point for the bus as shown in Figure 5-17 and click the left mouse button. Then complete the bus, clicking the left button at each corner point and at the end point. Click the right button to escape when you are done. The last step is to place the module port and bus label as shown in Figure 5-18.

Complete the remaining data and address buses on sheets two and three using the same techniques. Note that some of the entry objects are drawn using the backslash (\) style for better flow. This is strictly a cosmetic issue.

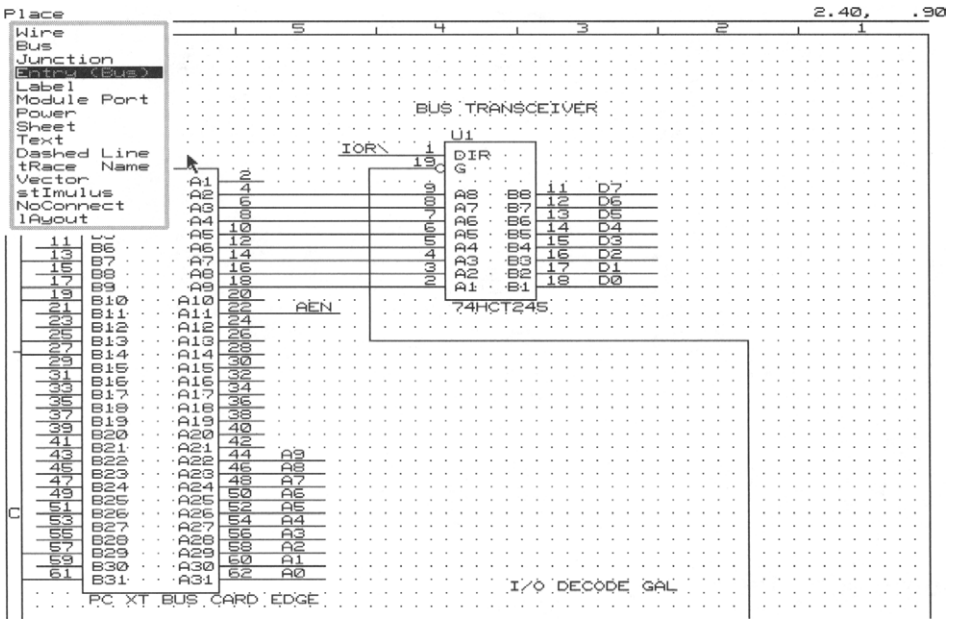


Figure 5-13 Selecting the Place Entry Object Command

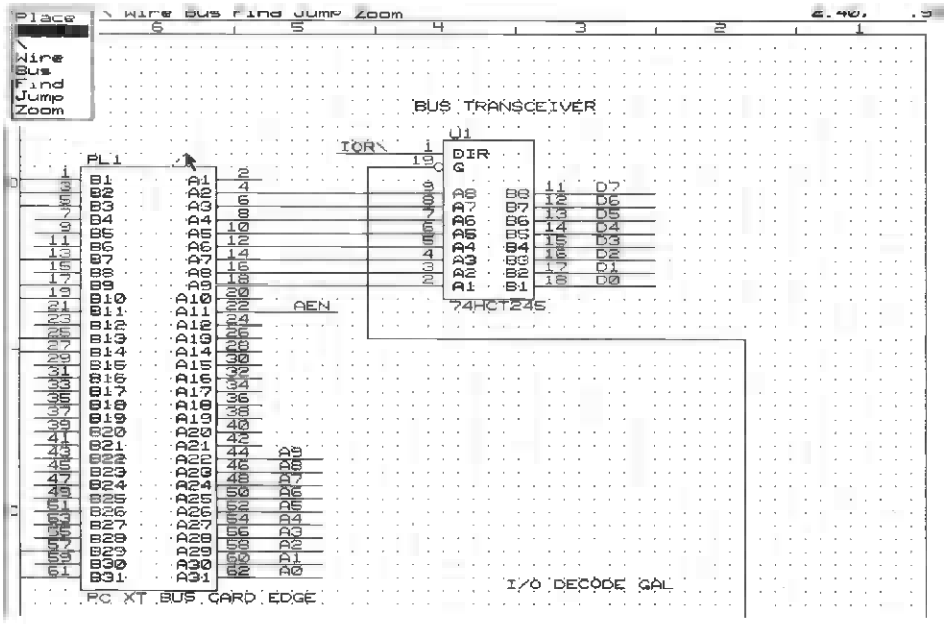


Figure 5-14 Selecting the Forward Slash Entry Object

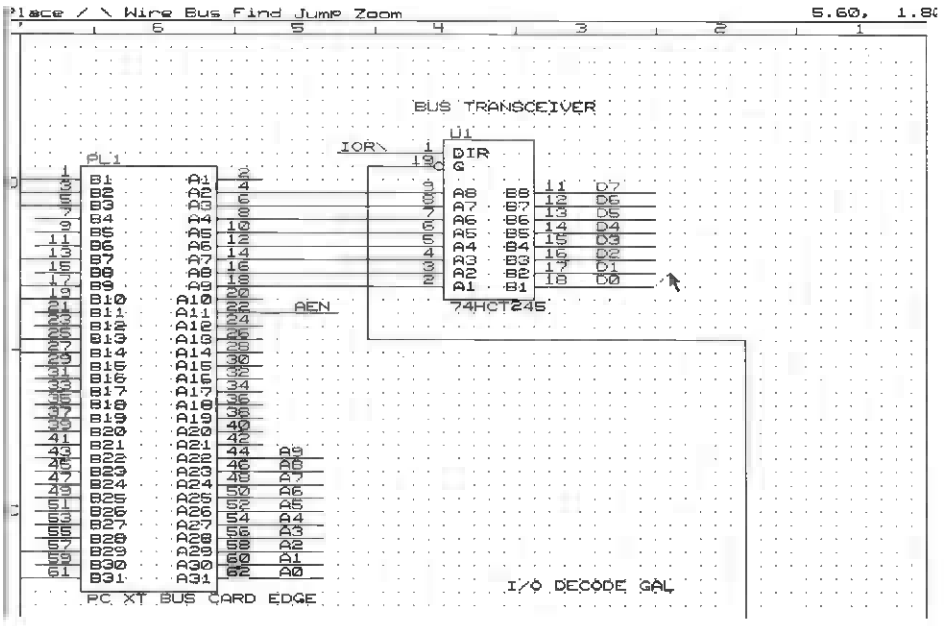


Figure 5-15 Placing the First Entry Object

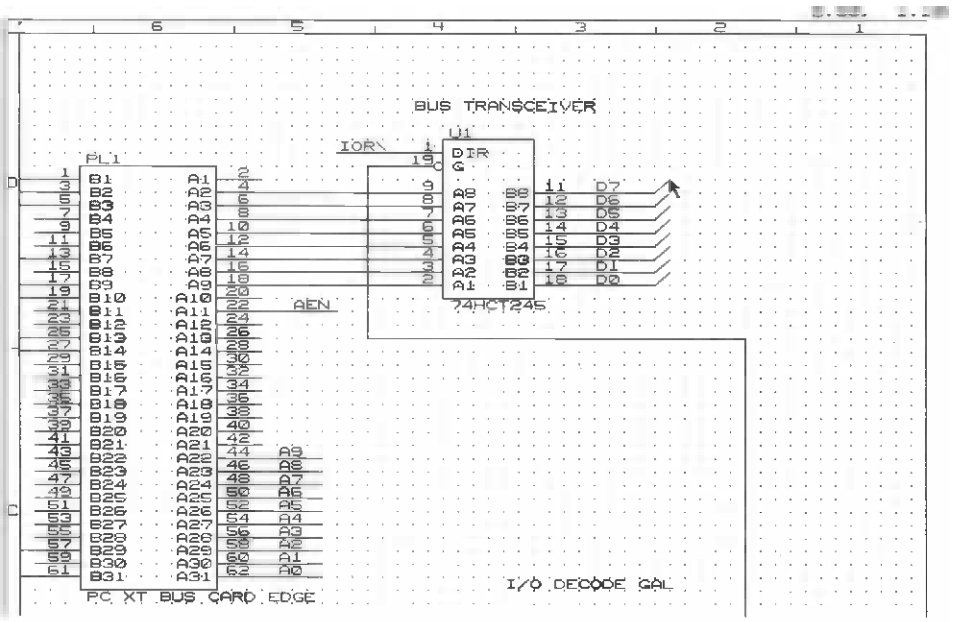


Figure 5-16 Completed Placement of Entry Objects

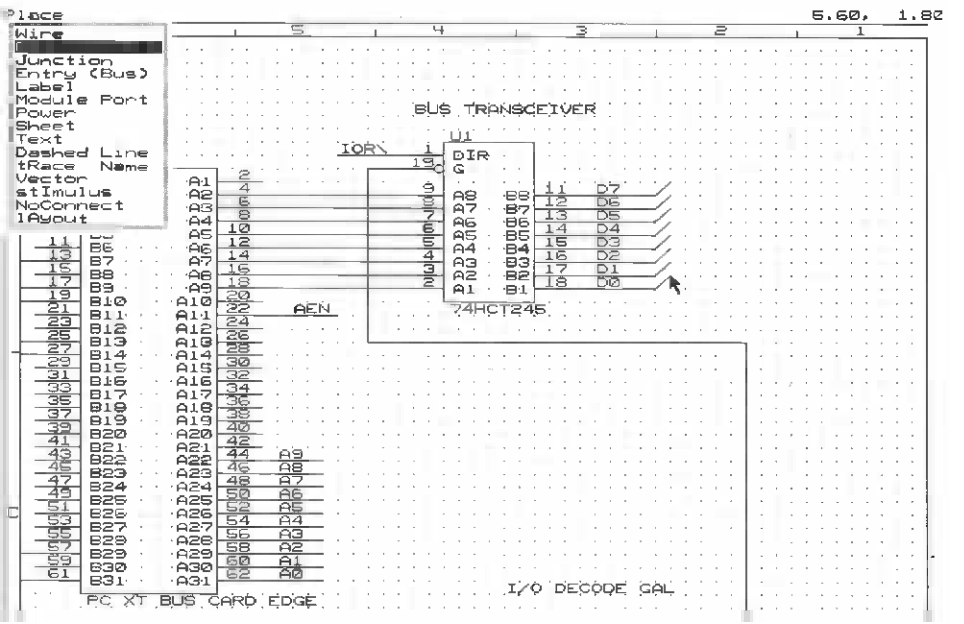


Figure 5-17 Selecting the Place Bus Command

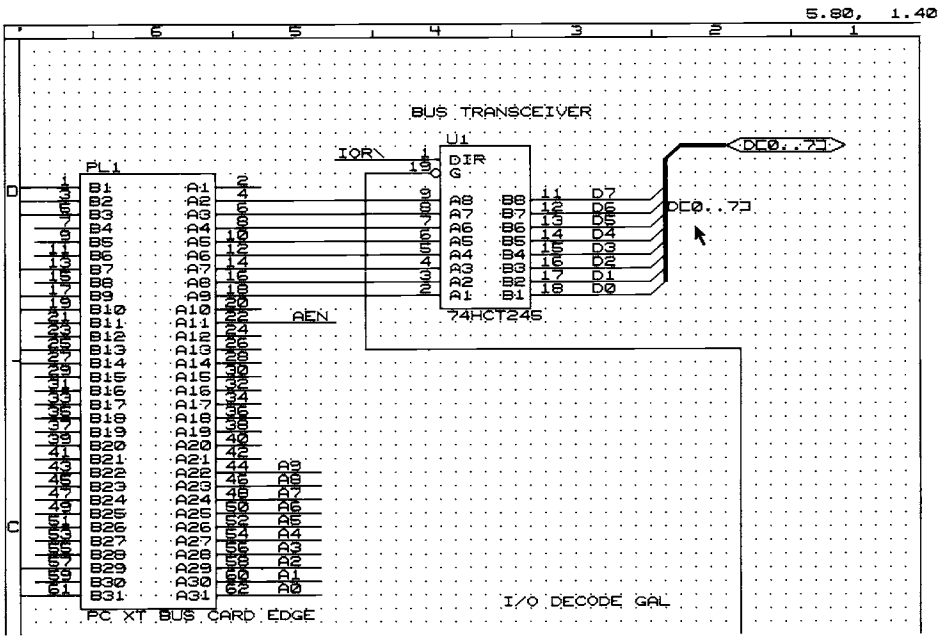


Figure 5-18 Completed Data Bus

Wrapping up the First Session

You have now finished most of the "manual labor" related to capturing the schematic for this tutorial. Exit Draft and run Print Schematic to generate hard copy. Carefully check your work and make any required corrections At this point all three sheets should appear as in Figures 5-1 through 5-3 except for the reference designators, which will be automatically assigned in the next session.

Before proceeding, backup all the files in the TUTOR3 design directory onto a floppy disk.

Second Session

The second session primarily involves new postprocessing tools. The first step is to run Cleanup Schematic, which you are familiar with from the last chapter. Afterwards, examine the #ESP_OUT.TXT file for error messages. Correct any errors and run Cleanup Schematic again if required. Then backup the design to a floppy disk. Good practice dictates backing up all design files whenever any major step, such as a postprocessing routine, has been completed.

Automatic Annotation of Reference Designators

In the first two tutorial exercises, all reference designators were manually edited and assigned the same suffix as the tutorial model. In the real world, most schematic capture tasks involve new designs. Exactly copying an existing schematic is not a consideration. OrCAD provides a convenient postprocessing routine for automatically numbering reference designator suffixes. This operation is referred to as "annotating the schematic" and can save a considerable amount of manual editing time. It also eliminates the possibility of errors, such as skipped or duplicate reference designators.

OrCAD's Annotate Schematic routine is very flexible and has some additional useful features. One extremely useful feature is the capability to perform an incremental annotation. In this case, only reference designators that still have an unassigned suffix (such as R?) are annotated. The routine automatically starts numbering from the last assigned reference designator. The need for incremental annotation often arises when an engineering change is made to an existing product and parts are added to the PCB. Annotate Schematic can also report the last used reference designators, which is useful for checking against the bill of materials or other documentation.

Running Annotate Schematic

At the main SDT screen, click on Annotate Schematic and then click on Local Configuration as shown in Figure 5-19. This brings up the configuration screen for Annotate Schematic. Verify that your local configuration matches that shown in Figure 5-20 and make any required changes.

The local configuration screen has elements similar to other local configuration screens that you have already encountered, such as the File Options section.

Additional processing options include:

Quiet mode. Same function as in other OrCAD routines. Suppresses display of text messages. Leave deactivated, so that status messages will be displayed.

Do not change the sheet number. Prevents Annotate Schematic from changing sheet numbers if they have already been assigned. Generally the sheet numbers follow in some logical relationship to the hierarchical structure and changes would be undesirable.

Unannotate schematic. Removes existing reference designator suffix assignments. Limited usefulness. If things get really messy, unconditional annotation is the quickest way to start over.

Report the last assigned reference designator. This option is extremely useful if engineering changes are being made or just to get an idea of the total parts count.

Reset reference numbers to begin with 1 on each sheet. Limited usefulness. Using this option will cause serious problems with the bill of materials listing and the netlist for PCB design.

Annotation options. Two buttons select between unconditional or incremental annotation. For a new design, where no reference designators have been preassigned, select unconditional annotation. If an engineering change is being made or parts such as switches, connectors, jumpers, or trimpots have preassigned reference designators, select incremental annotation.

Ignore warnings. Same function as in other OrCAD routines. Generally left deactivated, so that the program will terminate rather than generating incorrect results.

After you have verified the configuration options, click OK to exit. Then double click on Annotate Schematic to launch the tool. Status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appeared during execution, there is usually no reason to examine this file.

The last used reference designator information is written to the file TUTOR3.END in the form of ASCII text. Exit to DOS and then use the LIST.COM utility to examine TUTOR3.END. Figure 5-21 shows a sample listing. Note the added bonus of an unused parts in packages report. This is very useful for finding unused gates in "glue logic" ICs, which must have inputs grounded.

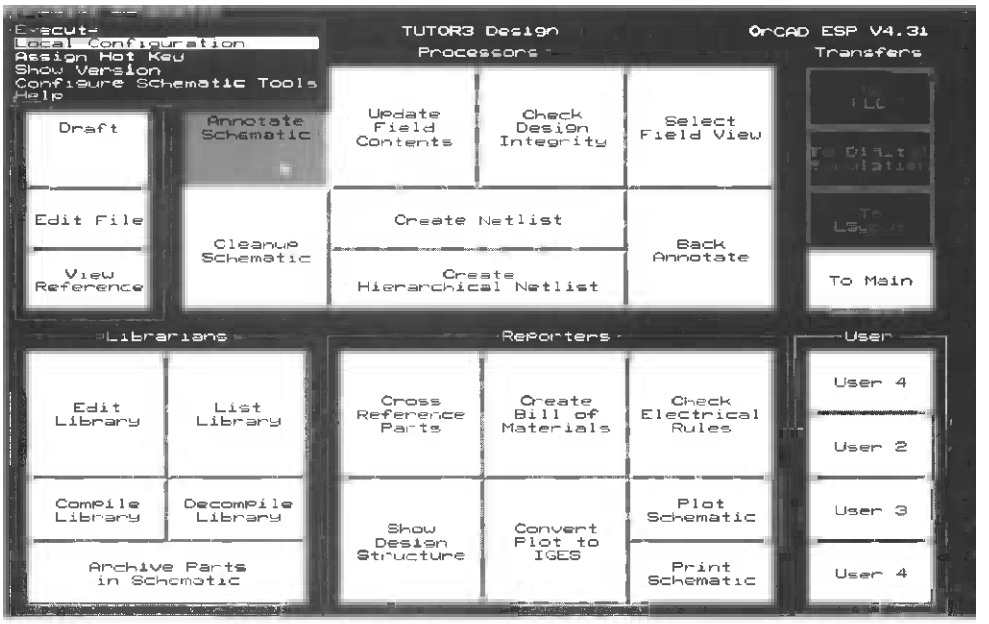


Figure 5-19 Selecting Annotate Schematic Configuration

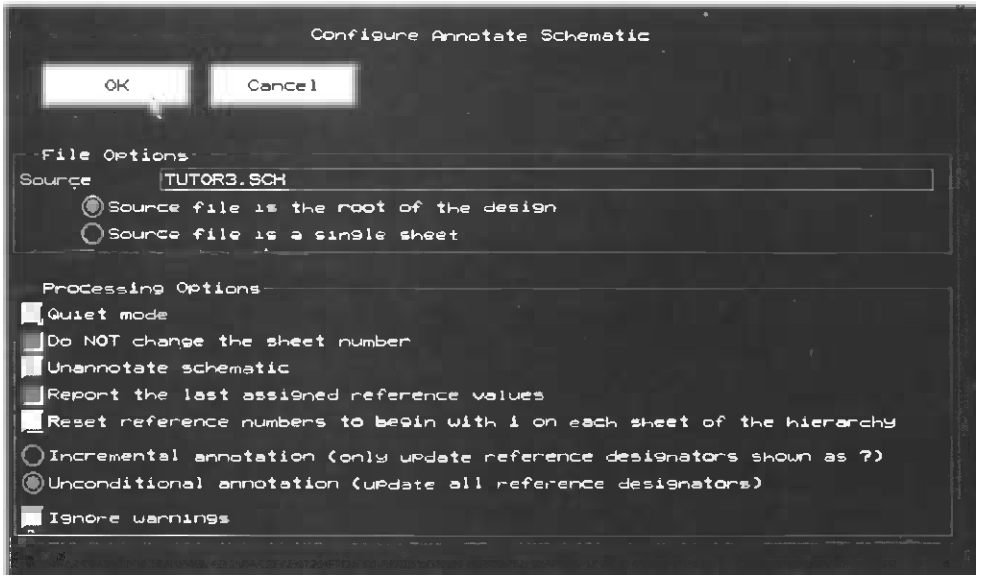


Figure 5-20 Annotate Schematic Configuration Options

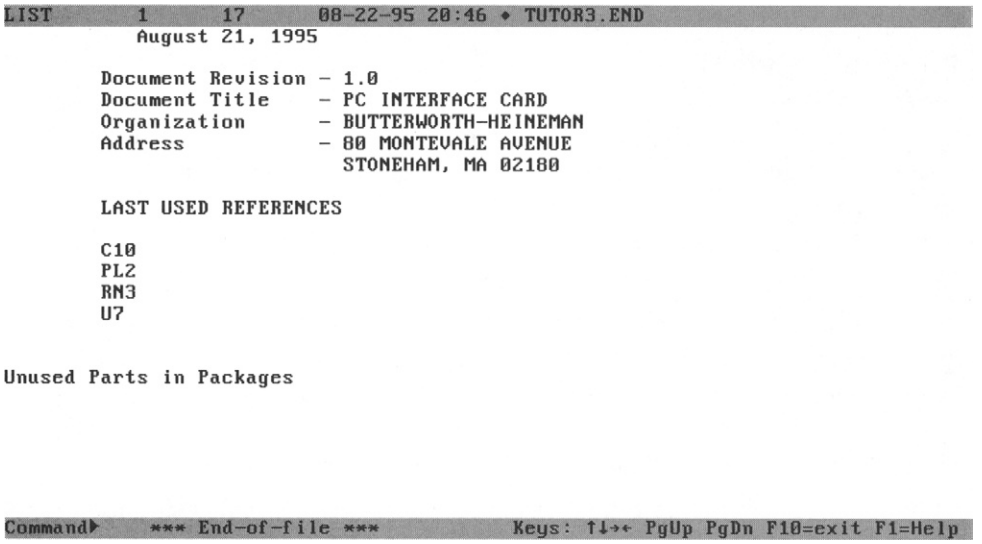


Figure 5-21 Last Used Reference Designator Report

After running Annotate Schematic and examining the last used reference designator report shown above, go back into Draft and examine the schematic. Your reference designator assignments will probably not match those shown in Figures 5-2 and 5-3. Annotate Schematic assigns reference designators roughly in the chronological order that the parts were placed as the design was being created. This is not a serious limitation in digital designs with limited numbers of VLSI parts, but can result in difficulties in dense analog circuits. If the results are too messy, use the unannotate option and then manually edit the design.

Creating a Bill of Materials

Use the techniques learned in chapter 4 to run the Create Bill of Materials postprocessing routine. Then use a text editor to edit and print out the bill of materials. Carefully examine the print out. At this point, you can catch many simple errors such as wrong part descriptions or reference designator assignments. If you find any errors, go back and correct the design. Remember to always back up all design files onto a floppy disk before making major changes or going on to the next postprocessing step.

Check Electrical Rules Overview

The Check Electrical Rules postprocessing routine checks the entire design for possible violations of basic electrical connectivity rules. OrCAD uses a decision matrix to analyze all possible combinations of connections. The decision matrix is shown in Figure 2-11 in chapter 2. Empty blocks on the decision matrix represent valid connections. Blocks labeled E or W represent errors and warnings. OrCAD defines errors as situations "that must be fixed" and warnings as "situations that may or may not be right."

In most cases, errors do require correction or at least very careful analysis. Two IC outputs connected together would be flagged as an error and in almost all cases would represent a serious design flaw. An IC output connected to a bidirectional module port would also be flagged as an error, yet there are special circumstances where this would be a valid connection.

Many warnings are in fact valid connections. However, warnings still need to be looked at and given some consideration. OrCAD lumps connections into broad categories and no analysis is done on the basis of signals. Almost all designs will have some warning situations. A common example is power supplies tied together. In the tutorial exercise, +5V and VCC are deliberately tied together and will be flagged as a warning. The electrical rules decision matrix can be modified via the Configure Schematic Design Tools screen. One might be tempted to blank out the warning caused by interconnected power supplies. But what if +5V and +12V are accidentally tied together? Experience shows that the default decision matrix is well thought out and best left alone. An occasional nuisance error or warning is preferable to letting a real mistake slip by.

Running Check Electrical Rules

Before configuring and running the routine, let's deliberately create an error in the design for the routine to catch. Select the tutorial design, launch Draft, and enter sheet two. Then add a wire connecting the two unused 74HCT14 inverter outputs as shown in Figure 5-22. Update the file so that the change is saved. Exit Draft.

At the main SDT screen, click on Check Electrical Rules and then click on Local Configuration as shown in Figure 5-23. This brings up the configuration screen for Check Electrical Rules. Verify that your local configuration matches that shown in Figure 5-24 and make any required changes.

The local configuration screen has the familiar File Option section, similar to most other OrCAD postprocessing routines.

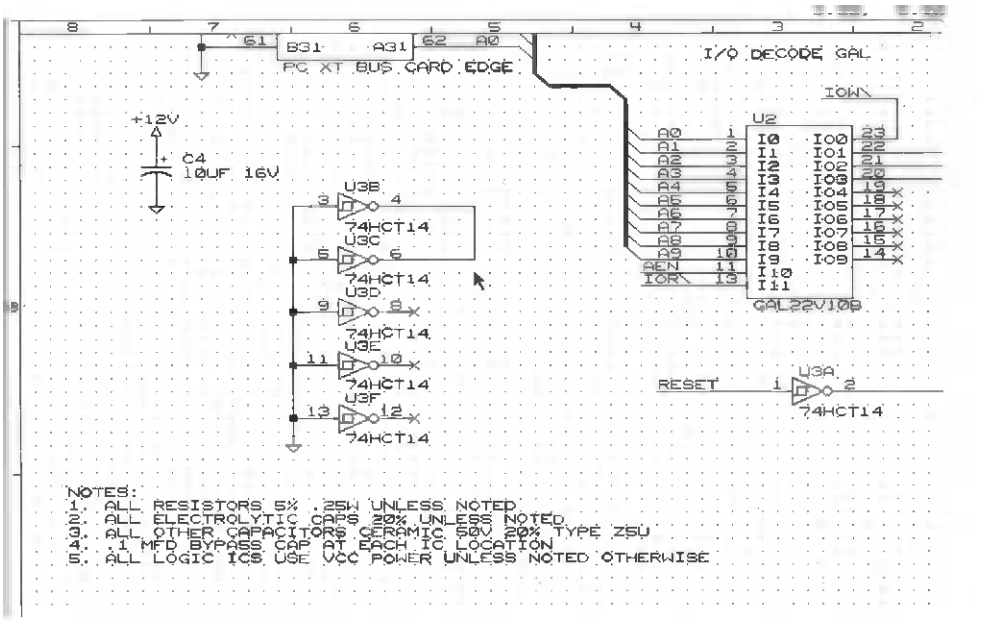


Figure 5-22 Creating a Deliberate Error at U3

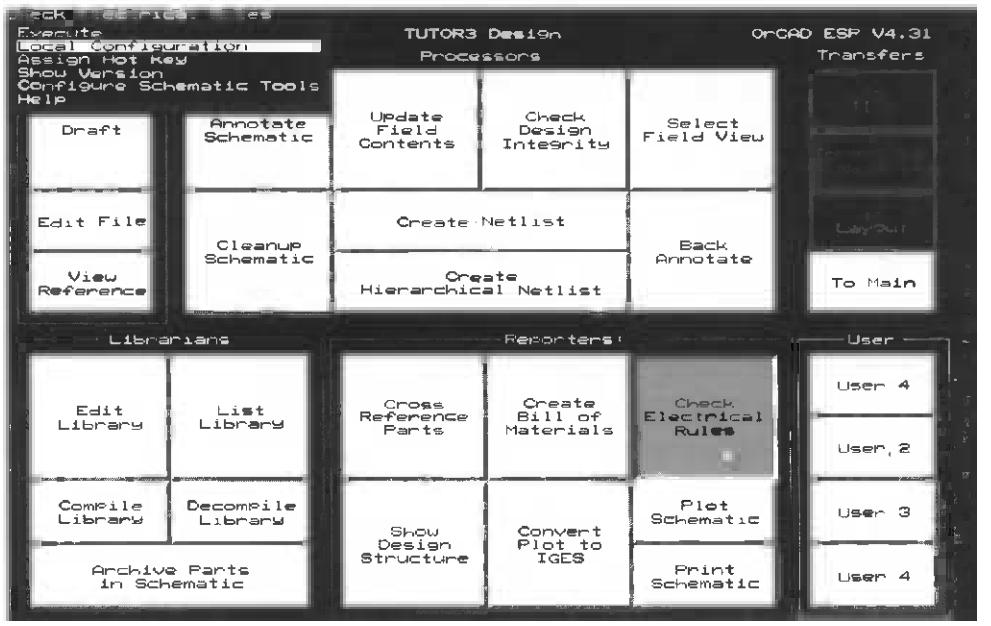


Figure 5-23 Selecting Check Electrical Rules Configuration

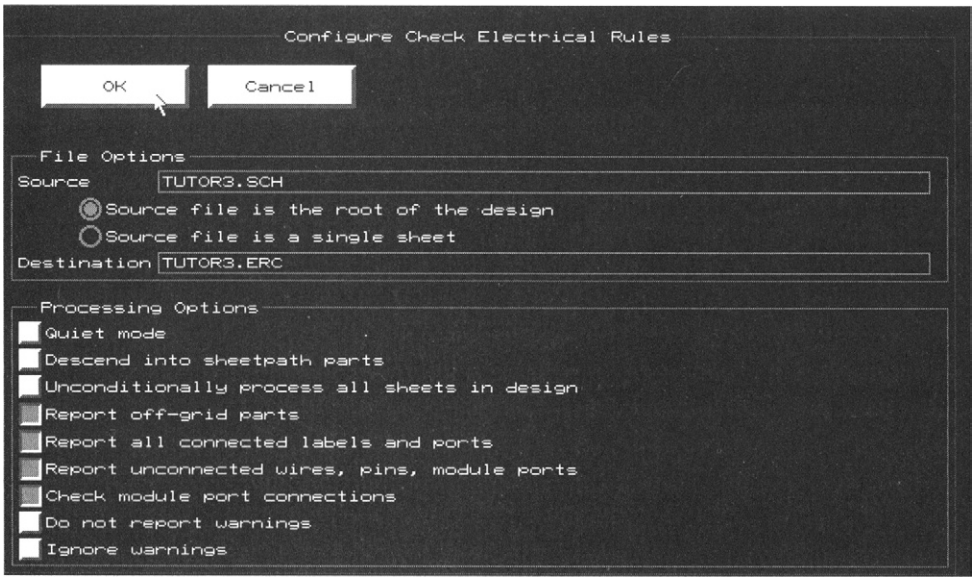


Figure 5-24 Check Electrical Rules Configuration Options

Additional processing options for Check Electrical Rules include:

Quiet mode. Suppresses display of text messages. Leave deactivated, so that status messages will be displayed.

Descend into sheetpath parts. Sheetpath parts are entire circuit blocks defined as parts. If sheetpath parts are used and are to be checked, this option must be highlighted.

Unconditionally process all sheets in design. Check Electrical Rules normally skips any sheets that have not been edited since a previous check. This saves time. In most cases, this option can safely be left deactivated.

Report off grid parts. Very important, since off grid parts are not likely to be properly connected.

Report all connected labels and ports. This report is very useful for catching subtle errors, such as incorrectly named, missing, or redundant signals.

Report all unconnected wires, pins, module ports. Good practice requires a no connect object to be placed on all unconnected pins and objects. This option catches any missed objects. Also useful for locating unused inputs that should be tied to ground.

Check module port connections. Verifies that module ports correspond to sheet nets. Using different names or leaving off a sheet net or module port is a very common mistake that this option will catch.

Do not report warnings. Not suggested, as some warnings may represent serious problems.

Ignore warnings. The terminology is misleading. In this case the "warnings" are program execution related, not electrical connection warnings. Generally left deactivated.

After you have verified the configuration options, click OK to exit. Then double click on Check Electrical Rules to launch the tool. Status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appeared during execution, there is usually no reason to examine this file.

Check Electrical Rules flags any errors and warnings by placing a donut shaped error object at the affected location on the schematic. In addition, information generated for selected reporting options (such as connected labels and ports) is written to the error report file TUTOR3.ERC in the form of ASCII text.

When working within Draft, you can use the Inquire command to display the messages associated with each flagged error or warning. After running Check Electrical Rules on the tutorial design, go back into Draft, enter sheet two, bring up the main menu, and click on the Inquire command as shown in Figure 5-25. hen click on the error object as show in Figure 5-26. The associated message is displayed at the top of the screen. Note that a second error object appears on the bottom 74HCT14 gate. This is the warning about +5V and VCC supplies being tied together. Depending on what order parts were placed on your schematic, this error object could appear at a different location.

Correct the deliberate error by deleting the wire between the two outputs of the 74HCT14. The next step is to remove the error objects. Error objects can be deleted using the Delete command, just like any other object. The disadvantage of this approach is that a complex schematic may have many error objects related to trivial warnings that you are going to ignore. Finding and manually deleting all these error objects is tedious and time consuming. A much easier approach is to run Cleanup Schematic with the Remove error objects from schematic option enabled. This will automatically remove all error objects. Go ahead and run Cleanup Schematic to remove the error objects.

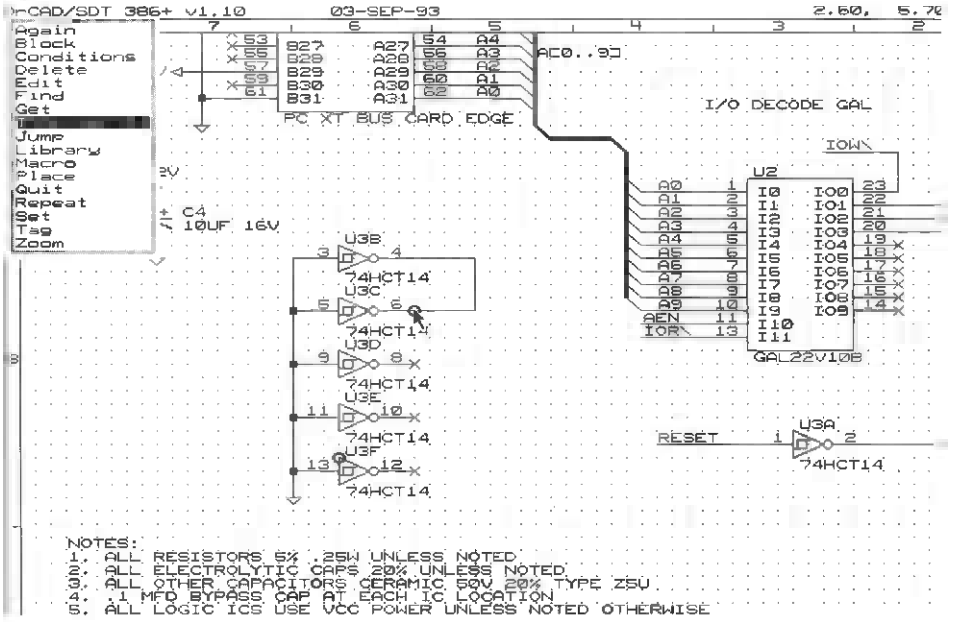


Figure 5-25 Using the Inquire Command on an Error Object

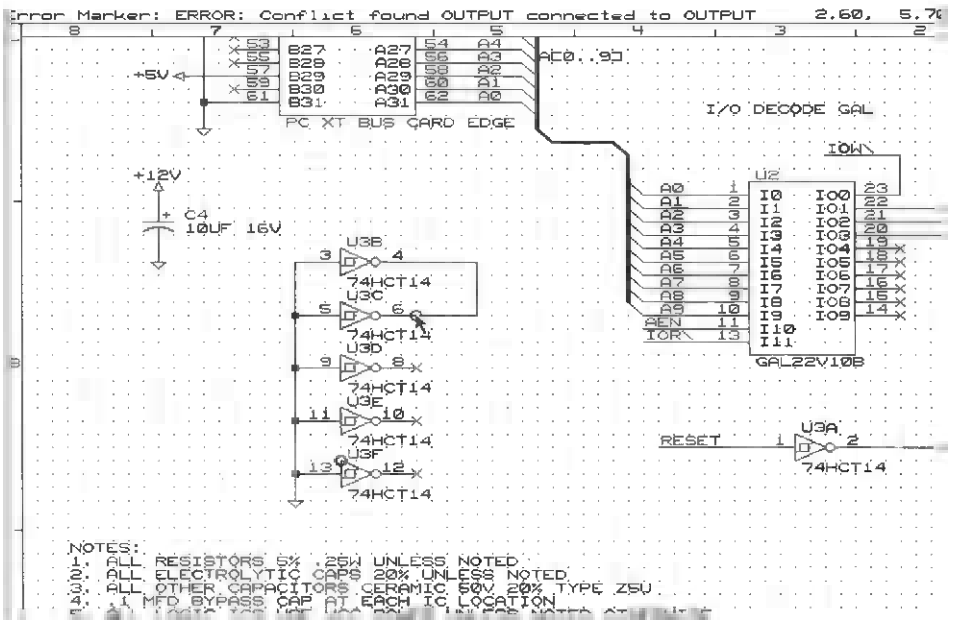


Figure 5-26 Examining the Error Message

Listed below are some of the more common error messages and suggested solutions. Note that the messages finish with one or more signal or part/pin names and in some cases a X,Y location.

WARNING: Unconnected module port ..... at X=xxx Y=yyy. Besides the obvious case of an unconnected module port, this message appears if a module port is connected to a bus and the bus is incorrectly named. Both the bus and module port must be named in the form NAME[A..B], where NAME is alphanumeric and A and B are numbers. Note that A must be less than B. Negative numbers are not allowed.

WARNING: Power supplies are connected..... This warning appears whenever two power supplies (planes) are connected together. Ground connected to CMOS VSS power pins is also considered as power supplies connected together. In most cases, this warning is the result of a deliberate action and can be ignored.

WARNING: Input has no driving source..... Appears if no wires or other electrical objects are connected to an input pin of a part. If the screen display shows that another object is connected to the input pin, the object overlaps the pin and OrCAD cannot recognize the connection. The overlapping condition might also be at a remote point such as two wire segments located some distance from the input. Leaving inputs unconnected is not good practice. Unused CMOS inputs should always be grounded. Unused TTL inputs are usually tied to +5V.

ERROR: Module port on bus does not have proper format..... This message appears if a module port is connected to a bus and the module port is incorrectly named. Both the bus and module port must be named in the form NAME[A..B], where NAME is alphanumeric and A and B are numbers. Note that A must be less than B. Negative numbers are not allowed.

ERROR: Bus label does not have proper format..... The bus label must be named in the form NAME[A..B], where NAME is alphanumeric and A and B are numbers. Note that A must be less than B. Negative numbers are not allowed.

ERROR: Sheet net on a bus does not have proper format..... This message appears if a bus is connected to a sheet net and the names do not match or they have the wrong format. Both the bus and sheet net must have the same name an the name must be of the form NAME[A..B], where NAME is alphanumeric and A and B are numbers. Note that A must be less than B. Negative numbers are not allowed.

The error messages listed above appear on the screen when using the Inquire command on error objects. The same messages along with other information specified by selected options appear in the error report file.

Examine the error report file using the LIST.COM utility. A section of the file TUTOR3.ERC, with information related to the second sheet, is shown below in Figure 5-27.

```
LIST      94                08-23-95 19:39 • C:\ORCAD\TUTOR3\TUTOR3.ERC
File: PCXT.SCH

UNCONNECTED REPORT


PCXT.SCH

Electrical Rules Check Report
PC INTERFACE CARD, BUS DECODE  Revised:    August 23, 1995
1002          Revision: 1.0
BUTTERWORTH-HEINEMAN
80 MONTEVALE AVENUE
STONEHAM, MA 02180


ERROR: CONFLICT Found OUTPUT connected to OUTPUT
 X =     2.60,Y =     5.70  U3C,O
WARNING: POWER Supplies are CONNECTED +5V <-> VCC

Checking Sheet Nets --- "PCXT.SCH"

Command▶                                  Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 5-27 TUTOR3.ERC Error Report

Modifying the Check Electrical Rules Decision Matrix

The decision matrix used for checking electrical connections shown in Figure 2-11 in chapter 2 can be modified. Modifications may be required for special applications when the default connectivity rules cause too many nuisance errors and warnings. From the main SDT menu, click on Draft and then click on Configure Schematic Design Tools. Scroll down to the Check Electrical Design Rules Matrix section. Various types of connections are shown as intersections of rows and columns. Each intersection is either empty (a valid connection) or contains a "W" for warning or 'E" for error. You can toggle between these three settings by clicking the left mouse button on the intersection. You can return the entire matrix to the default connectivity rules by clicking on the Set to Defaults box in the top left corner.

Overview of Hard Copy and Data Transfer from OrCAD

OrCAD SDT provides two postprocessing routines for generating hard copy. Print Schematic was introduced in the first tutorial. The other routine is Plot Schematic. Plot Schematic can directly drive a pen plotter or send plot data to a file. Up until a few years ago, the distinction between printers and plotters was clear-cut. Printers were typically small format, 8.5 x 11 inch A-size raster devices. Plotters were large format vector devices, some capable of doing up to 36 x 48 inch E-size drawings. Raster devices generate an image from a pattern of small dots, typically 300 or 600 DPI for laser printers. Vector devices generate an image by drawing a series of straight lines, arcs, circles, and text characters.

Today this distinction is less clear. Laser printers, and even color inkjet printers, are now readily available in A (8.5 x 11 inch), B (11 x 17 inch), and even C (17 x 21 inch) sizes. Pen plotters have almost entirely been replaced by large format inkjet plotters, such as the HP (Hewlett-Packard) DesignJet series. These inkjet plotters are raster devices. The inkjet plotters evolved very rapidly. To maintain compatibility with existing software packages, all of the new inkjet plotters can emulate older pen plotters. Most will accept data in HPGL (Hewlett-Packard Graphics Language) and directly emulate HP7580/7586 pen plotters. Because HPGL format has been around for over a decade, the software drivers are usually very stable and bug-free.

OrCAD uses the term plotter to refer to vector devices and the term printer to refer to raster devices. The Plot Schematic routine only generates vector data.

Today, only a few companies still use pen plotters to plot schematics. Most schematics are done as multiple sheet hierarchical designs on A-size paper using a laser printer for hard copy. Some situations do exist where larger format drawings are required. Hard copy for these large drawings is typically generated using a large format laser or inkjet plotter. Because these newer devices are not directly supported by OrCAD, the solution is to use the HP emulation capability and send out HPGL data.

Often the need arises to transfer schematics captured with OrCAD into other software packages. One common scenario involves annotating schematics with signal waveforms or details of specialized electromechanical devices using CAD software such as AutoCAD. Another scenario is pasting schematics into technical publications, such as this book.

The easiest method of transferring data from OrCAD into other CAD systems is via DXF format. DXF format originated with AutoCAD and is now widely supported throughout the EDA and CAD industry. OrCAD SDT can be configured for a DXF output driver as a plotting option in Plot Schematic. The

DXF data is routed to a file which can then be loaded into AutoCAD. The DXF driver even does a great job of transferring text.

Before DXF format became widely accepted, the CAD industry adopted IGES (Initial Graphics Exchange Standard) format for data transfer. OrCAD has a half hearted IGES capability via the Convert Plot To IGES routine. In stark contrast to the ease of generating DXF data, to get IGES data you must first configure OrCAD for a "generic" vector plotter (this is the term OrCAD uses), then run Plot Schematic, and finally run Convert Plot To IGES.

Pasting large circuit blocks or entire schematics into technical publications is best accomplished by transferring HPGL data. Configure OrCAD for one of the common HPGL plotters and direct the output to a file. Microsoft Word, Corel Draw/Ventura, and Aldus Pagemaker all accept HPGL format. This book was written using Microsoft Word and Figures 5-1 through 5-3 were imported as HPGL files.

Using Plot Schematic

Plot Schematic is used to plot schematics and output vector data for transfer to other systems. The routine can directly drive a pen plotter attached to one of the serial or parallel ports. Plot Schematic also provides an option for driving a printer. However, contrary to the published OrCAD manual, printing via Plot Schematic seems to result in lower quality hard copy than using the Print Schematic routine. This appears to be the case for Hewlett-Packard laser printers and may be the result of a compatibility problem or a bug within the OrCAD driver.

For the purpose of this tutorial, Plot Schematic will be used to generate an HPGL data file, such as would be required for pasting a schematic into a technical publication. This appears to be the most useful way to introduce the Plot Schematic routine, since few readers are likely to have a pen plotter connected on-line to their system. For additional information on using Plot Schematic with an on-line plotter, refer to appendix A.

At the main SDT screen, click on Plot Schematic and then click on Local Configuration as shown in Figure 5-28. This brings up the configuration screen. Verify that your local configuration matches that shown in Figure 5-29 and make any required changes.

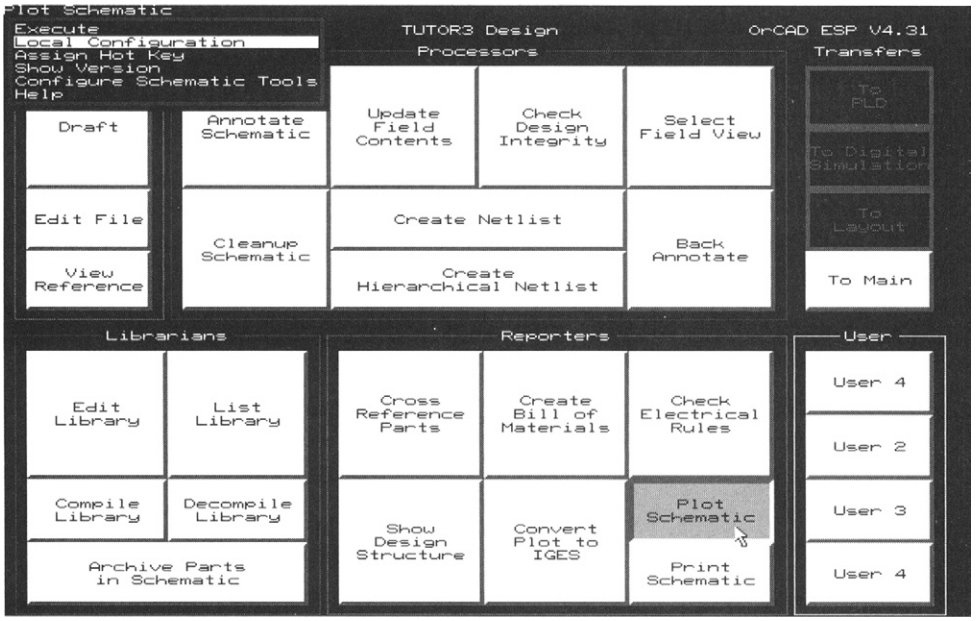


Figure 5-28 Selecting Plot Schematic Configuration

The local configuration screen has the familiar File Options Source section and a number of other options as explained below:

Send output to plotter. Sends data to the plotter selected in the OrCAD SDT configuration.

Send output to printer. Sends data to the printer selected in the OrCAD SDT configuration. When OrCAD is configured for a HP LaserJet compatible printer driver, the hard copy generated by this routine appears rough. Use Print Schematic instead. The rough print may be a bug.

Send output to a file. Data is sent to a file. Use this option to generate graphics data for transfer and import into other systems.

Create a print file. This option becomes available when the file output option is selected. Sends raster data to the file specified in the filename box. Data format is for the configured printer. Contrary to the OrCAD manual, data for the entire design is sent to a single file with embedded page breaks.

Create a plot file. This option becomes available when the file output option is selected. Generates a separate vector data file for each sheet, using the filename extension specified in the entry box. Data format is for the configured printer.

Quiet mode. Suppresses display of status messages, same as in other OrCAD postprocessing routines.

Descend into sheetpath parts. Plots out any sheetpath parts as separate sheets with full detail. Otherwise sheetpath parts appear as a block.

Plot grid references around the worksheet border. Plots out the border area containing alphanumeric grid references. Including a border with grid references is good practice.

Ignore fill commands. Suppresses stroking of filled areas, such as inside diode symbols. This option is not normally set.

Produce 1:1 scale plot. The plot is the same size as the worksheet setting.

Automatically scale and set X,Y offsets for specified sheet size. Allows selecting a different plot size. Does not automatically calculate scale factors and offsets. Uses the values set on the template table (OrCAD SDT configuration). Large format HP pen plotters, such as the HP7580/7586 series, use an origin at the center of the sheet. Even with this option selected, OrCAD does not set correct origin locations on these plotter models. You must manually enter special offsets in the template table. These HP plotters will typically require negative offsets equal to one-half the selected plot size. Refer to appendix A for more details.

Manually set scale factor and X,Y offsets. Additional entry boxes are provided for the scale factor and the two offset values. The allowed range for scale factors is .10 to 10.000. Offsets are entered in inches with an allowed range of -30.000 to +30.000. Negative values shift the plot origin to the left. HP plotters with origins at the center of the sheet will typically require negative offsets equal to one-half the selected plot size. Refer to appendix A for more details. Some plotter drivers, including the HP driver, have a bug that causes a divide-by-zero error if no offset is specified.

Plotter uses single sheet paper. Some older plotters were offered with and without roll feed option. Use this option for older plotters without roll feed and for newer inkjet models using HPGL emulation, where paper feed is automatic once an end of plot command is received.

Plotter uses roll feed paper. Use this option for older roll feed plotters.

Printer has narrow paper. Normal default for printer hard copy.

Printer has wide paper. Use this option only for mechanical dot matrix printers using wide paper. Not applicable to laser printers.

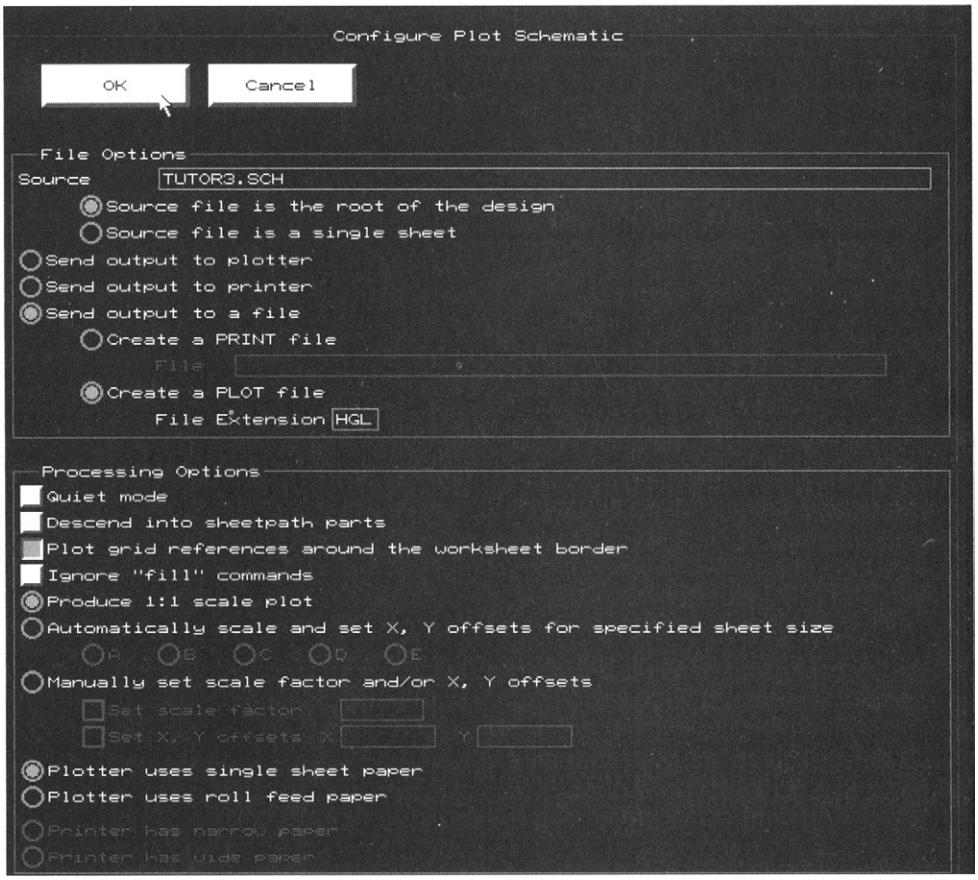


Figure 5-29 Plot Schematic Configuration Options

After you have verified the local configuration options, click OK to exit. Then double click on Plot Schematic to launch the tool. Status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appears during execution, there is usually no reason to examine this file.

Exit to DOS and use the DIR (directory) command to examine the design directory. If you used the suggested plot file extension, HGL, three plot files with the names TUTOR3.HGL, PCXT.HGL, and DIGITAL.HGL should appear in the directory. HPGL format files are ASCII data and can be examined with the LIST.COM utility. Looking at one of the files will give you a feeling for the appearance of HPGL data. This knowledge may be helpful in diagnosing problems importing HPGL data into other applications. If you use a word processor or desktop publishing program that accepts HPGL data, you can try to

import one of the plot files. Microsoft Word accepts HPGL data files and uses the default extension .HGL.

Plotting to Network and RS-232 Devices

Other potential uses for plotting data to a file occur in network environments or where the plotter is attached to a different PC. Networks frequently use virtual device names for shared resources such as a plotter. It is not uncommon to find a virtual device name such as LPT5. OrCAD does not directly support plotting or printing to such devices. However, DOS can be used to copy the plot file to the virtual device. For example, the DOS command

COPY PCXT.HGL LPT5: /B <ENTER>

would copy the PCXT.HGL file to LPT5. The /B option forces a binary transfer. This assures that all data is transferred. Without this option certain control characters can be lost.

The same form of the DOS copy command can be used when sending a file to a plotter or printer attached to a remote PC. First use a floppy disk to transfer the file to the remote PC. Then copy the file to the attached device. If the device is attached to an RS-232 serial port such as COM1 or COM2, you must use the DOS MODE command to configure the port before copying any files. Refer to your DOS manual for details. Note that using the COPY command on a serial port only works with hardware handshaking. COPY will not work with the simpler three-wire RS-232 hookup used by some devices. Three-wire RS-232 requires a communications program that supports software handshaking.

Wrapping up the Second Session

Learning the in's and out's of Plot Schematic completes the second session. On a final note, bear in mind that setting up all the parameters for successfully plotting on-line to large format plotters can be a formidable challenge. A certain amount of "cut and try" is usually required. The same applies to transferring graphics data to other applications.

Before proceeding, backup all the files in TUTOR3 design directory onto a floppy disk. You can then delete all the plot files with .HGL extensions. Always do the backup first, just in case you delete the wrong files.

Third Session

The third and final session covers netlist generation. This is one of the major benefits of schematic capture using a program such as OrCAD. The primary use for a netlist is as input to a PCB design system. Netlists can also be used as input to circuit analysis programs such as SPICE or as input to PLD (Programmable Logic Design) programs. The focus of this tutorial is on generating a netlist for PCB design.

Netlist Overview

Netlist is the term used for a file that contains data representing the parts and electrical interconnections found in a design. The electrical interconnections are referred to as nets. A net consists of part pins (referred to as nodes) that are connected together. Power and ground are also considered to be nets. Parts are described in terms of the reference designator and part description.

No industry standard exists for netlist data formats. Each software vendor has evolved their own proprietary format. At last count, OrCAD supports over thirty different netlist formats. Many of these formats are obsolete with the original vendors having long since gone out of business or having reorganized. Most PCB design software vendors support several of the more common netlist formats. Of these, one of the most widely-supported is FutureNet format. FutureNet was the first schematic capture software package that would run on a PC. The FutureNet product did not survive, but the netlist format lives on. You will learn to generate a FutureNet netlist in this session.

Create Netlist is OrCAD's very powerful and complex postprocessing routine for generating netlists. Create Netlist consists of three subroutines. The last of these subroutines runs an executable netlist formatter that contains the program used to generate the netlist. A unique formatter exists for each supported netlist format.

Transfer of data between software applications is usually a complex matter, and netlist files are no exception. Many of the netlist formats supported by OrCAD impose severe restrictions on part reference designator and label/module port names. These restrictions may include limitations on the length of names and what characters are considered valid. Create Netlist does a good job of checking for invalid names and attempts to make corrections where possible.

The user must expect to spend a significant amount of time manually editing the netlist file generated by OrCAD (or any other schematic software package) before transferring the data. Time spent editing the netlist file is often a good tradeoff for even more time that would otherwise have to be spent setting things up in OrCAD or in the PCB design software. You must make sure that that

library parts defined in OrCAD will match those defined in the PCB design software. Both pin names and part descriptions must match.

Pin Names and Pin Numbers

The subject of pin names and pin numbers requires additional discussion. Pin names and numbers are pin definition fields that are discussed in detail in the next tutorial (chapter 6), which covers the parts library editor. All OrCAD parts have pin names, but some parts are defined without pin numbers. ICs are examples of parts that have both pin names and pin numbers. Most discrete parts including capacitors, inductors, resistors, diodes and transistors are examples of parts that have no pin numbers. These parts have pin names, but the pin names are invisible.

Users sometimes become confused by the use of numeric pin names in OrCAD library parts. Capacitor, inductor, and resistor pins are named 1 and 2. Polarized electrolytic capacitors have pin number 1 assigned to the positive pin. Discrete semiconductor devices typically have alphanumeric pin names. Diode pins are named CATHODE and ANODE. Transistor pins are named EMITTER, BASE, and COLLECTOR. In all of these examples of discrete parts, the information is in the pin name field. The pin number field is not used.

Netlist formats may use either the pin name or pin number field to identify pins. If the netlist format uses pin numbers and the pin number field for a particular part is undefined, OrCAD will substitute the pin name. This is where the problems start. Not all netlist formats or PCB design software supports alphanumeric pin numbers, such as would occur with diodes and transistors.

Parts such as small ICs have pin numbers defined by the manufacturer or industry standards. Pin numbers for such standardized parts are usually not an issue. Large PGA (pin grid array) ICs can pose problems. These parts have alphanumeric pin numbers such as A11, where the alpha character corresponds to the vertical axis of the grid (A to Z from top to bottom) and the numeric character(s) corresponds to the horizontal axis starting from 1 at the left. If the PCB design software does not allow alphanumeric pin names, one possible work around is to just arbitrarily number the pins in rows and columns.

Discrete semiconductor devices are especially problematic due to the variety of possible pin arrangements and case styles. For example, let's consider transistors. While a single standard symbol is used to represent all NPN transistors on the schematic, many different case styles are available. Case styles range from SOT-23 surface mount devices to large industrial versions with screw terminals. The pin arrangement varies with case style. A given case style can even have several different pin arrangements. For example both 2N4401 and BF224 transistors

come in TO-92 cases. The 2N4401 has pin arrangement EBC (emitter, base, and collector) while the BF224 has pin arrangement CEB from left to right, viewing flat front surface of the parts.

Let's assume the design requires a 2N4401 and a BF224. There are several alternatives for dealing with the pin arrangement situation, including:

- Create two custom OrCAD library parts, one for each NPN transistor pin arrangement. Assign numeric pin names 1,2,3 in order from left to right. Create a single TO-92 part in the PCB design software, also using pin numbers 1,2,3 from left to right. Transfer the netlist without editing.
- Create a single OrCAD library part, representing an NPN transistor with emitter as pin 1, base as pin 2, and collector as pin 3. Create two TO-92 parts in the PCB design software, using pin arrangements 1,2,3 and 3,1,2. Transfer the netlist without editing.
- Use the existing OrCAD library part for an NPN transistor with pin names EMITTER, BASE, and COLLECTOR. Create a single TO-92 part in the PCB design software, using pin numbers 1,2,3 from left to right. Edit the netlist and replace the pin names with numbers 1,2,3 or 2,3,1 as required.

Which approach is best? Since the OrCAD library editor is very quick and easy to use, the first alternative might seem to be the logical choice. In the real world you will encounter many different types of transistors. MOSFETs and some ICs also come in TO-92 packages. The OrCAD libraries already have symbols defined for these parts. Likewise, most PCB design software packages come with a parts library that includes common case styles such as TO-92.

If you use either of the first two alternatives, you will still have to double-check the pin arrangements at some point during the design process to make sure a mistake did not occur. The author's experience suggests that staying with the standard libraries and using the third alternative is the most efficient approach.

Part Descriptions

The minimum part information required for the netlist output includes reference designator and part description. Reference designators identify particular parts both on the schematic and on the PCB layout. Few problems occur transferring reference designator information via the netlist.

For netlists, the part description consists of the part value and the module value. The term module value should not be confused with module ports. The two concepts are completely unrelated. Module value is the term OrCAD uses for optional part information that can be extracted for one of the ten available part fields. Remember that each part has a reference designator field, part value field,

and eight optional user defined fields. Any one of the user defined fields can be configured to represent module value. In the context of netlists for PCB design, the module value is the description of the part shape and pad layout. For example, a design might have two 10K 5% .25 watt resistors. These two parts have the same part value, but one might be a conventional axial lead package and the other surface mount. The module value field could be used to differentiate the two package styles.

The module value field is defined in the SDT configuration under the Key Fields section (refer back to chapter 2 for details). The part field entered for Create Netlist Module Value Combine in the Key Fields section is used to extract the module value. If no entry is made, the part value will be also be used for the module value. This SDT configuration is used for the tutorial exercises.

There are pros and cons to defining specific module values for each part during the schematic capture process versus just using the part value.

Let's use common two terminal resistors as an example. A single standard symbol is used for all resistors on the schematic. The parts description typically includes the resistance value, tolerance, and wattage rating. On the PCB layout, the major consideration is the case size and pad pattern. Different wattage resistors require different part layouts. Likewise surface mount parts require a different layout than conventional axial leaded parts. A given schematic may have dozens of different .25 watt resistor values all of which use the same layout. There are two alternatives for dealing with part descriptions including:

- Generate the schematic with a complete part description including module value. The first part field is used for the module value, such as R\40LS for all .25 watt resistors. If OrCAD SDT is properly configured, the information in the first part field will automatically be transferred into the PCB design package without requiring any netlist editing.
- Generate the schematic using only the reference designator and part values fields without a separate field for module value. When the netlist is generated, the part value is also used for the module value. Prior to transfer to the PCB design package, the resulting netlist is edited. The module value for each resistor is replaced with the appropriate layout part description. All .25 watt resistors might be named something like R\40LS (resistor with .40 inch lead spacing).

At first glance, the first alternative might appear to be the most straightforward. However, several factors weight in favor of the second alternative:

- Most designs have parts such as the transistors described in the previous section. Pin naming considerations related to multiple pin arrangements may mean that some amount of netlist editing is inevitable anyway.
- The electronics engineer doing the schematic design may not have prior knowledge of all the correct module values (part layout descriptions) if PCB design is done by another department or subcontractor. This is more often the case than not and especially holds true if the design requires new parts.
- Editing the netlist may actually involve less work. Most text editors have convenient and powerful search and replace functions. If part value is also used for the module value, you could easily replace all instances of a 10K .25W resistor module value with the appropriate layout description. Using Draft to enter the PCB layout description into the module value field for every part usually involves much more work.

Unless there are strong factors in favor of the other alternative, the suggested approach is to use the part value for module value when generating the netlist and then edit the netlist.

How OrCAD Creates a Netlist

Create Netlist is the OrCAD postprocessing tool for creating "flat" netlists (also referred to as "linked" netlists). These are the types of netlists most commonly used for PCB design, which is the focus of this tutorial. "Flat" netlists list all module ports, signals, and parts in the design with unique names. Any structural information about the design hierarchy or reuse of design elements including sheetpath parts is removed. The design is resolved into one giant sheet with all the connectivity information required for a PCB.

The Create Netlist tool consists of three independent routines, each of which incrementally process design information to generate the final netlist. Each routine generates intermediate files. While Create Netlist somewhat automates this complex process, the user must still possess a detailed understanding of the individual steps in order to avoid error conditions. The process has a certain elegance, but OrCAD has introduced an unnecessary level of complexity for most users. In actual everyday use, the internal details are of little concern for the average user. You can skim through most of the explanatory material.

The internal process of creating a netlist is not unlike compiling and linking a computer program written in a high-level language. The process is shown in Figure 5-30. File names given in the figure are for the actual files occurring in this tutorial exercise. Schematic files with .SCH extension are initially compiled by INET. INET creates what OrCAD refers to as an incremental connectivity

database. The result includes one .INX file for the entire design and an individual .INX file for each sheet. The .INF files contain connectivity information for the individual sheets. The .INX file is an index of the .INF files.

INET compares the time stamp for each sheet against the time stamp of the corresponding .INF file if one already exists. The .INF file is only recompiled if the sheet has a more recent time stamp, which implies that the sheet has been changed. If Create Netlist is run several times because errors were spotted on the netlist or engineering changes were made to the design, INET only recompiles sheets with changes. Theoretically this might result in some time savings for very large designs run on slow processors. In practice the time savings is nil.

The incremental nature of INET can result in several problems. If the system clock is not set properly, INET may fail to recognize that a sheet has been changed. INET also has a serious bug that is a direct consequence of incremental processing. Let's say that several sheets in a multiple sheet design have errors reported by INET. If the errors are corrected on only one of the sheets and INET is rerun (as part of Create Netlist), it will fail to recognize and report the uncorrected errors on the remaining sheets. The remaining sheets do not have a revised time stamp and thus INET will not recompile them. INET does provide an option that forces recompilation of all sheets. However, a better approach is to erase all the intermediate files before running Create Netlist again.

The next step in the netlist process is the ILINK routine. ILINK creates three binary files with .INS, .RES , and .PIP extensions. These files are used in the final step, the flat netlist formatter IFORM. The .INS file contains parts information The .RES file contains connectivity information. The .PIP file contains any pipe commands present in the schematic. Pipe commands are special instructions placed on the schematic in the form of text strings. Pipe commands are usually required when schematics are generated for input to electrical simulation packages such as SPICE.

The final step is the IFORM routine. IFORM formats the .INS, .RES, and .PIP files into one of over thirty available netlist formats.

Create Netlist automates the individual steps in this process and makes them transparent to the user.

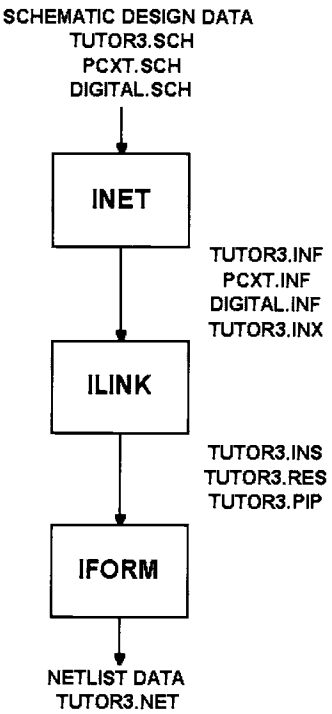


Figure 5-30 Create Netlist Process

For the advanced user, OrCAD provides a means of compiling netlist format files for use with IFORM in order to support new or custom netlist formats. The OrCAD SDT 386+ Reference Guide Supplement describes, in detail, the process of compiling netlist format files. The format files are executable files compiled using the Microsoft C version 6.0 or later compiler. OrCAD provides C source code examples and a batch file that automates the compilation process. An in-depth knowledge of C language programming is a prerequisite. Compiling netlist format files is of relevance primarily to professional system developers looking to provide links to OrCAD. The capability of compiling custom netlist format files is an extremely powerful feature, but most users will never delve into this level of detail and complexity. Compiling custom netlist format files is beyond the scope of this book.

Running Create Netlist

Each of the Create Netlist routines (INET, ILINK, and IFORM) have their own local configuration. Most of the configuration options are obscure and are not required for ordinary usage. Skim through the material, but make sure that your local configuration is set as shown in the figures before you try to run Create Netlist.

At the main SDT screen, click on Create Netlist and then click on Local Configuration as shown in Figure 5-31. This brings up the second menu level that allows configuration of INET, ILINK, and IFORM (Figure 5-32). Click on Configure INET. Verify that your INET configuration matches that shown in Figure 5-33 and make any required changes.

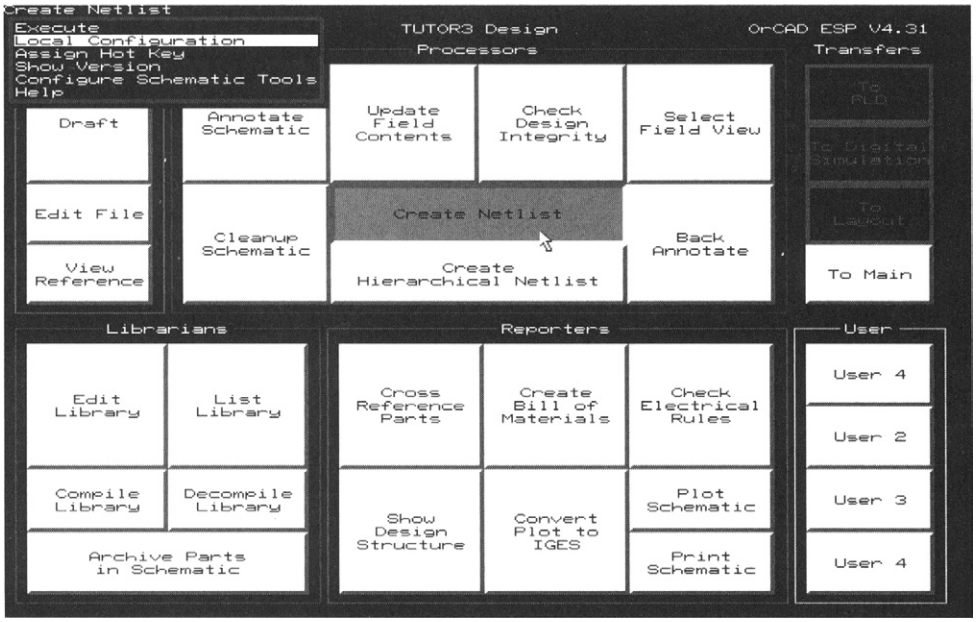


Figure 5-31 The Create Netlist Configuration Menu

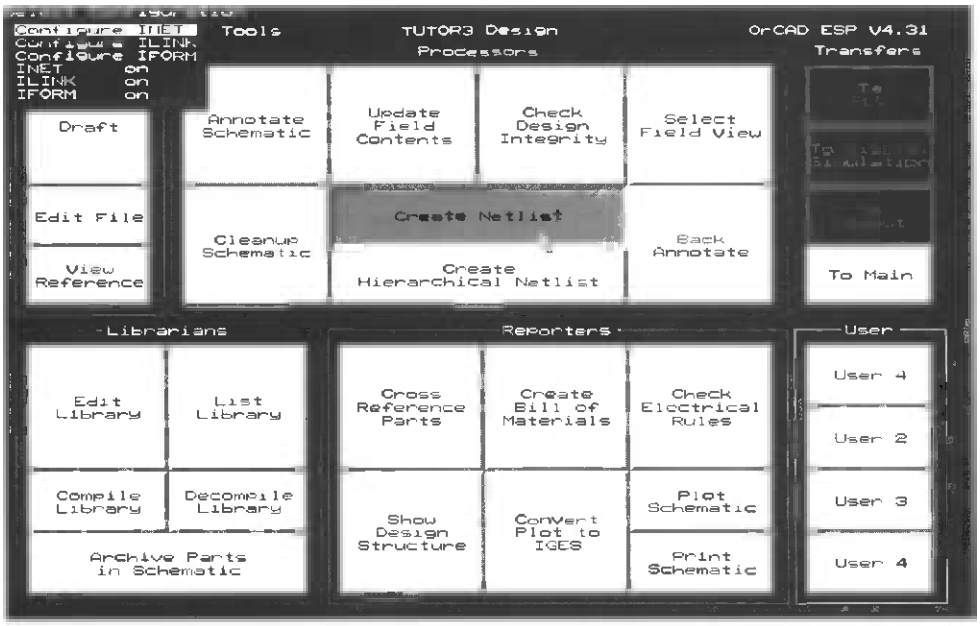


Figure 5-32 Selecting INET Configuration

Figure 5-33 INET Configuration Options

The INET configuration screen has the familiar File Options Source section and other options that are explained below:

Reports Destination. Reports which include status messages are normally sent to the screen. Allows specifying a file destination for reports.

Quiet Mode. Suppresses display of text messages. Leave deactivated so that status messages will be displayed.

Descend into sheetpath parts. If sheetpath parts are used, it is generally advisable to highlight this option. Causes sheetpath parts to be exploded.

Assign a net name to all pins. Forces INET to assign a net name to all pins including unconnected pins. Generally not required.

Report off grid parts. Causes INET to check for any off grid objects. Generally not required. Run Check Electrical Rules before Create Netlist.

Report all connected labels and ports. Generally not required. Run Check Electrical Rules before Create Netlist.

Report unconnected wires, pins, and module ports. Generally not required. Run Check Electrical Rules before Create Netlist.

Run ERC on all sheets processed. ERC refers to Check Electrical Rules. Do not run Check Electrical Rules from ILINK. Problems occur if all errors are not corrected before ILINK is run again, due to the bug described on page 192. Find and correct all errors before running Create Netlist.

Check module port connections. Generally not required. Use Check Electrical Rules for this function before generating a netlist.

Rebuild file stack. Rebuilds the .INX file. Exact function of this option is unclear. Not required if suggested steps in this tutorial exercise are followed.

Unconditionally process all sheets in the design. Forces INET to process all sheets. Helps avoid the bug described on page 192. Not required if all intermediate files are erased before running Create Netlist.

Do not create .INF file (report only). Generates only the report.

Process one sheet only. Another workaround to the bug described on page 192. However, it would require running Create Netlist twice to be effective. Not required if suggested steps in this tutorial exercise are followed.

Include unused signals connected to module ports. Limited usefulness. Unused signals connected to module ports are typically design rule

violations and do not correspond to any physical entity that would appear on an actual PCB.

Use reference designator as sheet name for sheetpath parts. Applies only to designs with sheetpath parts. Limited usefulness.

Report empty module values. Issues a warning if a part with an empty module value is encountered. The module value is extracted from the part field defined in the Create Netlist Module Value Combine key field during SDT configuration. If no field is defined, the part value field is used. If a module value field is defined, use this option to report any parts where the module value field was left empty.

Ignore warnings. This option is generally selected to prevent INET from terminating abnormally if a minor warning occurred. If not selected, any warning condition, will cause the Create Netlist process to stop after INET.

After you have verified the INET configuration options, click OK to exit. Go back to the Create Netlist configuration screen shown in Figure 5-32 and click on Configure ILINK. Verify that your ILINK configuration matches that shown in Figure 5-34 and make any required changes.

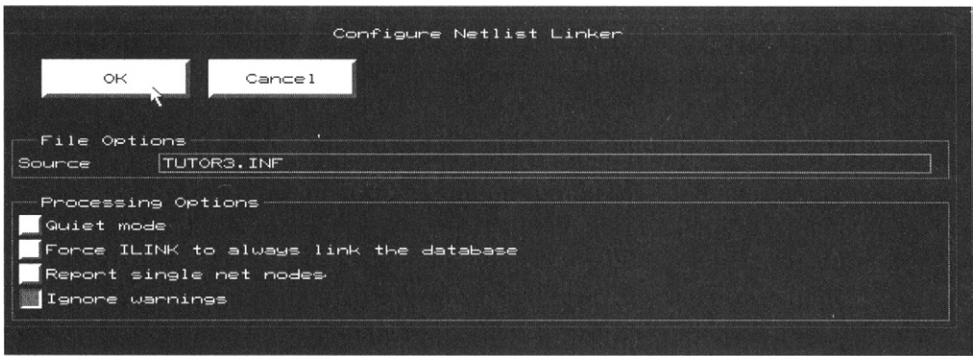


Figure 5-34 ILINK Configuration Options

The ILINK configuration screen has a File Options Source section. Note that the name of the source file is TUTOR3.INF. This is the first file output from INET. The filename will always be the name of the design with an .INF extension.

Additional ILINK options include:

Quiet Mode. Suppresses display of text messages. Leave deactivated so that status messages will be displayed.

Force ILINK to always link the database. Forces ILINK to process all files, even if none have been updated. Helps avoid the incremental update bug described on page 192. Not required if suggested steps in this tutorial exercise are followed and all intermediate files are erased before running Create Netlist.

Report single net nodes. This option helps identify design errors related to unconnected pins. Generally not required. Use Check Electrical Rules for this function before generating a netlist.

Ignore warnings. This option is generally selected to prevent ILINK from terminating abnormally if a warning occurred. If not selected, any warning condition will cause the Create Netlist process to stop after ILINK.

After you have verified the ILINK configuration options, click OK to exit. Go back to the Create Netlist configuration screen shown in Figure 5-32 and click on Configure IFORM. Verify that your IFORM configuration matches that shown in Figure 5-35 and make any required changes.

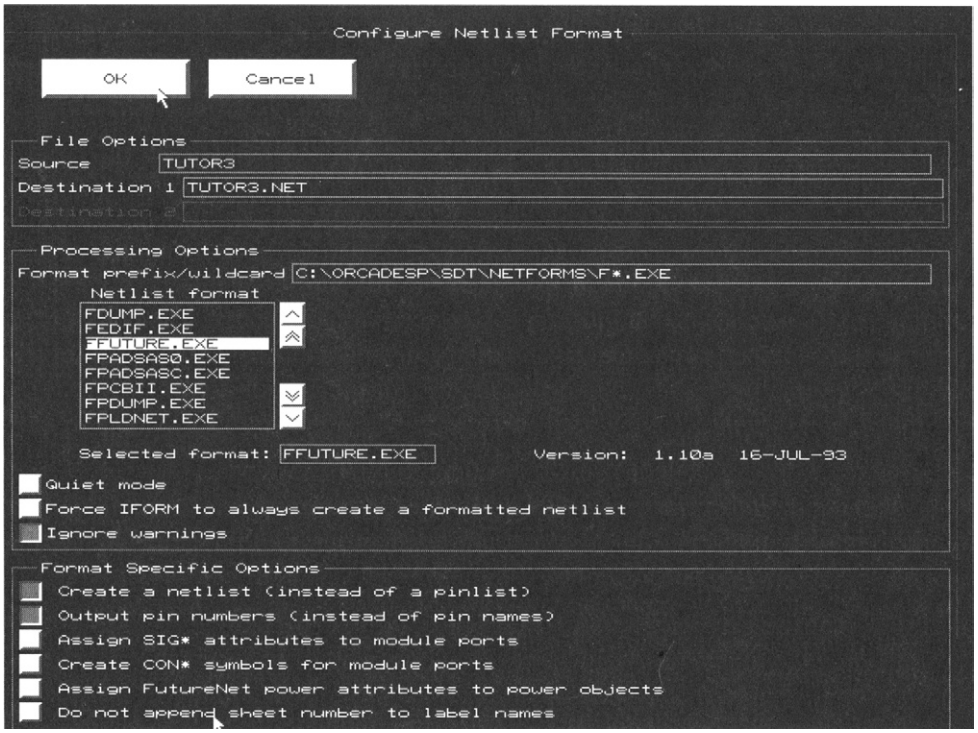


Figure 5-35 IFORM Configuration Options

The IFORM configuration screen has a File Options section. Note that unlike most other OrCAD routines, no filename extension is used for the file source. Only the filename prefix is entered and this must be the name of the design. The reason can be seen in Figure 5-30, as the input to IFORM consists of three files with extensions .INS, .RES, and .PIP.

One or two file destination entries are available. Some netlist formats result in two output files. Most netlist formats, including the FutureNet format used for the tutorial, result in only one output file. Unless you enter a different filename, OrCAD used the name of the design with .NET suffix.

The Processing Options section is used to select the desired netlist format and general processing options:

Format prefix/wildcard. This entry gives the path, prefix with * wildcard, and extension for all netlist format files. Unless your system has specialized netlist format files or a nonstandard subdirectory structure, this entry should be left unchanged.

Selected format box. Use the scroll buttons to scroll through the list of available netlist formats. Click on the desired format in the scroll box. The selection is highlighted and appears in the box. Use FFUTURE.EXE for the tutorial. This is the FutureNet format.

Quiet Mode. Suppresses display of text messages. Leave deactivated so that status messages will be displayed.

Force IFORM to always create a formatted netlist. Forces IFORM to process all files, even if none have been updated. Helps avoid the incremental update bug described on page 192. Not required if suggested steps in this tutorial exercise are followed and all intermediate files are erased before running Create Netlist.

Ignore warnings. This option is generally selected to prevent IFORM from terminating abnormally if a minor warning occurred.

Depending on the netlist format selected, additional options may appear in the Format Specific Options section. Note that not all netlist formats have processing options. Most of the options are fairly self-explanatory if one has some knowledge of the particular netlist format:

Create a netlist (instead of a pinlist). FutureNet format provides for both netlist and pinlist outputs. The netlist output is more widely used. The netlist output lists each net in the schematic along with the associated part pins. The pinlist output lists each pin on a part and the net the pin belongs to. Use the netlist output for the tutorial exercise.

Do not append sheet number to labels. Not all nets are associated with labeled signals. Normally, when a net is associated with a labeled signal, the label name is appended by an underscore followed by the sheet number on which the label appears. For example, the signal label D0 on sheet 1 becomes a net named D0_1. OrCAD labels are local to a given sheet unless the signals are explicitly tied together via module ports and sheet nets in the hierarchy. Signal D0 on sheet 1 can be electrically isolated from D0 on sheet 2. To avoid possible problems, such as inadvertently joining unrelated signals, this option should only be used with single sheet schematics.

Output pin numbers (instead of pin names). This option is usually selected since most PCB design systems use parts defined with pin numbers instead of pin names. However, not all OrCAD library parts have pin numbers. If this option is selected and the FutureNet formatter encounters a part without pin numbers (diodes and transistors are two common examples), the pin names are used.

FutureNet format associates a numerical attribute with nets. Some PCB design software packages allow specific design rules to be defined for signals with given attributes. For example, power and ground may be routed differently than normal signal connections. FutureNet attributes supported by OrCAD are listed in Table 5-1 below:

Table 5-1 FutureNet Signal Attributes

Signal Type	FutureNet Attribute
Generic Signal	5
Input Signal	10
Output Signal	11
Bidirectional Signal	12
Input Connector	24
Output Connector	25
Bidirectional Connector	26
GND	100
+5V	101
+12V	105
-12V	106
VEE	107

Additional options are provided that relate to these signal attributes. Any requirement for the use of these options will depend on the specifics of the PCB design software. None are used for this tutorial.

Assign SIG* attributes for module ports. All module ports are assigned signal attributes as listed above.

Create CON* symbols for module ports. All module ports are assigned connector attributes as listed above.

Assign FutureNet power attributes to power objects. Power attributes are assigned as listed above to any nets associated with power objects.

After you have verified the IFORM configuration options, click OK to exit. This completes the rather lengthy process of configuring the three Create Netlist routines, INET, ILINK, and IFORM. You can now launch the tool by double clicking on Create Netlist. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appeared during execution, there is usually no reason to examine this file.

Next, exit back to DOS. Use the DOS directory command to examine the design subdirectory and verify that all the intermediate files shown in Figure 5-30 have been created. Since Create Netlist has a bug associated with incremental updates, good practice dictates deleting all .INF, .INX, .INS, .RES, and .PIP intermediate files. Once the TUTOR3.NET netlist file has been generated, these intermediate files are of no value. Deleting these files will also save space and time when backing up the design directory.

Overview of FutureNet Format

Some understanding of FutureNet format will be very helpful if the requirement arises to check or edit a FutureNet netlist file. Examine the netlist file, TUTOR3.NET, using a text editor or the LIST utility. Figure 5-36 lists parts data. Figure 5-37 lists signal net data.

The netlist section shown in Figure 5-36 consists of an introductory header at the very beginning of the file and the parts listing. The introductory header contains data statements with the design name, drawing number, revision code, and last revision date. These are general data items useful for identifying the netlist file. Note that each data statement includes a numerical attribute.

For example the data statement:

DATA,50,PC INTERFACE CARD

uses numerical attribute 50 to identify the design name.

```
LIST        1              10-15-95 21:09 ♦ C:\ORCAD\TUTOR3\TUTOR3.NET
NETLIST,2
(DRAWING,ORCAD.NET,1-1
DATA,50,PC  INTERFACE CARD
DATA,51,1002
DATA,52,1.0
DATA,54,   August 21, 1995
)
(SYM,1-1,1
DATA,2,C1
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,2
DATA,2,C2
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,3
DATA,2,C3
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,4
DATA,2,C7
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,5
DATA,2,C8
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,6
DATA,2,C9
DATA,3,.1UF
DATA,4,.1UF
DATA,23,1
DATA,23,2
Command►    *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 5-36 FutureNet Netlist Parts Section

Immediately following the design information is the parts section. Parts are referred to as symbols (abbreviated SYM). All the data for a given symbol is enclosed in parenthesis. For example, the first symbol appears as:

Netlist Data	Data Type
(SYM,1-1,1	Symbol data for symbol 1
DATA, 2,C1	Reference designator
DATA 3,.1UF	Part value
DATA 4,.1UF	Module value (part value used)
DATA 23,1	Pin number
DATA 23,2	Pin number
)	

The top line identifies the data as symbol data. The character sequence 1-1 is a field separator that is used in many places throughout FutureNet netlist files. Each of the following data statements include a numerical attribute to identify the type of data explained above. Each symbol will include a list of pins. Attributes 21-23 identify symbol pins. Note that pin numbers appear in place of pin names.

```
LIST     397     770      09-14-95 19:59 ♦ C:\ORCAD\TUTOR3\TUTOR3.NET
(SIG,,IOR\_2,1-1,5,IOR\_2
PIN,1-1,19,U2,23,13
PIN,1-1,20,PL1,23,27
PIN,1-1,16,U1,23,1
)
(SIG,,***000002,1-1,5,***000002
PIN,1-1,16,U1,23,19
PIN,1-1,19,U2,21,22
)
(SIG,,RESET_2,1-1,5,RESET_2
PIN,1-1,14,U3,23,1
PIN,1-1,20,PL1,23,3
)
(SIG,,***000004,1-1,5,***000004
PIN,1-1,20,PL1,23,4
PIN,1-1,16,U1,22,9
)
(SIG,,***000005,1-1,5,***000005
PIN,1-1,20,PL1,23,6
PIN,1-1,16,U1,22,8
)
(SIG,,***000006,1-1,5,***000006
PIN,1-1,20,PL1,23,8
PIN,1-1,16,U1,22,7
)
(SIG,,***000007,1-1,5,***000007
PIN,1-1,20,PL1,23,10
PIN,1-1,16,U1,22,6
)
(SIG,,***000008,1-1,5,***000008
PIN,1-1,20,PL1,23,12
PIN,1-1,16,U1,22,5
)
(SIG,,***000009,1-1,5,***000009
PIN,1-1,20,PL1,23,14
PIN,1-1,16,U1,22,4
)
(SIG,,***000010,1-1,5,***000010
PIN,1-1,20,PL1,23,16
PIN,1-1,16,U1,22,3
)
(SIG,,***000011,1-1,5,***000011
PIN,1-1,20,PL1,23,18
PIN,1-1,16,U1,22,2
)
(SIG,,AEN_2,1-1,5,AEN_2
PIN,1-1,19,U2,23,11
PIN,1-1,20,PL1,23,22
Command►                                Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 5-37 FutureNet Netlist Signal Section

A section of the netlist signal data is shown in Figure 5-37. All the data for a given signal net is enclosed in parenthesis. For example, the first signal net appears as:

Netlist Data	Data Type
(SIG,,IOR_2,1-1,5,IOR_2	Net name, attribute 5
PIN,1-1,19,U2,23,13	Pin data for symbol 19, U2, pin 13
PIN,1-1,20,PL1,23,27	Pin data for symbol 20,PL1,pin 27
PIN,1-1,16,U1,23,1	Pin data for symbol 16,U1,pin 1
)	

The top line identifies the data as signal net data. The character sequence 1-1 is again used as a field separator. Note that the net name is taken from the OrCAD

signal label with the sheet number appended to it. Each of the following data statements are for symbol pins included in this net. Note that a pin attribute number (21-23) appears before each pin number.

Nets that are not associated with signal labels, such a the second net in Figure 5-37 are assigned numerical names such as ***000002.

Editing this netlist for PCB design would include replacing the module values in the parts (symbol) data section with the actual layout symbol names. Any alphanumeric pin names would also have to be changed back to numbers (none occur in the tutorial example). This would be done both in the symbol and signal net data sections. Finally, some pin numbers in the signal net data section might have to be changed if the physical pin arrangements did not match the OrCAD pinouts as explained on page 189. Any ASCII text editor could be used for this purpose.

Conclusion

Back up all the design data including the netlist file on a floppy disk. A subsequent tutorial in chapter 10 will use the TUTOR3 design as the basis for showing you how to edit netlists for transfer to PCB design.

You have now completed the third tutorial. At this point you have learned to use most of the features of Draft and important OrCAD postprocessing tools. You have also been introduced to basic netlist concepts.

6

Tutorial 4 - Library Editor

So far you have accessed existing library parts to complete the previous tutorial exercises. While OrCAD comes with an extensive parts library, sooner or later you will have to create new parts. One reason is that the electronics industry constantly introduces new components such as complex ICs. Also, many discrete components such as switches and transformers appear in an almost endless variety. The Edit Library routine is used to create new parts. Often a new part can be created by simply editing an existing part.

The library editor tutorial is divided into two sessions. In the first session, you will learn the basics by creating a transformer with center tapped windings. This new transformer will be created by editing an existing transformer in the DEVICE.LIB library. In the second session, you will create an IC not included in the OrCAD libraries, a Maxim MAX877 voltage regulator for battery powered systems.

Many of the skills you have learned in the previous tutorials are directly applicable to the library editor. The Edit Library main menu is quite similar to Draft. You will find that the ease of creating new parts is one of the major strengths of OrCAD. The tutorial will conclude with some tips on library management and an overview of related OrCAD routines.

Overview of Library Parts

OrCAD SDT 386+ stores all graphic data for library parts in vector form. Previous versions of OrCAD stored graphic data in both raster (OrCAD documentation used the term bitmap) and vector form and some early libraries only contained raster data. Raster and vector data was discussed in chapter 5. Raster data consists of patterns of dots called pixels. Vector data consists of entities representing straight lines, arcs, circles, and text characters.

Draft uses raster data to display parts on the screen. The PC graphics display is a raster device. Consequently, processing and displaying raster data is very fast. Raster data is also used by Print Schematic for printing hard copy. Vector data is used by Plot Schematic for plotting to pen plotters. Edit Library uses only vector data. Vector data is easier to edit.

Normally when a part is created or edited, Edit Library processes vector data. When the part data is required in one of the programs that use raster data, the vector form is converted on the fly. With SDT 386+, this process is entirely transparent to the user.

OrCAD supports three different types of parts:

- **Graphic parts**. Do not confuse this term with graphic data. All types of parts contain graphic data. OrCAD uses the term graphic parts for parts having bodies that can contain lines, circles, arcs, filled areas, and text. Examples of graphic parts include an OR gate and a resistor. Graphic parts do not display visible pin names. However, pin numbers can be displayed. Prior to SDT 386+, graphic parts were limited to 1.2 x 1.2 inches. SDT 386+ allows graphics parts of almost unlimited size.
- **Block parts**. These parts have a body that is limited to a square or rectangle. Most complex ICs are created as block parts. Both pin names and pin numbers are typically displayed. Block parts are limited to 12.7 x 12.7 inches.
- **IEEE parts**. These are specialized parts that comply with ANSI/IEEE standards. IEEE parts are limited to 12.7 x 12.7 inches. They are similar to block parts but can contain circles and special IEEE symbols used to denote logic functions. IEEE symbology originated in the early 1980s and did not evolve to meet the needs of today's complex VLSI devices. Few companies still use IEEE parts in new designs, and they are not covered in this tutorial. Of all major vendors, only Texas Instruments' data books still show some IEEE symbols. Unless dictated by company policy, avoid using IEEE parts.

Most libraries supplied by third parties for SDT +386 now contain vector data. No problems occur when editing parts in these libraries. Earlier, OrCAD and third party libraries contained both vector and raster data. OrCAD provides a conversion program for using these libraries with SDT 386+. At one point, some third party libraries contained parts without vector data. These libraries cannot be converted. To avoid conversion headaches, do not purchase any third party libraries unless the vendor guarantees they are fully compatible with SDT 386+. The term "third party" means supplied by an outside source. All of the libraries supplied on your OrCAD software disks are OrCAD libraries.

Starting the First Session

Start the library editor tutorial by creating a new design called TUTOR4 based on the standard template. Use the OrCAD Design Management tools as in previous tutorials. Then select TUTOR4 and launch OrCAD SDT.

At the main SDT screen, click on Edit Library and then click on Local Configuration as shown in Figure 6-1. This brings up the configuration screen shown in Figure 6-2.

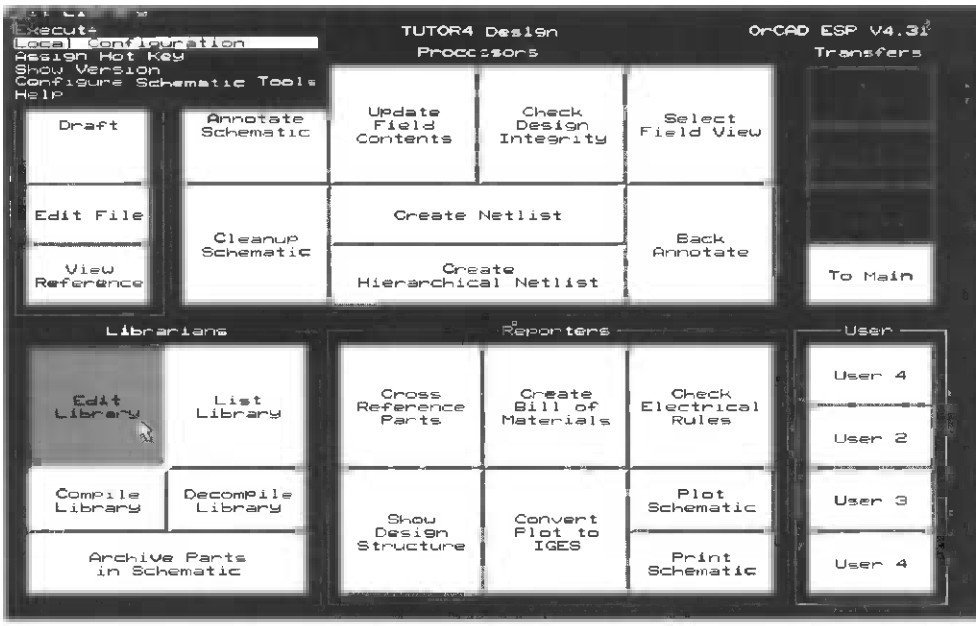


Figure 6-1 Selecting Edit Library Configuration

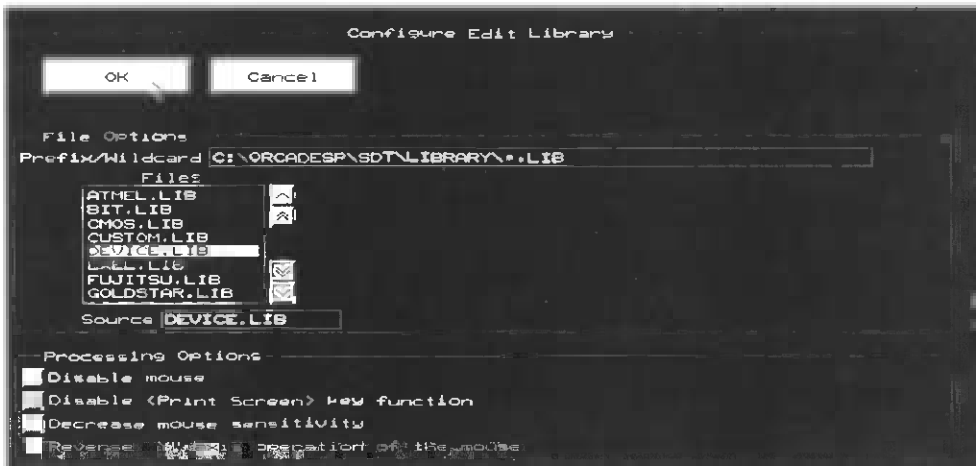


Figure 6-2 Setting Edit Library Configuration Options

Edit Library Local Configuration

Unlike the local configurations for other OrCAD tools, Edit Library has relatively few options. The file options section is used to select the library to be edited. The file options section and other processing options are examined in detail below:

Prefix/Wildcard. This entry gives the path, wildcard prefix (*), and extension used to search for all library files. Unless your system has a nonstandard subdirectory structure, this entry should be left unchanged. All of the libraries loaded onto disk during the installation process appear in the scroll box. During SDT configuration (refer to chapter 2) a subset of these libraries was selected to be available in Draft.

Source box. Use the scroll buttons to scroll through the list of available libraries. Click on the desired library in the scroll box. The selection is highlighted and appears in the source box. For now, select DEVICE.LIB.

Disable mouse and **Reverse "Y" axis operation of the mouse.** The rationale for these options is unclear.

Disable <Print Screen> key function. The print screen function does not work with all graphics boards, and the function is generally not very useful. An option is provided to disable print screen for compatibility with certain TSR screen capture utilities.

Decrease mouse sensitivity. When zooming in while using Edit Library, the cursor becomes very sensitive. If you are using a Microsoft mouse, an alternate approach is to use the Microsoft Control Panel to adjust the mouse sensitivity.

After you have selected the DEVICE.LIB library and verified the configuration options, click OK to exit. Then double click on Edit Library to launch the tool.

Library Editor Command Overview

After you have launched Edit Library, click the right mouse button to bring up the main menu shown in Figure 6-3. Edit Library has many similarities to Draft, including the use of the mouse to select menu commands. The Edit Library main menu is organized in the same manner as the Draft main menu. Edit Library and Draft also share many of the same commands.

Commands can also be selected by typing in the first letter of the command or by using macros. The previous tutorials introduced the use of Draft macros. Many

Draft commands such as Place Wire or Place Junction are so frequently used that the overhead associated with creating, learning, and remembering macro commands is justified based on the potential for a tremendous increase in productivity. Only a small fraction of the entire time spent on a design is spent using the library editor. A design that requires 10 hours to complete might only require 15-30 minutes of editing parts. Edit Library commands typically involve more submenu choices and text entry. These factors work against the efficient use of macro commands. In this tutorial, only the mouse is used for command selection.

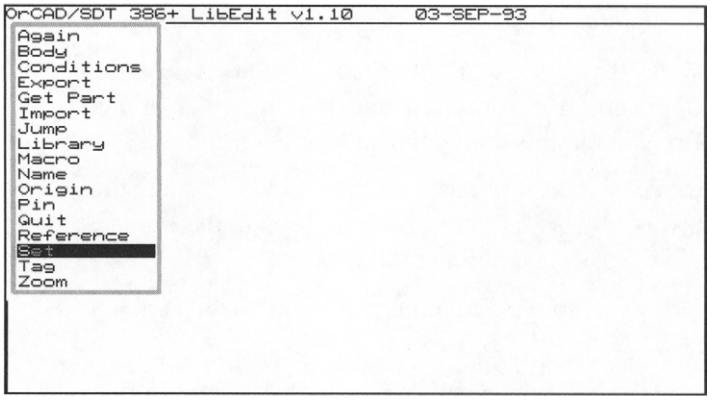


Figure 6-3 Edit Library Main Menu

A brief overview of the main menu commands follows. Most of the commands are explained in detail further on in the tutorial lessons. As with the Draft main menu, many of the commands on the Edit Library main menu are of limited use and can be ignored.

Again Repeats the last command, but has a flaw in that it only repeats the first menu level. For example if the last command was Body Graphic, using Again would only get you to the Body menu. You can forget about this command.

Body Commands on the Body menu are use to draw and edit the body of a part. This command has submenus for block, graphic, and IEEE type parts.

Conditions Another command you can forget about. Once used to check how much memory was left, back in the days before SDT 386+ when memory was limited to 640KB. This command has been stubbed off to show information

	about the part being created. This information is of little practical use.
Export	Writes data for the current part to a selected file. Useful for transferring part data to another library. Note that the part must have a name before using Export. Data is formatted as a library source file.
Get Part	Accesses the library and retrieves a part for editing.
Import	Reads part data from a library source file that was written using Export. Useful for transferring part data from another library.
Jump	Moves the cursor to a preset tag location (see Tag command), to a grid reference, or along the X or Y axis. Forget about this command and use your mouse.
Library	Accesses special library functions used to update the current library, list parts, browse through the library, delete a part, and define prefixes.
Macro	Used to capture macro commands and manage macro files.
Name	Used to add, delete, or edit names and assign prefixes to a part. Note that a part may have multiple names and prefixes.
Origin	Resets the X,Y origin at the current cursor position. Limited usefulness.
Pin	Commands on the Pin menu are use to add, delete, and edit the part pins.
Quit	Similar to the Quit command in Draft. Provides access to file management functions. Also used to exit from the library editor.
Reference	Used to specify and edit the reference designator prefix of a part.
Set	Allows changing parameters for numerous library editor options including display of body outlines, visible power pins, and grid dots.
Tag	Sets up to eight tag locations for use with the Jump command. Not frequently used.

Zoom Used to change the zoom scale. Normal scale is 1, which is adequate for most editing of parts. In some cases, zooming in on a detailed part may facilitate editing.

Before you proceed, spend a few minutes, select the most frequently used commands (Body, Export, Get Part, Import, Library, Name, Pin, Quit, Reference, and Set) and explore their submenu options. These options will be explained in more detail as they are introduced during the tutorial.

Creating the First Part

The first part you will create is the transformer with center tapped primary and secondary windings shown in Figure 6-4. The OrCAD-supplied DEVICE.LIB library already contains a transformer with center tapped secondary which will be used as the prototype. The procedure will be to export the prototype part from DEVICE.LIB, import the part into CUSTOM.LIB, edit the part to add the primary center tap pin, and then update and save CUSTOM.LIB with the modified part.

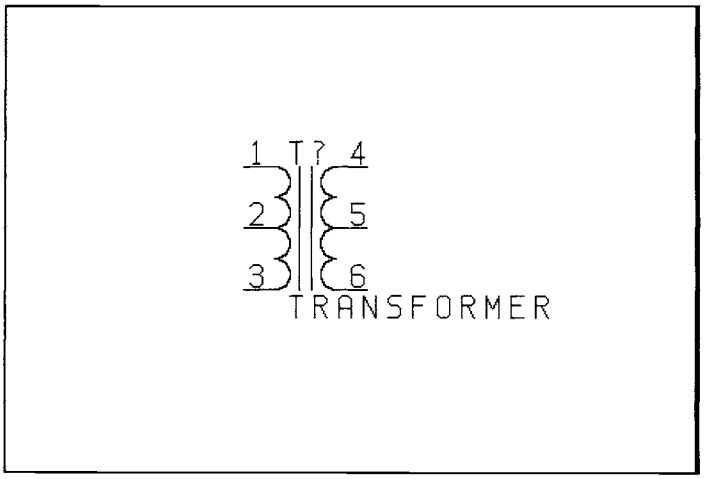


Figure 6-4 Transformer with Center Tapped Windings

Recall the caution in chapter 1 about editing OrCAD supplied libraries. If you edit an OrCAD supplied library, all the parts you modified or created in that library will be lost when you load a subsequent OrCAD software update that includes new library files. For this reason, the suggested practice is to always place modified or new parts in your CUSTOM.LIB.

Overview of the Set Command

One of the first steps whenever using the library editor is to turn on the grid dots display. Click on the Set command at the main menu, then click on Visible Grid Dots as shown in Figure 6-5 and finally click on the YES option.

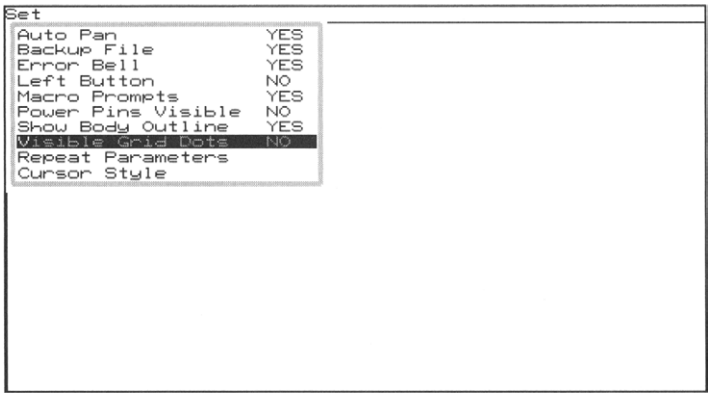


Figure 6-5 Selecting the Visible Grid Dots Command

Note that the Set command menu for Edit Library has many of the same options as in Draft, including Auto Pan, Backup File, Error Bell, Left Button, Macro Prompts, Repeat Parameters, and Cursor style. These options were discussed in chapter 5. New options unique to Edit Library include:

Power Pins Visible Displays power pins, which are normally invisible. Recommended practice is to always set this option on so that power pins are not inadvertently deleted or duplicated. In some cases, text associated with power pins may overlap other pin names and turning power pin display off is advantageous.

Show Body Outline Displays the body outline as a dotted line. Graphic parts have an overall body outline (the term border is more appropriate). Editing outside the body outline can cause unpredictable results. Always display body outlines when editing parts.

Getting and Exporting the Original Transformer

Figure 6-6 shows the library editor display area once grid dots have been made visible. The next step is to get the center tapped transformer. Use the Get command from the main menu. The Get command functions in much the same manner as in Draft, except that only the active library selected during local configuration of Edit Library is available.

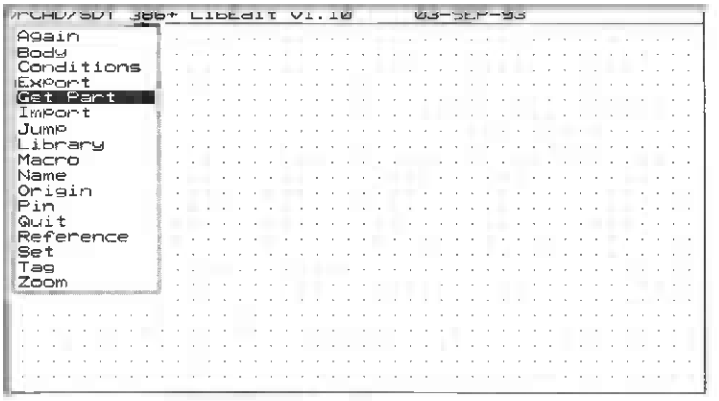


Figure 6-6 Selecting the Get Part Command

A scroll box appears as shown in Figure 6-7. Scroll down and click on TRANSFORMER CT (the CT stands for center tapped). The part will appear on the screen as shown in Figure 6-8 The active display area shown by the grid dots is automatically adjusted, based on the size of the part.

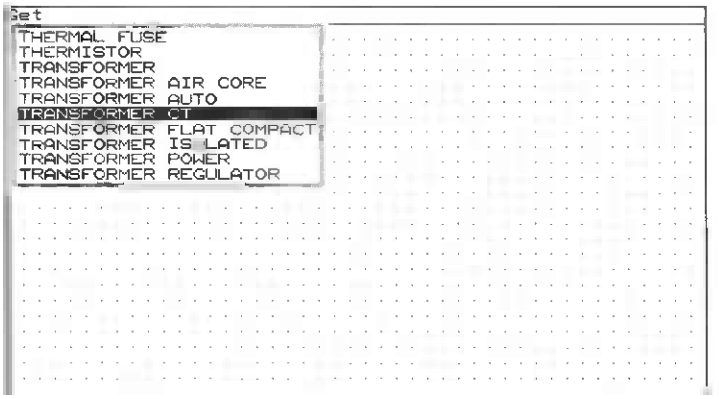


Figure 6-7 Selecting the Center Tapped Transformer

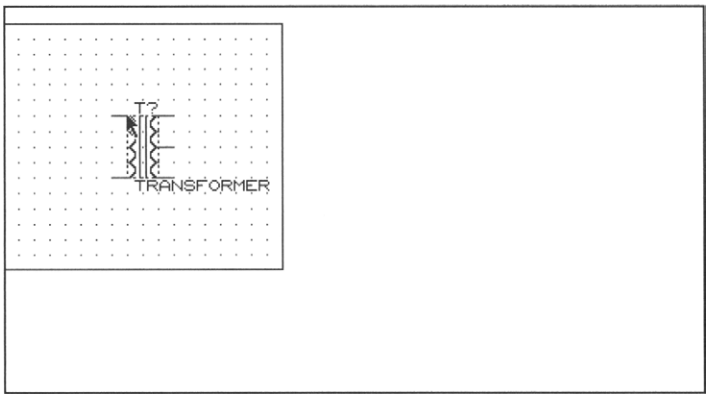


Figure 6-8 Prototype Transformer

The next step is to export the transformer part data to a file. Use the export command from the main menu as shown in Figure 6-9. One disconcerting aspect of OrCAD is that the menus overlap and obscure the active screen display area. This is more noticeable in Edit Library than in Draft. Complete the export process by entering a filename as shown in Figure 6-10. Use the name XFORMER.PRT. Data for the part will be written to this file in the TUTOR4 design directory.

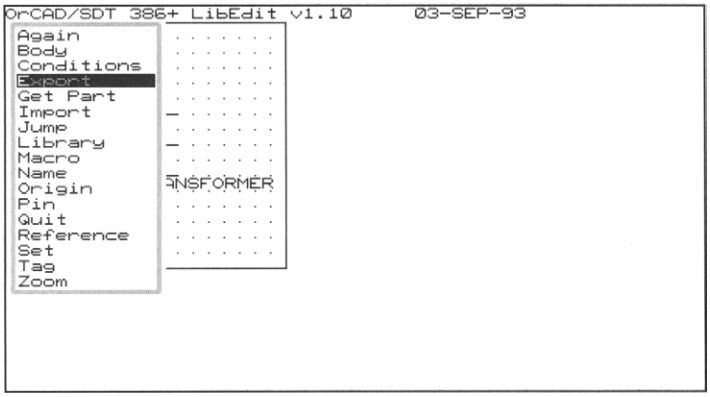


Figure 6-9 Selecting the Export Command

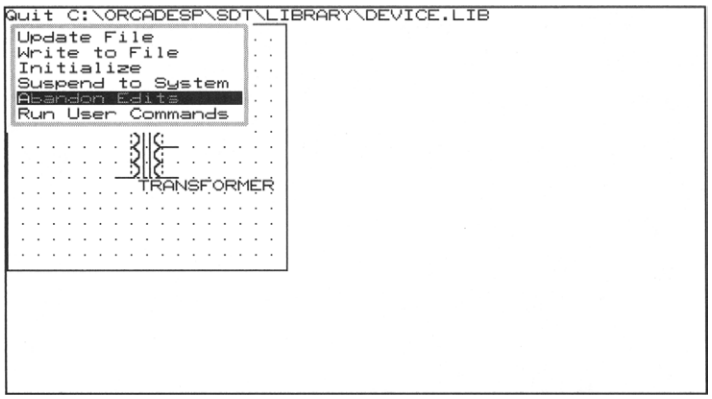


Figure 6-10 Entering the Export Filename

Overview Of The Quit Command

This completes the initial task of exporting the part to be edited from the source library. Quit the library editor. Click on Quit from the main menu and then Abandon Edits as shown in Figure 6-11.

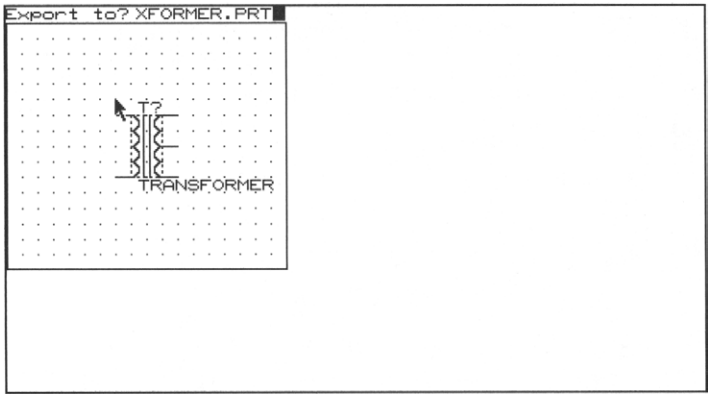


Figure 6-11 Selecting Abandon Edit to Exit the Editor

In previous Draft tutorials, you have always used the Update File option and the other Quit options were not examined. Let's take a closer look at these options and their usage:

Update File Writes the library or schematic sheet data to disk with all edits made during the session. Always use this option to save your work **before** exiting via Abandon Edits. Good practice is to also use Update File

	whenever a significant task has been accomplished (to save recent work in case of system crash or power failure) or before a major edit (the library editor has no undo capability).
Write to File	Same action as Update File except this option is used to write data to a specified file.
Initialize	Abandons any edits since the last file update and then prompts user for the name of a new library.
Suspend to System	Suspends program operation and loads the DOS command interpreter for DOS operations. Refer to the command reference on page 381 for more details.
Abandon Edits	Ends the editing session and exits back to the main SDT screen. Does not save any edits. You must use Update File before Abandon Edits in order to save your work. If any edits were made during the session, Abandon Edit asks for a YES/NO confirmation before exiting.
Run User Commands	Exits to DOS and runs an external program or batch file. Refer to the command reference on page 381.

Editing the Transformer

Now that you have exported the original part, the next step is to load CUSTOM.LIB into the library editor, import the transformer, make the required edits, and update the library.

Go back to the Edit Library local configuration screen, select CUSTOM.LIB, and the launch Edit Library. Repeat the setup process. Use the Set command options to select visible grid dots, power pins, and body outline.

Use the Import command from the main menu to import the transformer part. The command sequence is similar to the Export command used in the preceding section. Enter the same filename XFORMER.PRT that you previously used to export the part. The prototype transformer will appear on the screen.

The next step is to edit the body of the part. In the process, you will learn more about the Body command. From the main Edit Library menu, select the Body command as shown in Figure 6-12. This action brings up the options submenu shown in Figure 6-13.

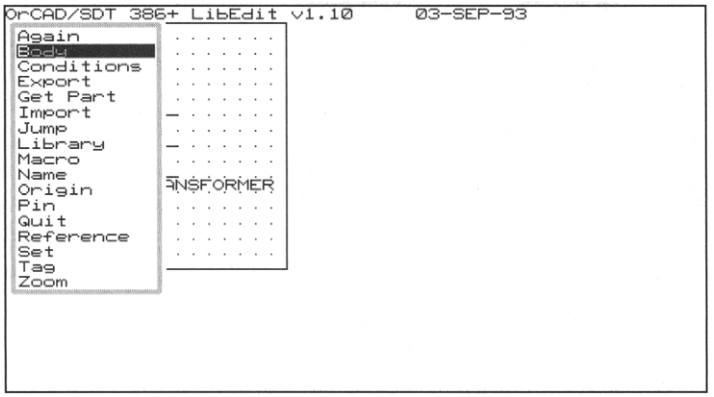


Figure 6-12 Selecting the Body Command

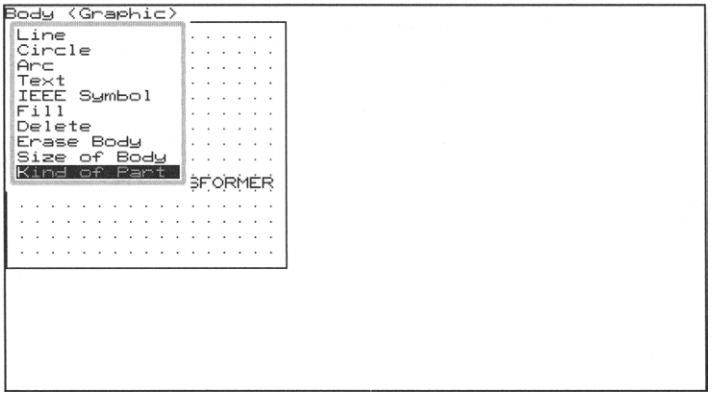


Figure 6-13 Body Command Options

Overview Of The Body Command

The prototype transformer is a graphic part and the library editor identifies it as such at the top of the screen. The type of part determines the options available when the Body command is selected. Graphic parts have the widest range of options. Block parts are limited to two options: Size of Body and Kind Of Part. Details of the Body command options are listed below:

Line Draws a line. Functions similar to the Place Wire command in Draft.

Circle Draws a circle within the part outline. Start by clicking the mouse cursor on the center point. The circle then expands as the cursor is moved out. Click the mouse again to define an edge point.

Arc Draws an arc ranging from zero to 90 degrees within the part outline. Start by clicking on the center, as with the Circle command. Initially, a circle appears and expands as the cursor is moved out. Click the mouse again to define the first point on the arc and repeat for the second point. Note that the endpoints of the arc must remain within the same quadrant (defined by imaginary horizontal and vertical lines passing through the center point of the arc).

Text Draws text within the part outline. Functions similar to the Place Text command in Draft. Primarily used for symbolic labels, such as the "+" polarity label on an electrolytic capacitor. Do not use the Text command for pin numbers, pin names, part name, or reference designator prefix. These entities are separately defined.

IEEE Symbol Draws IEEE/ANSI symbols within the part outline. A menu appears with a listing of available symbols. IEEE symbols are no longer widely used and are not reviewed in detail in this book.

Fill Draws a solid fill pattern within an enclosed area. Position the cursor within the enclosed area and then select the Fill command. The triangular filled-in area of a diode symbol is created using this command. Note that drawing or editing any graphics on a part will cause all fill patterns to be deleted. Use Fill only as a final step after all other graphics are completed.

Delete Deletes the graphic object. Functions similar to the Delete command in Draft. Note that unlike Draft, the library editor does not provide an undo option for Delete.

Erase Body Deletes all objects within the part body. Note that an undo option is not provided.

Size of Body Changes the size of the part body outline. The mouse cursor drags the lower right hand corner. Click to place the corner in a new location. Unpredictable results can occur if the outline size of an existing part is reduced.

Kind of Part Changes part type (block, graphic, or IEEE). Also used to access additional body parameters for each part type.

Note that changing from one type of part to another (for example graphic to block) generally causes any previous changes and part data to be lost.

Editing the Transformer Body to Display Pin Numbers

Graphic parts can display pin numbers but not pin names. The prototype transformer does not display pin numbers. A modification to the part body parameters will be required to display pin numbers. Click on Kind of Part as shown in Figure 6-13. The next submenu shown in Figure 6-14 appears.

To modify the body parameters, you need to entirely redefine the part body. The library editor makes no provision for accessing and editing individual parameters. Select Graphic Kind of Part as shown in Figure 6-14.

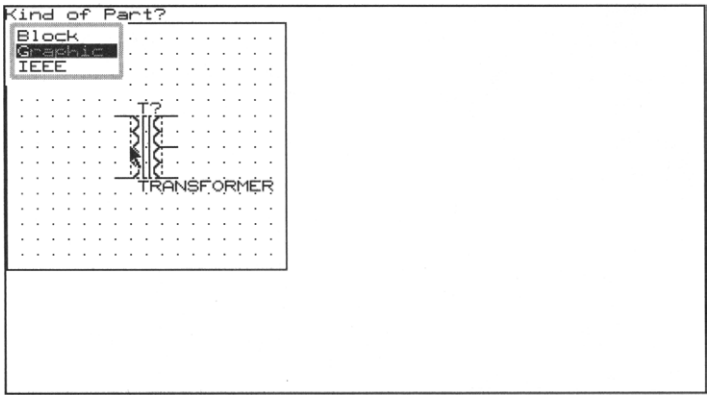


Figure 6-14 Selecting Graphic Kind of Part

The screen shown in Figure 6-15 allows defining the part as a grid array. Recall from previous discussion that grid array parts have pins arranged in rows and columns with alphanumeric pin numbers. Select NO, since the transformer is not a grid array part.

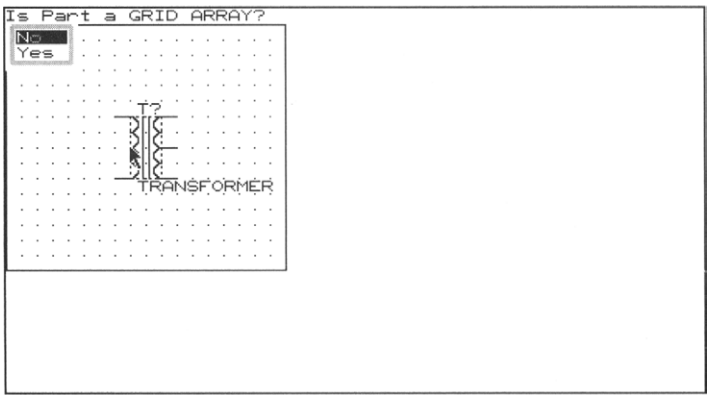


Figure 6-15 Setting the Grid Array Parameter

The screen shown in Figure 6-16 appears. This screen prompts for the number of parts per package. Many logic devices, such as a common 74HCT14 hex inverter, have multiple parts per package. Once you draw the first body of a multiple part package, the library editor automatically creates the remaining sections and assigns reference designator suffixes such as U?A, U?B, U?C, and so forth to these sections.

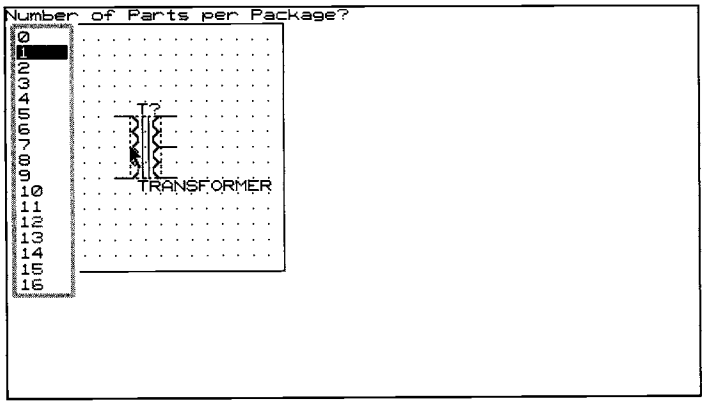


Figure 6-16 Setting the Number of Parts Per Package

Note that zero parts per package is one of the setup options. The library editor uses zero parts per package to define a part body that does not display pin numbers. Most discrete devices found in the DEVICE.LIB library do not display pin numbers.

There is no physical significance to zero parts per package and a menu selection such as "Part body without pin numbers" would certainly have been easier to understand. A useful mnemonic is "zero parts equals zero pin numbers."

The prototype transformer was originally created with a zero parts per package body in order to suppress pin numbers. The new part you are creating will show pin numbers, so select one part per package.

The final screen, used to define body parameters, appears as shown in Figure 6-17. This screen asks if the part has a converted form. Converted forms are primarily intended to be used with simple logic gates that have DeMorgan equivalents. For example, the DeMorgan equivalent of a NAND gate is an OR gate with negated inputs. If you select YES for this option, the library editor allows you to assign a second graphic body for the part. Because the transformer has no DeMorgan equivalent, select NO.

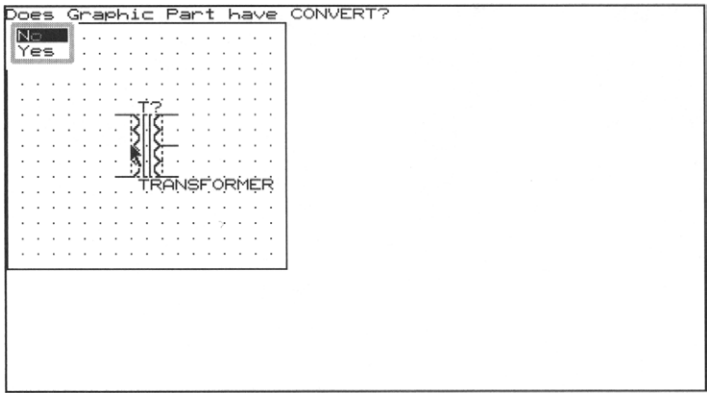


Figure 6-17 Setting the Part Convert Option

After the last of the graphic body options has been defined, the transformer reappears with pin numbers as shown in Figure 6-18.

Editing Part Pins

The next step is to edit the pin numbers to match those shown for the completed part in Figure 6-4 and to add the center tap pin on the primary (left) side. Start by editing the existing pin numbers. Place the cursor on the top left pin as shown in Figure 6-18 and then select the Pin command from the main menu as shown in Figure 6-19.

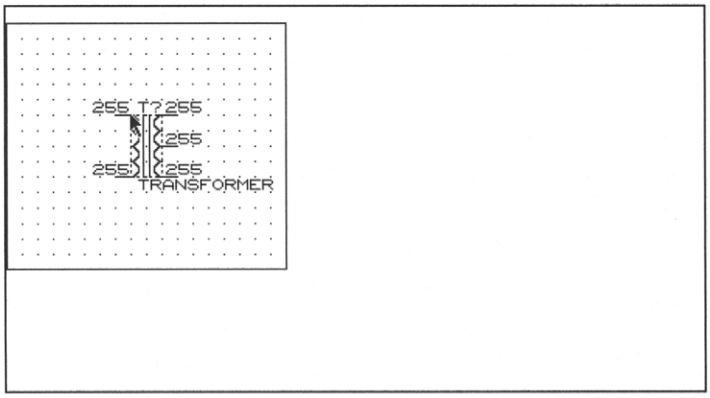


Figure 6-18 Transformer with Visible Pin Numbers

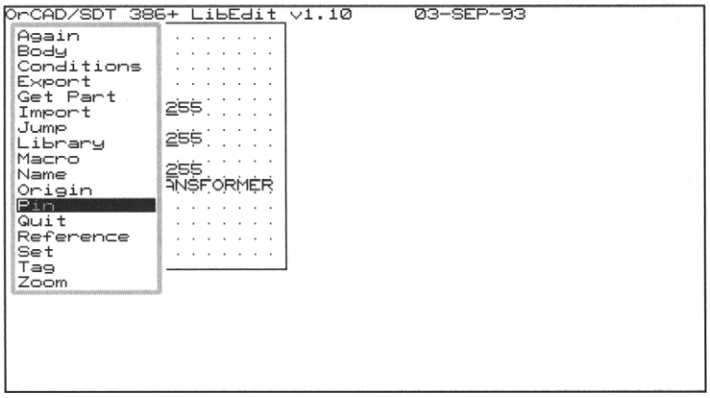


Figure 6-19 Selecting the Pin Command

Overview of the Pin Command

The pin command is used to add, delete, and edit pins. The command functions in essentially the same manner for graphic, block, or IEEE body parts. The available options are shown in Figure 6-20. When a pin is added, the library editor automatically steps through the process and prompts for the information required for each step. Unlike part body parameters, individual pin attributes can easily be edited using the available command options.

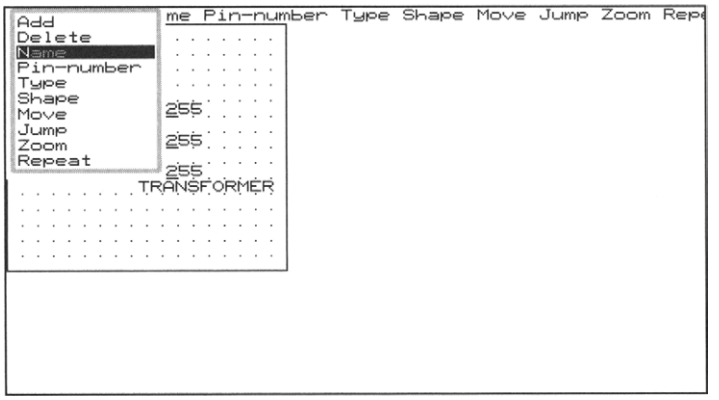


Figure 6-20 Pin Command Options

The following options are available with the pin command. Except for the Move option, the cursor should be located at the desired position (where a pin is to be added, deleted, or edited) before using the Pin command.

Add Adds a pin at the cursor location. The library editor then prompts for the name, number, type, and shape attributes (these are explained in detail below).

Delete Deletes the pin at the cursor location. Note that an undo function is not available.

Name Edits the name of an existing pin. All pins must have a defined name. Pin names are not visible on graphic parts.

Pin-number Edits the number of an existing pin. Pin numbers are only visible on graphic parts if the part is defined as having zero parts per package.

Type Edits the electrical type definition of a pin. The Check Electrical Rules tool uses pin type information to check for electrical interconnection rule violations as defined by a decision matrix (refer to chapter 5 page 174). A menu of pin type options appears as shown in Figure 6-24 and the current type is listed at the top of the display. Available pin types include:

Input. Signals are applied to a part via input pins. For example, pin 1 of a 74HCT14 hex inverter is an input pin.

Output. The part generates a signal at output pins. For example, pin 2 of a 74HCT14 hex inverter is an output pin.

Bidirectional. The pin can be either an input or output, depending on the internal state of the part. For example, pin 2 of a 74HCT245 bus transceiver is a bidirectional pin.

Power. Supplies power to the part and is automatically connected to ground or one of the supply rails. Power pins are invisible in Draft. The name of the power pin, such as GND, VCC, VSS, or VDD, determines which power or ground plane the pin is connected to.

Passive. Typically used with passive devices such as resistors or discrete semiconductors without external power connections. You can also use passive pins for visible power connections on ICs.

3 State. Special logic output signal with three states: active low, active high, or off (high impedance). For example, pin 2 of a 74HCT373 latch is a three state pin.

Open Collector. Special logic output signal that can only sink current (such as the collector of a NPN transistor with emitter grounded). Requires an external pull up resistor. Multiple open collector outputs can be "wired OR" together. For example, pin 2 of a 7406 hex inverter is an open collector pin.

Open Emitter. Special logic output signal that can only source current (such as the emitter of a NPN transistor with collector tied to VCC). Primarily found in high speed ECL logic, such as the Motorola MECL 10K/10H logic series. Requires an external resistor network for proper termination.

Shape

Edits the shape of a pin. Note that the shape of a pin has no effect on electrical properties. Pin shapes are shown in Figure 6-25. The current shape is listed at the top of the display. Available pin shapes include:

Line. A normal pin with a length of three grid units. Except for the short pin, all other pins are three grid units long.

Clock. A pin with the triangular clock symbol appearing inside the part outline. Used for clock inputs on logic devices.

Dot. A pin with the circular negation symbol appearing outside the part outline. Used for inverted inputs (active low) on logic devices. Also handy for drawing switches.

Dot Clock. A pin with both the circular inversion symbol and triangular clock symbol. Use for inverted clock inputs on logic devices.

Short. A normal pin with a length of one grid unit. Primarily used with passive and discrete devices drawn as graphic parts.

Move Used to move the location of an existing pin. Unlike the other Pin commands, start the command first. The Move command then prompts for the pin to be moved. Click on the selected pin. Then move the cursor and click on the desired location.

New users are often bewildered by the range of pin type options. If you are creating a part and are uncertain about which pin type is correct, use passive pins. Passive pins are general purpose and never cause electrical rules violations when connected to other types of pins or objects.

Invisible power pins can be a real nuisance on modern designs using multiple power levels. An easy solution is to import standard logic parts into your custom library and edit the power pins to assign them a passive type attribute. This makes the power pins visible and later allows directly connecting them to a particular supply voltage.

OrCAD uses two conventions to represent inverted signals. In Draft, labels and module ports use the backslash (\) convention for denoting inverted (or active low) signals. In Edit Library, pin names for inverted signals use the overbar ($\overline{\ }$) convention. The overbar convention is also used by most IC manufacturers to represent inverted inputs on device data sheets. To enter a pin name with an overbar, type a backslash after every character (for example C\L\K\).

Pins on block parts such as ICs always have both a defined pin name and pin number. The pin name describes the pin function and the pin number is determined by the physical location of the pin on the device package.

Pins on graphic parts have a defined pin name, which is always invisible. Pins on graphics parts can have a visible pin number. For graphic parts that display pin

numbers, suggested practice is to also use the same number for the pin name. This will avoid possible problems and confusion when generating a netlist.

Editing the Transformer Pin Names

After selecting the Pin command, click the left mouse button to bring up the option menu shown in Figure 6-20. To edit the top left pin, click on Name as shown in Figure 6-20. You are prompted for the pin name. Make the pin name the same as the pin number shown in Figure 6-4. Enter the character **1**.

Click the left mouse button to bring the menu up again. Click on the Pin number as shown in Figure 6-21. You are prompted for the pin number. Enter the character **1**.

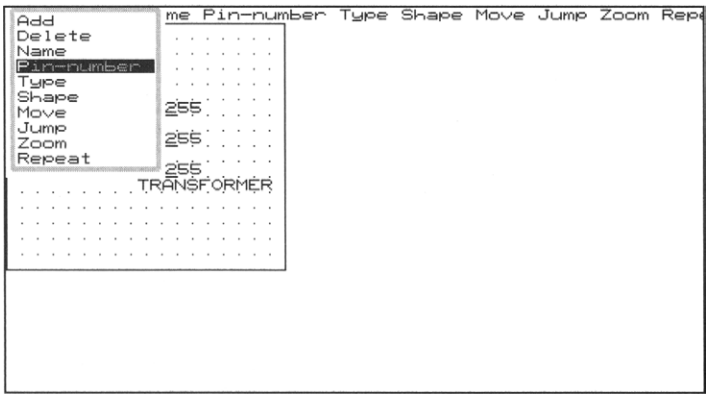


Figure 6-21 Selecting the Pin Number Option

Use the right mouse button to escape from the Pin command. Locate the mouse cursor on the next pin and repeat the editing process explained above. Edit all the existing pins. When you have completed the editing task, the transformer will appear as shown in Figure 6-22.

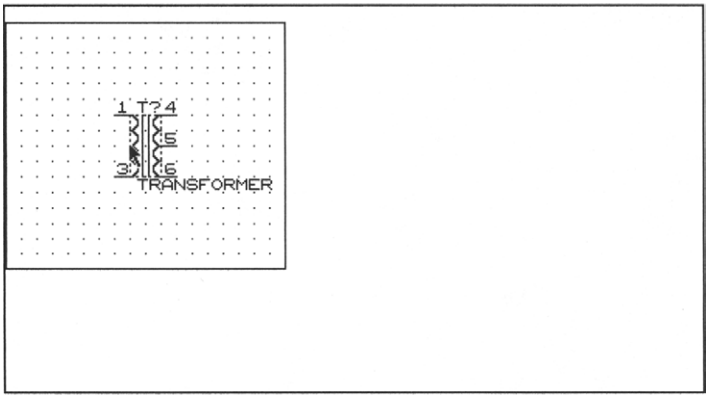


Figure 6-22 Transformer with Existing Pins Numbered

Adding the Center Tap Pin

The next step is to add the primary center tap pin. Locate the cursor at the center of the primary winding (left side) where the center tap pin is going to be added. Select the Pin command and Add option as shown in Figure 6-23. The library editor then prompts for the remaining pin attributes. Enter the new pin name and number as you did in the previous section.

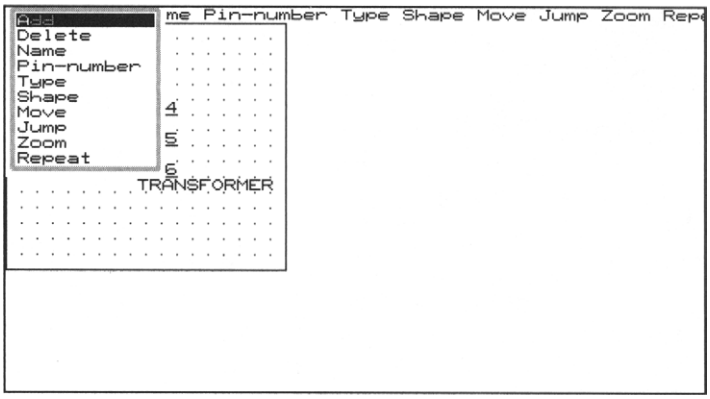


Figure 6-23 Selecting the Pin Add Command

The screens shown in figures 6-24 and 6-25 prompt you for the pin type and shape. Select passive type and short shape as shown in these figures. You have now completed the body and pin edits. The transformer should appear exactly as shown in Figure 6-4.

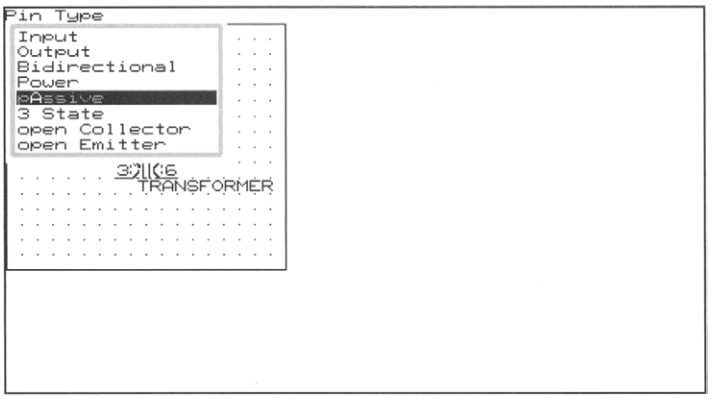


Figure 6-24 Pin Type Options

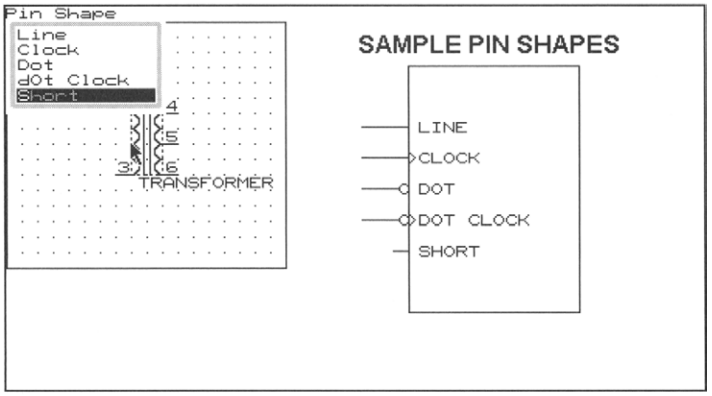


Figure 6-25 Pin Shape Options

Editing the Part Name

After completing the body and pin edits, the last step is to edit the part name. A given part can have multiple names. Names can be added, deleted, or edited. A list of prefixes can also be associated with the part name. This feature is useful for logic gates.

Many logic gates have standardized pinouts and are available in a range of logic families. For example the original TTL 7402 quad NOR gate is also available in other logic families such as Schottky (74S02), low power Schottky (74LS02), and CMOS (74C02). To save time creating parts libraries, OrCAD allows defining a list of prefixes for each library. The last several digits of the part, such as "02" for the quad NOR gate, are used as the part name. A list of prefixes, such as 74, 74S, 74LS, are then defined as available prefixes for the library. The Name Prefix command is used to display the list of available prefixes and determine

which prefixes are allowed for the particular part. In Draft, the user can enter the last digits of the part number and a selection list of complete part names with prefixes automatically appears.

New logic families are constantly appearing. In the last several years, the trend has been to 3V and 3.3V logic for reduced power consumption and radiated noise. If a new design required one of these low voltage parts, such as a 74LVQ02, the suggested approach is to get and place any "02" variant and then edit the part description. This approach would certainly be more effective from a time standpoint than transferring a part from the TTL.LIB to the CUSTOM.LIB and editing the available prefixes (never edit the OrCAD supplied libraries).

To proceed with this tutorial, click on the Name command from the main menu as shown in Figure 6-26. This brings up the option menu shown in Figure 6-27.

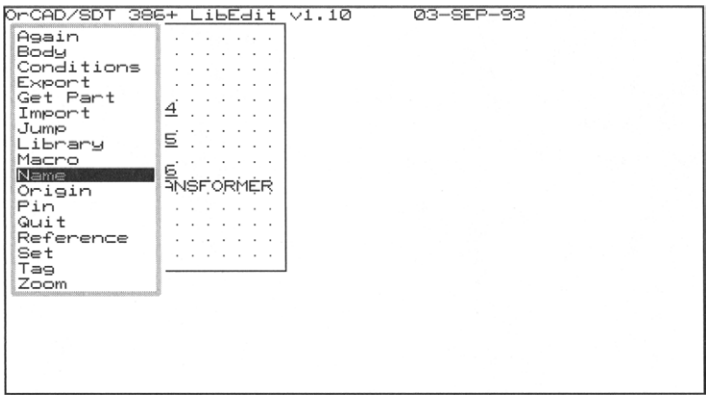


Figure 6-26 Selecting the Name Command

The library editor appears to have a minor bug that sometimes prevents editing a long part name. In the case of the transformer, delete the existing name by using the Delete option shown in Figure 6-27. Then use the Name command again and select the Add option to add the part name. Use the name TRANSFORMER CT/CT. This is shorthand for a transformer with center tapped primary and secondary.

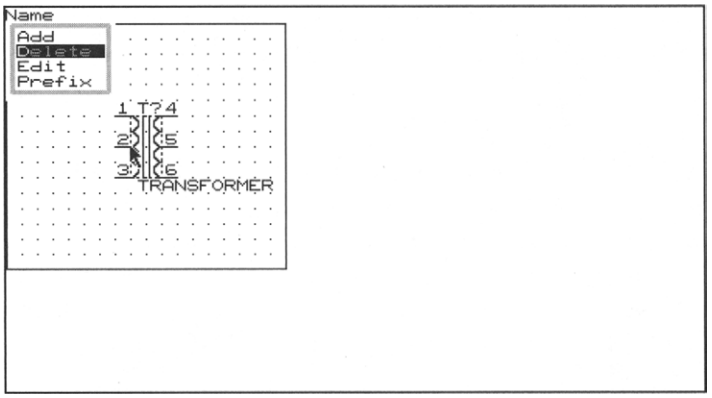


Figure 6-27 Name Command Options

Updating the Library

The transformer edits are now complete and the last step is to update the library. Unlike saving a schematic update in Draft, updating a library after editing requires two separate steps. New users often forget this fact and lose their work.

After a part has been created or edited, the part must first be saved into the library. Use the Library command as shown in Figure 6-28 and then select the Update Current option, as shown in Figure 6-29, to save the transformer. If the library editing session involves multiple parts, this step must be carried out after completing the work on each part. Before exiting the library editor, the Quit command and Update File option is used to save the library to disk.

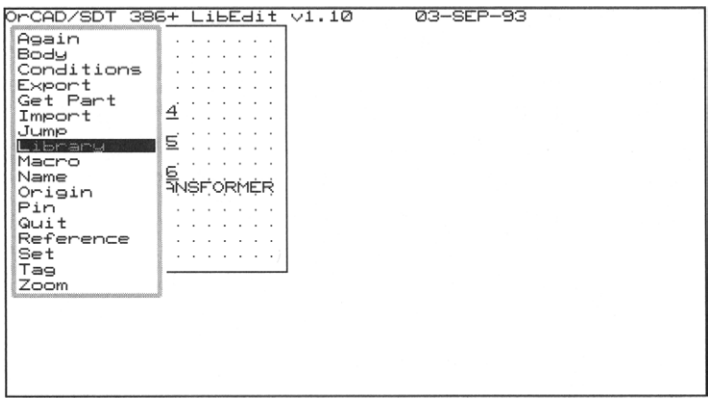


Figure 6-28 Selecting the Library Command

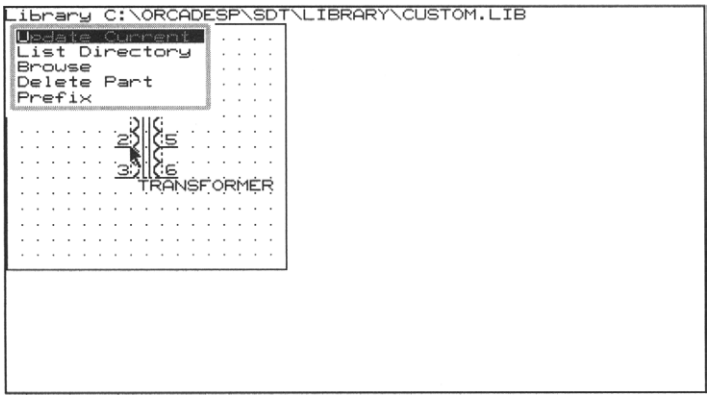


Figure 6-29 Library Command Options

Overview of Library Command Options

The library command has a number of important options. Let's examine these in more detail:

Update Current This command option must be used when a part has been created or edited in order to save the part information into the library. Only the library data in RAM memory is updated with this option. You must also use the Update File option of the Quit command to save the library to disk. **Both steps must be completed or data will be lost**.

List Directory Lists all parts in the library. A submenu allows listing the directory to the screen, the printer, or a file.

Browse Can be used for scrolling through the list of available parts. Browse cannot retrieve parts for editing. Note that the Get command also allows scrolling through the list of parts.

Delete Part Used to delete a part from the library. The part name (suffix only for parts with defined prefixes) can be directly entered. If the <ENTER> key is pressed without specifying a part name, the entire parts list is displayed. You can then scroll through the list and select the part to be deleted. If a part has multiple names, you must delete every part name in order to delete the part itself.

Prefix Used to edit the list of available prefixes. The existing list is displayed and a submenu provides Add, Delete, Edit, and Quit options. The list can contain up to 16 prefixes. The Quit option saves changes and returns you to the Library menu.

While a clear understanding of library concepts such as the use of prefixes is important, most users will not create extensive new libraries of logic parts. In most cases, users create and maintain a custom library which serves as a "catch all" for any parts not found in the OrCAD -supplied libraries. Assigning a list of prefixes to a custom library is probably a waste of time. Just use a basic part number in the library and then edit the part description after placing the part in Draft.

Wrapping up the First Session

Make sure that you have updated the library as explained in the previous section and then used the Quit command and Update File option to save your work.

As a final step, exit to DOS and backup your modified CUSTOM.LIB parts library onto a floppy disk. As with design files, recommended practice is to always backup modified library files at the end of every editing session. The path to the OrCAD library files is shown in chapter 2 on page 58. The path will also appear in the file section of the Edit Library local configuration.

You have now learned most of the basic library editor functions and how to create and edit a graphic part.

Second Session

The second session starts with the creation of a new part, an IC with a block body. Block bodies are used for most complex IC functions and are considerably easier to create than graphic bodies. This session includes a discussion of other OrCAD tools with functions related to parts library management: Archive Parts in Schematic, List Library, Compile Library, and Decompile Library.

Compile Library and Decompile Library were intended to facilitate off-line parts library preparation using a text editor. A valid rationale existed for the use of these tools at a time when fast PC workstations with high resolution graphics were expensive and not widely available. Because these tools are no longer applicable in today's design environment, they are only briefly discussed.

Custom parts libraries can represent a considerable investment in time. This session concludes with some helpful tips on parts library management.

Starting the Second Session

Use the OrCAD Design Management tool to select the TUTOR4 design, as in the previous session. Launch OrCAD SDT and then double click on Edit Library to launch the library editor. Note that the previous local configuration for Edit Library was saved, including the selection of CUSTOM.LIB as the active library for editing. As suggested for all new parts, the part you are creating should be saved into the CUSTOM.LIB library.

Creating the New Part

The new part you will create is a Maxim MAX877 voltage regulator IC. This part uses a novel switching regulator technology to generate a constant 5.0 volt supply from a 1.2 to 6.0 volt input and is ideal for battery powered systems using four alkaline or Nicad battery cells. The MAX877 is not in the OrCAD-supplied libraries and will be required for the Advanced Tutorial in the next chapter. The Advanced Tutorial includes a design with multiple, isolated power supplies.

Figure 6-30 shows the finished MAX877 IC. Signal names and pin arrangements are taken from the Maxim data book. Recommended practice for analog and power ICs is to avoid invisible power and ground pins, thus all pins on the MAX877 are visible.

Steps involved in creating the new part include: defining the body type and size, adding the pins, editing the reference designator prefix, naming the part, and updating and saving the library.

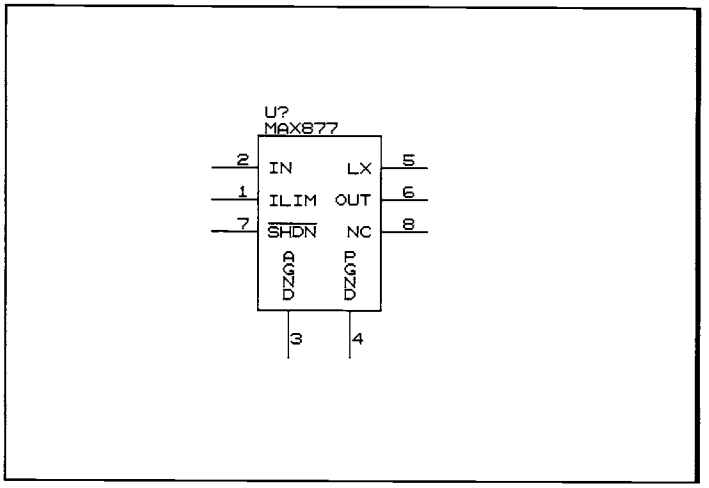


Figure 6-30 MAX877 Voltage Regulator IC

Creating a Block Body for the MAX877

Unlike the graphic part body used for the transformer in the previous session, a block body is limited to a simple rectangular or square shape. No additional lines, circles, arcs, or text may appear in a block body. Block bodies are appropriate for most complex ICs. Block bodies are also appropriate for circuit modules, such as DC/DC converters or hybrid amplifiers.

Bring up the Edit Library main menu. Click on the Set command and set visible grid dots to ON. Then escape back to the main menu and click on the Body command. Click on Block. The next screen allows you to select the number of parts per package. As with a graphic body, entering zero parts per package results in a part without visible pin numbers. Since the MAX877 requires visible pin numbers, select one part per package.

The following screen allows you to select whether or not the part is a grid array. Grid array parts display alphanumeric pin numbers. Select NO, since the MAX877 is not a grid array part.

The final block body setup screen appears. This allows you to set the size of the body. Size measurements are based on grid units, which are usually set at .1 inch as part of the OrCAD SDT configuration. The X and Y axis dimensions of the body appear in the upper right corner of the display. The lower right corner of the body outline moves with the mouse cursor. Position the mouse so that the body has dimensions of X=8 and Y=11 grid units as shown in Figure 6-31. Double click the right mouse button to place the lower right corner and complete the body.

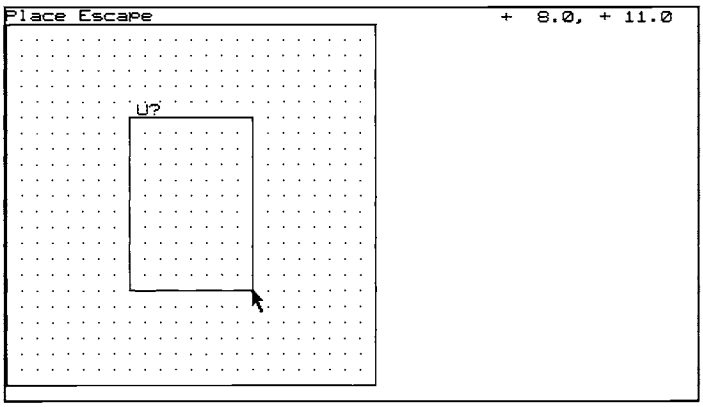


Figure 6-31 Setting the Body Size for the MAX877

Adding Pins to the MAX877 Body

The next step is to add the pins to the block body you just created. Click on the Pin command from the main menu. Then locate the mouse cursor at the location for the top left pin, which is pin 2. This is two grid locations down from the top corner. Click on the Add option. Then enter the pin name and pin number, using Figure 6-30 as a model. Select passive type and line shape. The first pin appears as shown in Figure 6-32.

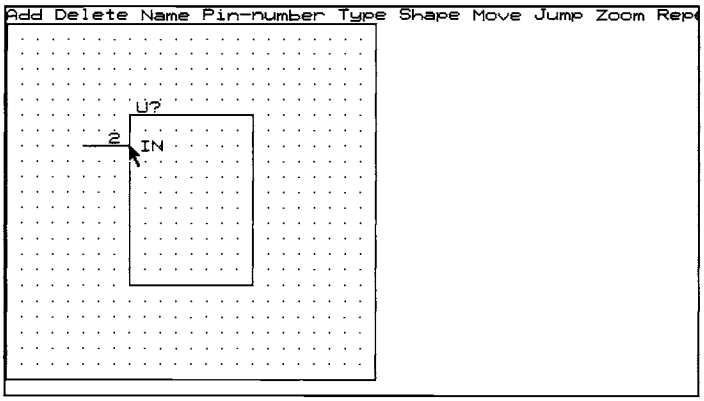


Figure 6-32 MAX877 with Pin 2 Added

The Pin Add command remains modal until you escape. You can move the cursor down to the location for pin 1 and click the right mouse button to add the next pin. The library editor makes adding pins very easy. Use the same passive type and line shape parameters for all the remaining pins. Note that the name for pin 7 requires an overbar to denote an inverted signal. Recall that an overbar

requires entering a backslash after each character in the name. Enter the name for pin 7 as **S\H\D\N\.** The completed MAX877 body with all pins is shown in Figure 6-33 below.

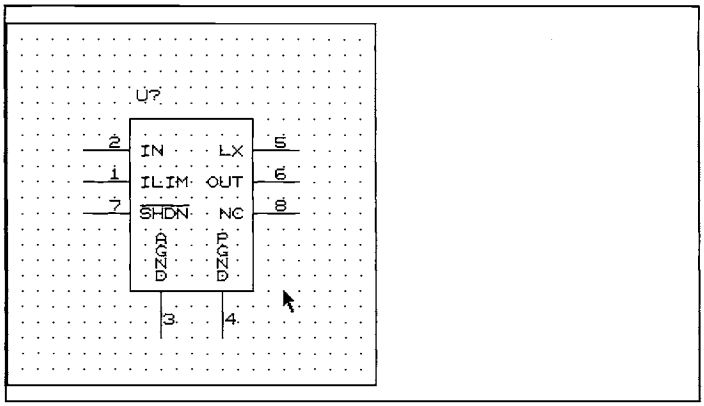


Figure 6-33 MAX877 with All Pins Added

General Guidelines for Part Layout

When creating a new part, spending some time up front to plan out the part usually pays off. Using the approximate body size and pin arrangement shown in a typical circuit application on the manufacturer's data sheet is a good starting point. Some data sheets only show the pins arranged in the same order as on the device package. This is usually not the optimum arrangement from a circuit standpoint. The layout of a part should follow similar signal flow guidelines as used for schematics. Signal flow guidelines along with a few suggestions on pin names are given below:

- Signals should flow from left to right.
- Bussed signals, such as multiple data, address, or device select signal lines should be grouped together and ordered according to the signal, that is D0, D1, D3..., regardless of the actual pin numbers.
- Power supply connections should be made at the top and bottom, with positive voltages at the top and negative voltages and ground at the bottom.
- Avoid invisible power pins except for simple TTL and CMOS logic devices used in single supply designs. Never create analog or power devices with invisible power pins.
- Use pin names as shown on the manufacturer's data sheet, unless an overwhelming reason exists to use a different name. Other technical documents, such as test and repair procedures, often reference the pin names

used on schematics. When these pin names do not match part data sheets, confusion and errors can result.

- Pins with inverted signals can be represented either by a dot shape on the pin or by an overbar above the pin name. Do not use both representations, as they cancel each other out. Note that OrCAD does not make any provision for a backslash at the end of pin names.

Editing the Reference Designator Prefix

The library editor uses "U" as the default reference designator prefix for new parts, so the prefix should already be correct for the MAX877. A list of industry standard reference designator prefixes appears in Table 1-1 in chapter 1.

For practice, let's run through the steps involved in editing the reference designator prefix. Click on the Reference command at the main menu. Then enter U for the reference designator prefix as shown in Figure 6-34.

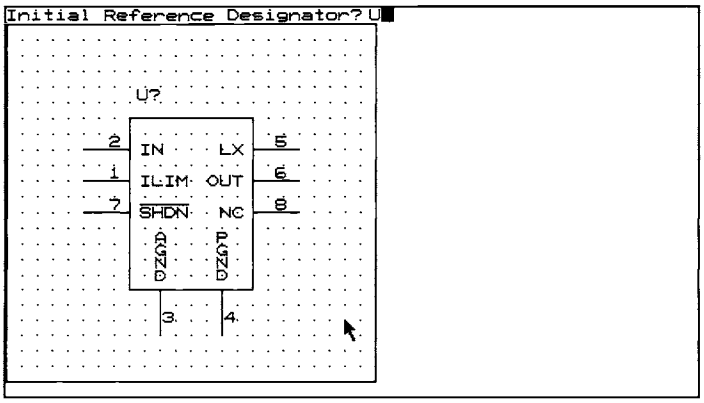


Figure 6-34 Editing the Reference Designator Prefix

The library editor adds a question mark (?) following the reference designator prefix you enter. The question mark is used as a placeholder for the reference designator suffix. The Annotate Schematic tool automatically replaces the question mark with a number denoting the instance of the part in the design. For multiple part packages, an alphabetic suffix starting with an "A" is also added to denote the instance of the part in the package.

Special Rule for Creating Invisible Reference Designators

If the part body was defined as having zero parts per package, the library editor does not automatically place a question mark character after the reference designator prefix. Unless the user manually enters the question mark, the

reference designator and part name become invisible in Draft. This special rule is used to create parts such as ground objects.

Adding the Part Name

Because the MAX877 is a new part, no name exists. Click on the Name command from the main menu and then click on the Add option. You are prompted for the name. Enter **MAX877**. The next screen prompts for a sheet path. Because the MAX877 is not a sheet path part, press the **<ENTER>** key. The MAX877 is now complete and should appear as shown in Figure 6-35.

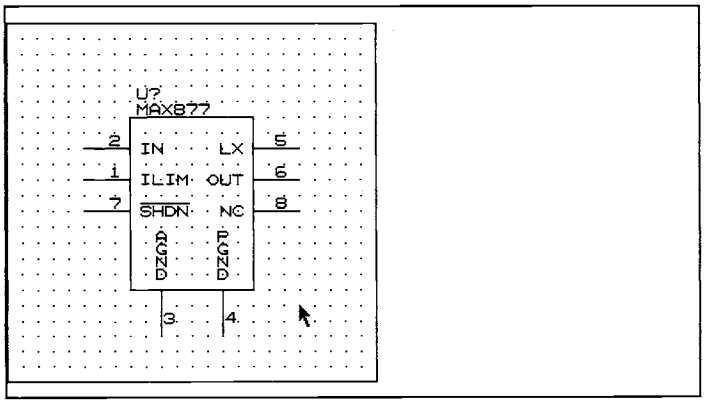


Figure 6-35 Finished MAX877 Part

Wrapping up the MAX877

The last step is to update the library. Recall that this requires two steps. First use the Library command and select the Update Current option to save the MAX877 to the library in RAM memory. Then use the Quit command and select the Update File option to save the library to disk.

You will require the MAX877 for the tutorial in the next chapter. Exit to DOS and backup the CUSTOM.LIB library file onto a floppy disk before proceeding.

Listing Library Parts

A list of available library parts, especially for the custom library, provides a useful source of information when starting a new design. You can generate the list by using the Library command and then selecting the List Directory option. The parts directory list can be displayed on the screen or sent to a file.

As part of the tutorial, use the Library command to list the parts in the custom library. First, display the directory list on the screen. A short submenu with options More and Quit appears at the top of the screen. A large library may require more than one screen to display the entire listing. Press **M** to scroll down one page and **Q** to quit. OrCAD ignores the mouse and keyboard cursor controls. You must press **M** to scroll down.

Next, send the output of the Library List Directory command to a file. Use the file name CUSTOM.TXT. Then exit to DOS and use the LIST.COM utility or your text editor to view the CUSTOM.TXT file. A section of the directory listing appears in Figure 6-36.

```
LIST       1      69      09-27-95 17:49 ♦ C:\ORCAD\TUTOR4\CUSTOM.TXT
Directory of C:\ORCADESP\SDT\LIBRARY\CUSTOM.LIB
+12V POWER SUPPLY                    +12V POWER SUPPLY1
+24V POWER SUPPLY                    0-35V DC POWER SUPPLY
14051                                14527
20RA10                               20X10
2764                                 27C64
4013                                 4025
4051                                 4066
4093                                 4527
4538                                 4584
4584 5V                              7406
7407                                 74HC4051
74HCT14                              74HCT4051
ADC0808                              ADC0809
ADC7802                              BCD EECO 3500 RIGHT ANGLE
BCD PUSHWHEEL SWITCH                 BCD PUSHWHEEL SWITCH CHERRY T59
BL100302 DPM                         BUZZER
CA3085A                              CIRCUIT BREAKER
CONNECTOR IBM PC XT                  CURRENT TRANSFORMER
DPST PUSHBUTTON                      DS1286
DS26LS31C                            DS26LS32AC
FUSE                                 GAL22V10B
GM ALPHA CONN                        GM NUMERIC CONN
GND POWER                            GND SIGNAL
H11F3                                HA-5640
HEADER 1                             HEADER 10
HEADER 12                            HEADER 14
HEADER 16                            HEADER 2
HEADER 26                            HEADER 3
HEADER 4                             HEADER 5
HEADER 6                             HEADER 8
HEADER 9                             HEADER15
ICL8211C                             ICL8211CPA
IGBT                                 L482
L484                                 L497
LED BICOLOR                          LM2903
LM2917                               LM317
LM317/ALT                            LM324
LM324A                               LM331
LM339                                LM555
LM555CN                              LM556
LM7805                               LM78L05
MAX505                               MAX506
MAX508                               MAX877
MC145026                             MC145028
MC34063AP1                           MC7906
METER MA                             MIC2951
MOSFET N                             NPN DAR
Command▶     *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 6-36 CUSTOM.LIB Parts Listing

You can print the list by sending the file to a printer. Use the DOS Copy command:

COPY CUSTOM.TXT LPT1: <ENTER>

OrCAD also provides a separate tool just for creating a library parts listing. The List Library tool appears on the main SDT screen. This tool duplicates the functionality of the Library List Directory command available in Edit Library.

Most users will find the List Library Directory command to be more flexible. List Library Directory displays a directory of parts on the screen, and it allows scrolling down the list one page at a time. The List Library tool does not allow any screen display. Output can only be sent to a file. The only apparent advantage of List Library is that the list generated by this tool includes a summation of the total number of parts in the library.

Archiving Library Parts in the Design

Most EDA systems use libraries to speed-up the design process by offering quick access to predefined parts. Once a design has been created, two possible approaches exist for storing information about the parts used in the design. The approach used in a particular system has a major impact on how the user must manage design and library files.

One approach is to directly include all part data in the design file. In this approach, data for a given part is only read from the library once, when the part is first placed into the design. The design file can continue to be used even if the original part libraries are no longer available. Because the design file includes all part data, the file is larger. Updates to the part libraries will not automatically be reflected in the design.

The second approach is to reference the part name in the design file, but not include part data. Actual part data is read from the part libraries whenever the design is loaded into the system. If the part libraries are no longer available or referenced parts are deleted or corrupted, the design cannot load correctly. The affected parts appear as blank areas and the system issues a warning message. On the other hand, advantages of this approach include smaller design files and automatic updates.

OrCAD uses the second approach but provides a simple method for managing the part data in designs. The Archive Parts In Schematic tool is used for this purpose. This tool scans the design and creates a special library to permanently archive all parts used in the design. The archive library can then be added to the configured libraries list for the design. Recall that this is part of the SDT configuration and can be customized for each design (see chapter 2, page 50). If the archive library is placed at the top of the list, Draft will search the archive library first. Changes to the other libraries will not affect the integrity of the design.

Recommended practice is to always create an archive library and add it to the configured library list when a design has been completed. If the design is later revised and new parts are added, make sure you create a revised archive library. Including an archive library with the design file is also the only feasible method

of sending OrCAD designs to other sites or companies that may have incompatible parts libraries.

Creating an Archive Library

To illustrate the use of the Archive Parts In Schematic tool, let's create an archive library for the TUTOR3 design. Recall that this is the three sheet hierarchical design completed in the previous tutorial.

Use the OrCAD Design Management tool to select the TUTOR3 design. Launch OrCAD SDT. At the main SDT screen, click on Archive Parts In Schematic and then click on Local Configuration as shown in Figure 6-37.

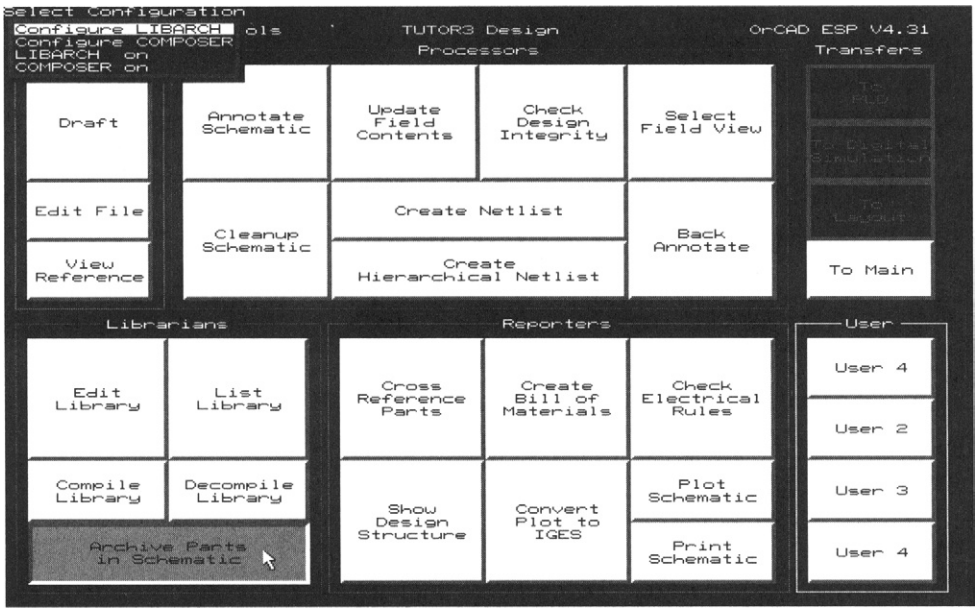


Figure 6-37 The Archive Parts in Schematic Configuration Menu

The Archive Parts In Schematic tool consists of two routines: LIBARCH (Library Archive) and COMPOSER (Compile Library). LIBARCH searches through the design and builds an archive library source file containing all the parts in the design. This library source file is an ASCII text file with OrCAD SDL (Symbol Description Language) format. SDL format is briefly discussed in the following section. COMPOSER then compiles the library source file into the standard library file format required by Draft and Edit Library. Once local configurations are properly set, the entire library archiving process is transparent to the user.

Click on Configure LIBARCH. Set the local configuration the same as shown in Figure 6-38.

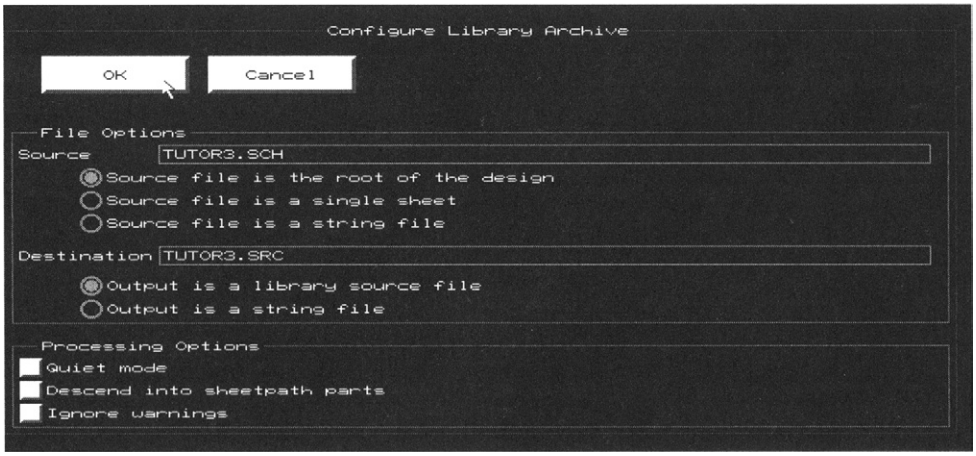


Figure 6-38 LIBARCH Configuration Options

The LIBARCH configuration screen has file and processing options similar to other OrCAD routines. The file source (do not confuse with library source file, which is a file format description) is normally the root design file. Library source information will be written to the destination. OrCAD uses the extension .SRC for library source files.

After you have verified the LIBARCH configuration options, click on OK to exit. Go back to the Archive Parts In Schematic screen shown in Figure 6-37 and click on Configure COMPOSER. Set the local configuration the same as shown in Figure 6-39.

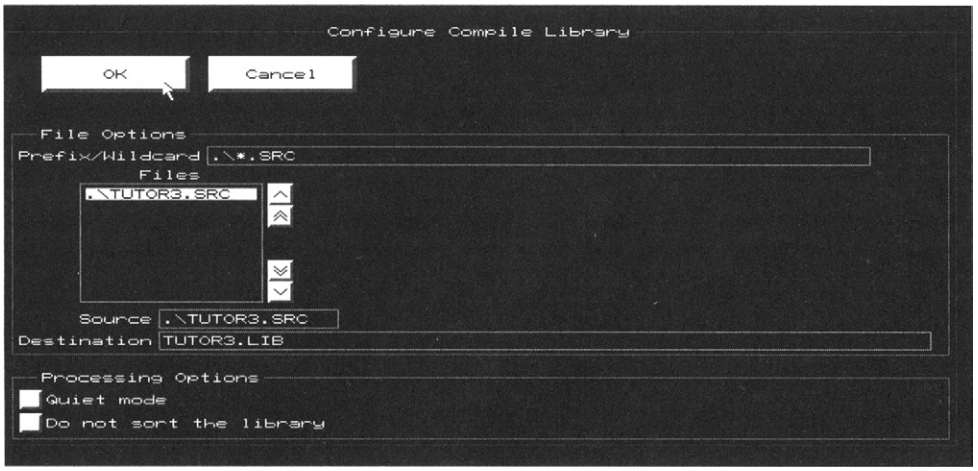


Figure 6-39 COMPOSER Configuration Options

The COMPOSER configuration screen has file and processing options similar to LIBARCH except that a scroll box is used for file source selection. The file source is the library source file created by LIBARCH. Enter the prefix and wildcard as shown in the figure, then click on the TUTOR3.SRC file in the scroll box. The archive library file created by COMPOSER is written to the destination file. This uses the same .LIB extension as other OrCAD library files.

COMPOSER normally sorts library files in alphanumeric part name order. The "Do not sort" option appears to have limited usefulness.

After you have verified the COMPOSER configuration options, click on OK to exit. Launch the tool and create the archive library by double clicking on Archive Parts In Schematic. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appears during execution, there is usually no reason to examine this file.

Next, exit back to DOS. Use the DOS directory command to examine the TUTOR3 design subdirectory. You should see a TUTOR3.SRC and TUTOR3.LIB file. You can delete TUTOR3.SRC since this intermediate file is of no further use. Always keep the archive library file in the design subdirectory. Backup the archive library file onto a floppy disk along with the other design files.

Configuring SDT for the Archive Library

The new archive library you have just created must be added to the configured library list in the SDT configuration for the TUTOR3 design. Once this last step is completed, the archive library will automatically be loaded with the design whenever Draft is launched.

Click on Draft and then click on Configure Schematic Tools. For details refer to back to chapter 2. Scroll down to the Library Options section of the configuration screen. Click on the Insert a Library button (Figure 6-40). Position the green bar (insertion pointer) to the top of the Configured Libraries scroll box. Then click on TUTOR3.LIB in the Available Libraries scroll box and click on Insert. TUTOR3.LIB should appear at the top of the configured libraries as shown in Figure 6-40. Scroll back to the top of the configuration screen and click on OK to exit.

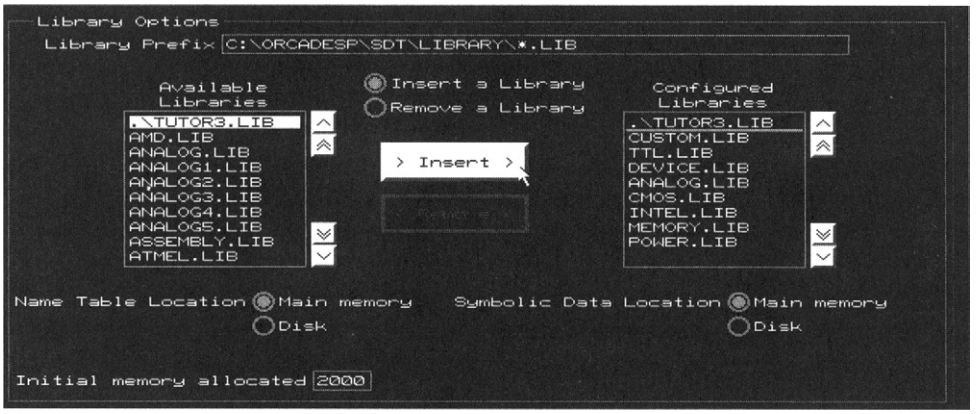


Figure 6-40 Adding the Archive Library to SDT

Additional OrCAD Library Tools

OrCAD provides two additional library tools: Compile Library and Decompile Library. As previously discussed, these two tools were intended to facilitate off-line parts library preparation using a text editor. Off-line preparation of EDA input files was once a common practice. EDA programs only ran on very expensive workstations. Using dumb terminals to prepare input files for part libraries and netlists was very cost effective. OrCAD and other vendors of early PC based EDA software sought to accommodate the established work flow procedures and included routines to facilitate off-line data preparation.

In the case of OrCAD, source files for part libraries could be written in OrCAD SDL language on any ASCII text editor. The Compile Library tool was then used to convert the source file to an actual OrCAD library file. The Decompile Library tool could convert an existing library back to ASCII for editing.

Today, no user could possibly find justification for using these tools. Besides being highly prone to errors, using off-line text entry to create a parts library would probably require an order of magnitude more time than using the Edit Library tool. Consequently, Compile Library, Decompile Library, and OrCAD SDL are not covered in this book.

Tips on Creating Parts and Library Management

You have now completed the fourth tutorial and learned the basics of the library editor. This chapter concludes with some tips on creating parts and library management. Much of the focus of this chapter has been on the custom library. Creating new parts in the custom library is an integral step in the design process. Your custom library also represents a considerable investment in time and money. The following tips are intended to summarize the main points discussed in this chapter and to provide some guidelines for saving time and maximizing your return on investment:

- **Graphic versus block body**. Only use graphic bodies for discrete components and simple logic gates that require a specialized symbol. Use block bodies for all complex parts.
- **Invisible power pins**. Today's trend towards multiple voltage levels in logic circuitry renders invisible power pins more of a nuisance than a feature. Do not hesitate to edit existing logic parts to make power pins visible, if this will improve the clarity of the schematic.
- **Clearing the screen**. Once Edit Library has been launched and the first part created or edited, there is no convenient way to clear the screen, other than using the Quit Initialize command.
- **Selective pin display with multiple parts per package**. When creating parts with multiple parts per package, pins on a given part can be made invisible by assigning them pin number zero.
- **Converted forms**. Only parts with graphic bodies can have a converted form. The converted form is just an alternate graphic body. There is no requirement that the converted form represent a DeMorgan logic equivalent.
- **OrCAD SDT 386+ compatibility**. Library files from older OrCAD versions are not compatible and must be converted using the CONVERT program.

Refer to the OrCAD manual for details. CONVERT removes raster data (bitmaps) once used in the older libraries. OrCAD SDT 386+ only uses vector part definitions.

- **OrCAD supplied libraries**. Never modify OrCAD supplied libraries. When you install updated libraries, your changes will be lost. If you need to modify an OrCAD library part, export the part to a file, import the file to your custom library, and then modify the part.
- **Custom library**. Avoid creating multiple custom libraries. They will become difficult to manage. Even a busy design department is not likely to wind up with more than a few hundred parts in the custom library since the OrCAD supplied libraries are so extensive. Avoid creating a new part if the only change is to the part name and a library part already exists with the same pinout. This frequently occurs with logic devices as new logic families evolve. Many of the new 3V and 3.3V logic devices maintain the same pinout as standard TTL.
- **Archive library**. Always create an archive library when a design has been completed. The OrCAD supplied libraries and the custom library may change. An archive library in the design subdirectory protects the integrity of the design.

7

Tutorial 5 - Advanced Features

The fifth tutorial is the last of a series of exercises designed to get you off to a fast start using OrCAD SDT to capture schematics. In this tutorial, you will explore the remaining OrCAD SDT tools and examine several Draft commands in more detail. The tutorial exercise is divided into three sessions.

In the first session, you will draw a hierarchical schematic for a memory expansion board that requires isolated power in one section. The hierarchical structure is preferred for large and complex designs. OrCAD also provides for conventional multiple sheet schematics, which are referred to as a "flat" design structure. In the second session, you will redraw the schematic using a flat design structure. The third session includes a detailed discussion of key fields and part fields and introduces advanced bill of materials and netlist techniques. The third session also covers OrCAD tools used for managing part fields.

Starting the First Session

Use figures 7-1 through 7-3 as the model for the first session. Start by creating a new design called TUTOR5, based on the standard template. Use the OrCAD Design Management tools as for the previous tutorials. Then select TUTOR5 and launch OrCAD SDT and Draft.

Using a Hierarchical Structure for a Small Design

In previous tutorial exercises, hierarchical schematics were drawn in a formal fashion, with the first sheet representing the hierarchical structure. Only sheet objects (representing major circuit blocks drawn on subsequent sheets) and the interconnections between these sheet objects appeared on the first sheet.

For small designs, such as the memory expansion board used as the model for this session, a less formal approach improves readability. The main circuit block appears on the first sheet (Figure 7-1). Additional circuit blocks are shown as sheet objects and then detailed on subsequent sheets (Figures 7-2 and 7-3). Using the more formal approach, an additional sheet would have been required just to show the hierarchical structure.

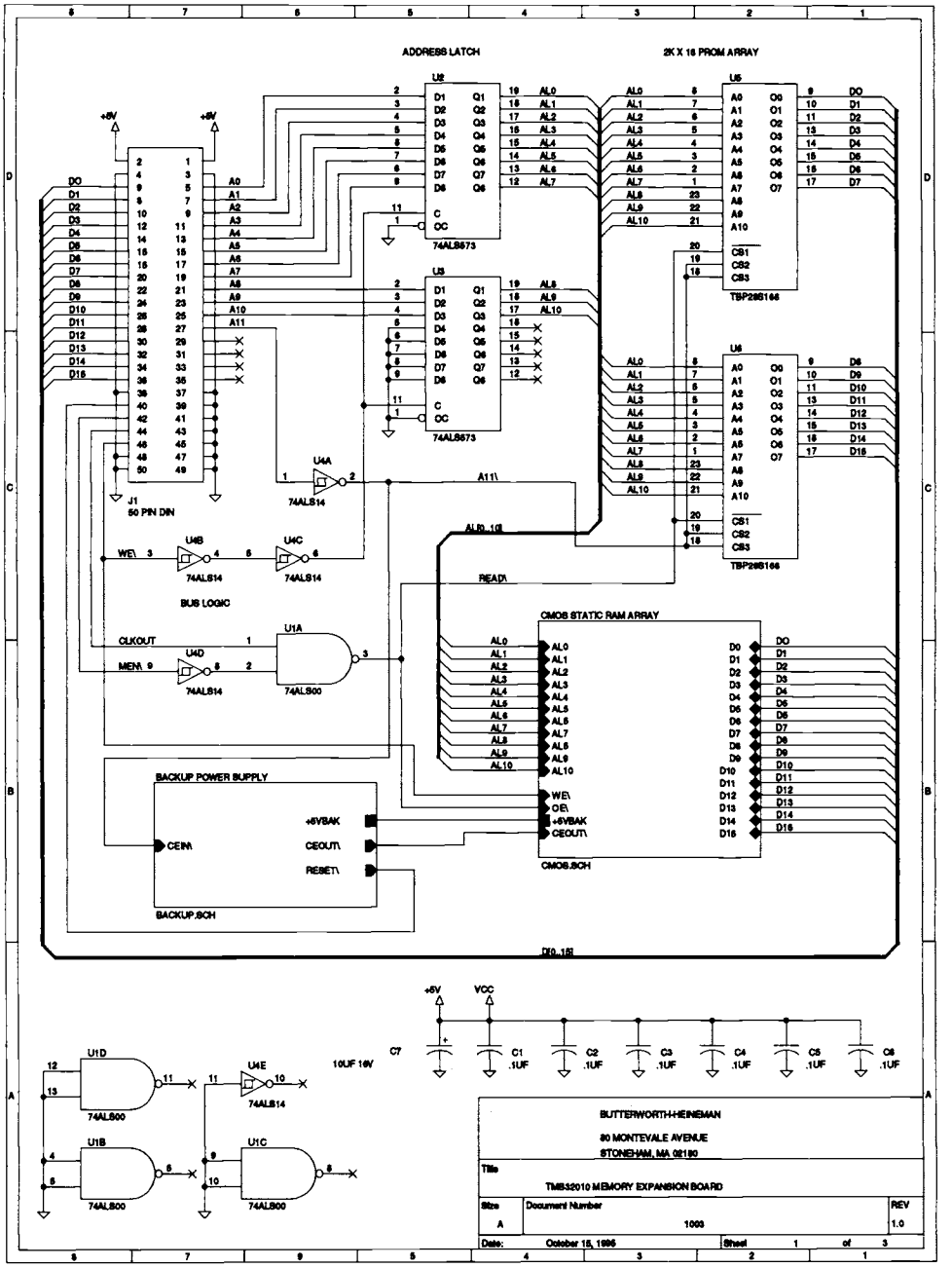


Figure 7-1 Fifth Tutorial Sheet One

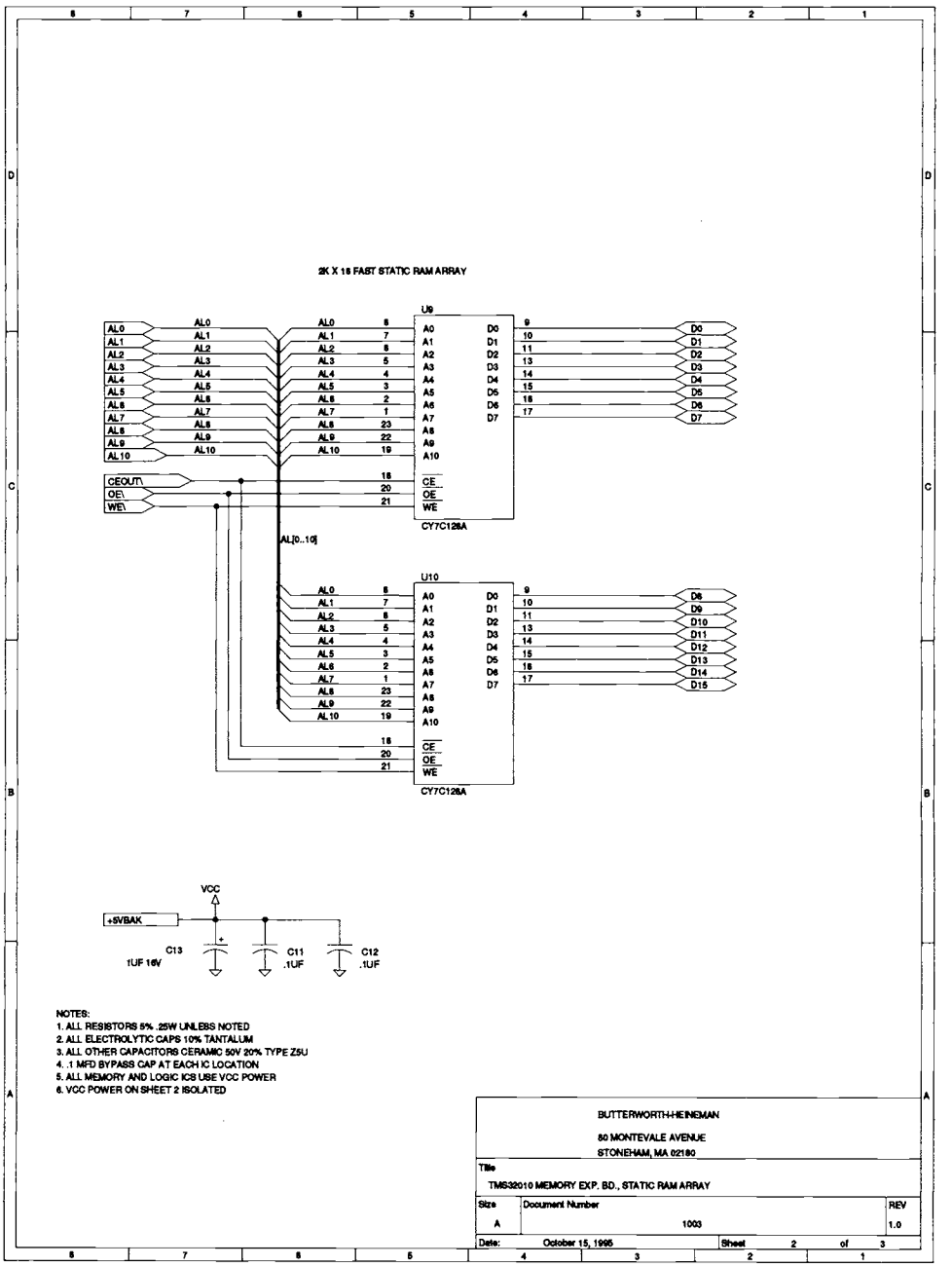


Figure 7-2 Fifth Tutorial Sheet Two

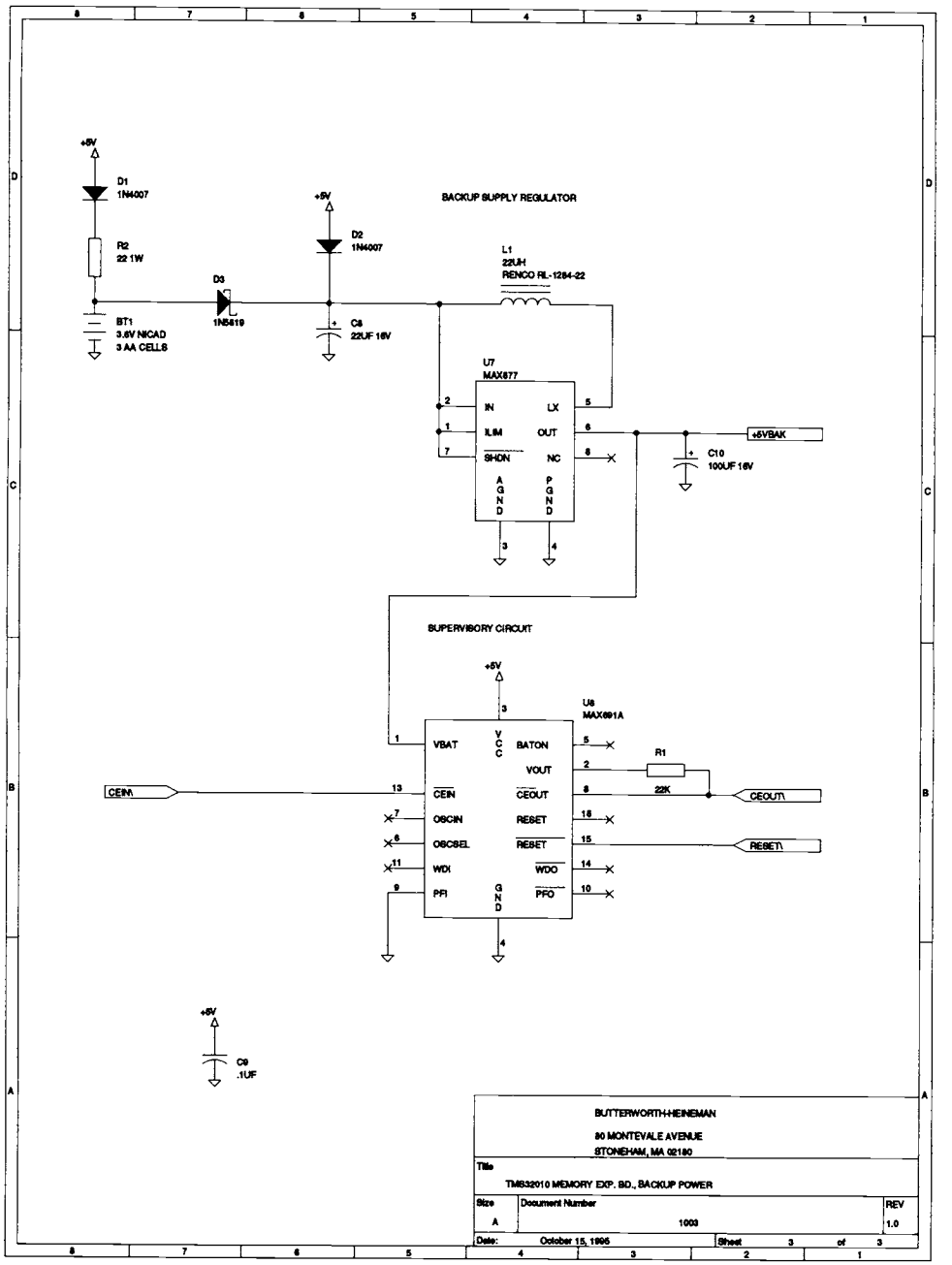


Figure 7-3 Fifth Tutorial Sheet Three

As a general rule of thumb, try to keep the number of sheets to a minimum. For designs requiring more than four sheets, reserve the first sheet for sheet objects showing the structure of the hierarchy. For less complex designs, place the main circuit block on the first sheet along with sheet objects for the remaining sheets.

Drafting the Schematic

By now you have mastered the basic skills required to complete the schematic shown in figures 7-1 to 7-3. Since this tutorial introduces some new techniques, just follow along for now. Listed below are hints on parts to help you get started:

- **50 Pin Din connector**: CON50A from DEVICE.LIB.
- **74ALS series logic ICs**: these are in TTL.LIB. The 74ALS14 is shown with a smaller symbol (for convenience and fit) than the part in the TTL library. Use the 74HCT14 part from the CUSTOM.LIB and edit the part description.
- **TBP28S166 TTL PROM**: 28S166 from MEMORY.LIB.
- **CY7C128A CMOS Static RAM**: 7C128 from MEMORY.LIB.
- **3.6V Nicad battery**: BATTERY from DEVICE.LIB.
- **1N5819 Schottky diode**: DIODE SCHOTTKY from DEVICE.LIB.
- **22UH Inductor**: INDUCTOR IRON from DEVICE.LIB.
- **MAX877 and MAX691A ICs**: these are in CUSTOM.LIB. The MAX877 IC was created in the previous tutorial.

For now, complete the first sheet of the schematic except for the bypass capacitors and spare gates at the very bottom. Leave off the signal and bus labels. In the next two sections, you will learn how the Block command can be used to copy sections of a schematic thus speeding up the drafting process.

Overview of the Block Command

So far you have learned to use the Block Drag and Block Move commands. Let's look at the other Block commands. Click on Block from the main menu. The Block menu appears as shown in Figure 7-4. The following options are available:

Move Used to move objects crossing a selection box from one location to another on the same sheet. Any wire or bus segments crossing the selection box are moved in their entirety, including the endpoints. Connections at wire and bus endpoints are broken.

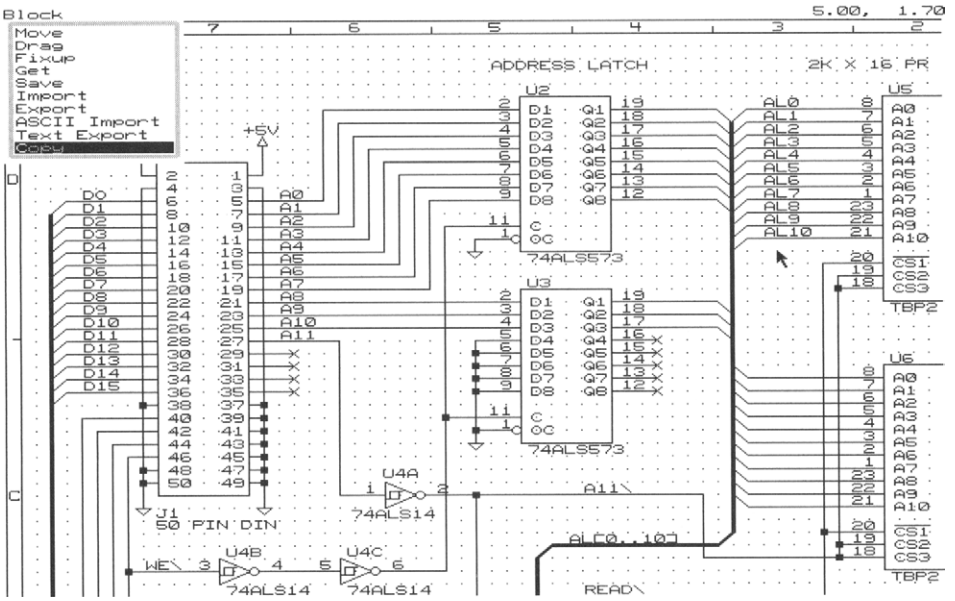


Figure 7-4 Block Command Options

Block command options continued:

Drag Used to drag objects crossing a selection box from one location to another on the same sheet. Any wire or bus segments crossing the selection box are "rubberbanded." The wire or bus segments are stretched as required. Endpoints outside the selection box remain fixed and connections are not broken.

Fixup Specialized Block command used to "fix up" non-orthogonal wires and buses. Adds new segments to make the selected wires or buses orthogonal. This command has limited usefulness, since most work in Draft is done with orthogonal mode ON. Fixup seems cumbersome to use. Many users find that deleting and redrawing is more convenient when a mistake does require fixing up.

Get Retrieves objects previously saved to the clipboard (OrCAD uses the term buffer) via Block Save. Displays a box containing the objects. Multiple copies can be placed at desired locations on the sheet by clicking the left mouse button. Note that the clipboard is also used by

the Block Drag and Move commands. Anything saved in the clipboard is lost when these commands are used. A further limitation is that the clipboard is flushed when navigating between sheets. In most cases, using the Block Copy command (see below) is more convenient.

Save Objects crossing a selection box are saved to the clipboard. These objects can then be retrieved using the Block Get command. Block Save and Block Get are analogous to Windows' Copy and Paste. A limitation is that the clipboard is flushed and any saved objects are lost when you use Block Drag or Move commands or when navigating between sheets. In most cases, using the Block Copy command is more convenient.

Import Retrieves objects previously saved to a file via Block Export. Prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Displays a box containing the objects retrieved from the file. Multiple copies can be placed at desired locations on the sheet by clicking the left mouse button.

Export Objects crossing a selection box are saved to a specified file. These objects can then be retrieved using the Block Import command. Block Export and Import are useful for moving objects between sheets or designs. After objects are selected, Block Export prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Suggested practice is to use the .BLK extension for exported block files.

ASCII Import Retrieves ASCII text from a file. Prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Places text immediately to the right of the cursor location. The ASCII text file can be created with any text editor or via the Block Text Export command and can contain multiple lines. When using a text editor, make sure that the imported text is free of any special formatting characters. The Block ASCII Import command is commonly used to import standardized notes.

Text Export Text objects crossing a selection box are saved to a specified file as ASCII text. The ASCII text can then be edited with a text editor or retrieved using the Block ASCII Import command. After text objects are selected, Block Text Export prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Suggested practice is to use the .TXT extension for exported ASCII text files. Note that only text objects are selected. Any characters associated with labels, part descriptions and reference designators, module ports, sheets, title block, or other objects are ignored.

Copy Objects crossing a selection box can be copied to a new location. Multiple copies can be placed at desired locations on the sheet by repeatedly clicking the left mouse button. The Block Copy command combines the action of Block Save and Block Get into one command.

Using Selection Boxes

When using the Block Move, Drag, Save, Export, Text Export, or Copy commands, objects are selected by defining a "box." This is referred to as a selection box. The Delete Block command also uses a selection box. Selected objects are referred to as belonging to a selection set.

In OrCAD, all objects that touch or cross a selection box become part of the selection set. Other EDA and CAD programs, notably AutoCAD, offer more flexibility as far as object selection, such as tagging individual objects one by one, selecting only those objects that are entirely enclosed within a selection box, or removing objects from a selection set. The fact that OrCAD always selects all objects crossing a selection box reduces the flexibility of the Block commands. A creative workaround to this situation will be explained in the next section.

A selection box is defined by moving the cursor and clicking the left mouse button at a beginning point and again at an ending point. OrCAD is very flexible in this regard - the two points can be on a line or they can even be the same point. Any objects that cross or touch the box, line, or point will become part of the selection set.

Creative Use of the Block Copy Command

Microprocessor and microcontroller related circuits tend to have many repetitive elements, such as address and data buses and memory arrays. Creative use of the Block Copy command can speed the schematic drafting process.

Assuming that you have completed most of the first sheet up to the point shown on Figure 7-4, let's examine how the Block Copy command can be used to copy the address labels AL0-AL10 from U5 to U6. First, use the Block Copy command to copy the address labels to a temporary location. Define a selection box around the labels as shown in Figure 7-5. Then place the selected objects in a temporary location near the bottom left corner of the sheet as shown in Figure 7-6. Figure 7-7 shows the result of the Block Copy command. The address line wires also crossed the selection box and became part of the selection set that was copied to the new location. This is unavoidable since OrCAD provides no means of limiting selection sets. Any objects crossing the selection box become part of the selection set.

The next step is to remove the extraneous wires. Select the Delete Block command as shown in Figure 7-7. Then define a selection box as shown in Figure 7-8. The result is shown in Figure 7-9. Now the address labels are all that remains. The last step is to use the Block Copy command to copy the labels to the desired location at U6, as shown in Figure 7-10. The final result is shown in Figure 7-11.

This example illustrates how to eliminate unwanted objects prior to copying, which is an important technique to master. The general procedure is to copy the objects to a temporary area, use the Delete command to remove unwanted objects, and then copy the remaining objects to the final location. Note that the Delete Object command can be used to delete specific objects. If more than one object touches the cursor location, Delete Object allows choosing the particular class of object to delete.

The above example was intended to illustrate a general technique. An alternate approach would have been to draw all the circuitry around U5 that is repeated at U6 and then use the Block Copy command.

Copying objects is not always the most efficient approach. If the selection set will include many extraneous objects that must first be removed, copying may be less efficient than just redrawing. If many labels or module port signal names must be edited afterwards, copying becomes an inefficient process.

Planning ahead, even to the extent of first drawing a very rough sketch by hand, can prove very helpful. If you have some idea of what the finished schematic will

look like, you can intelligently use the Block Copy command to save time and effort.

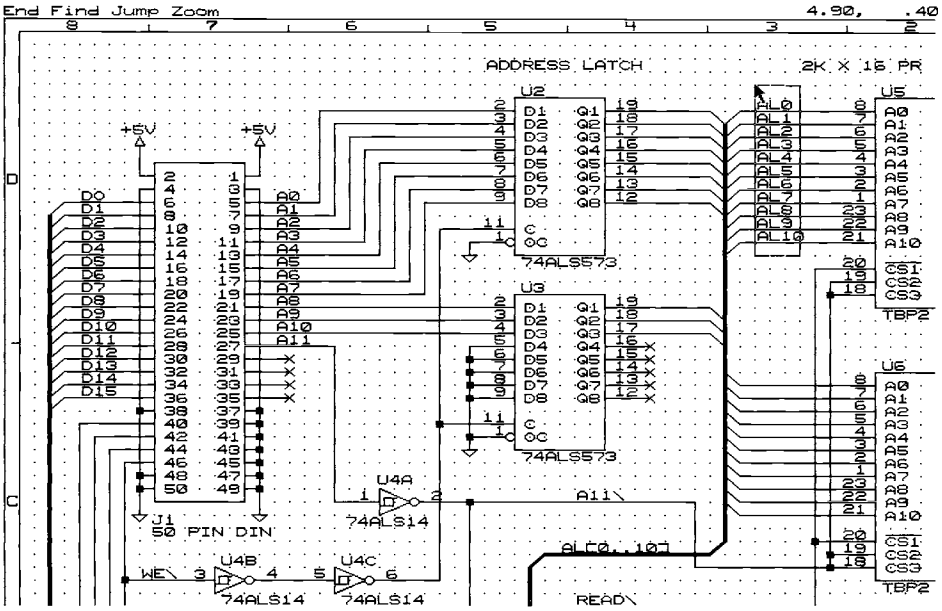


Figure 7-5 Selecting Address Labels to Copy

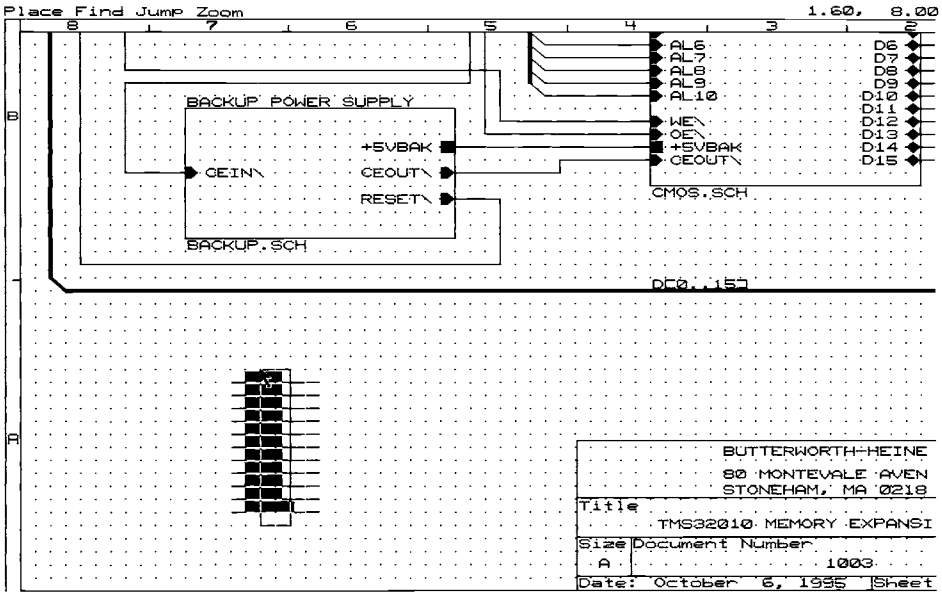


Figure 7-6 Copying the Labels to a New Location

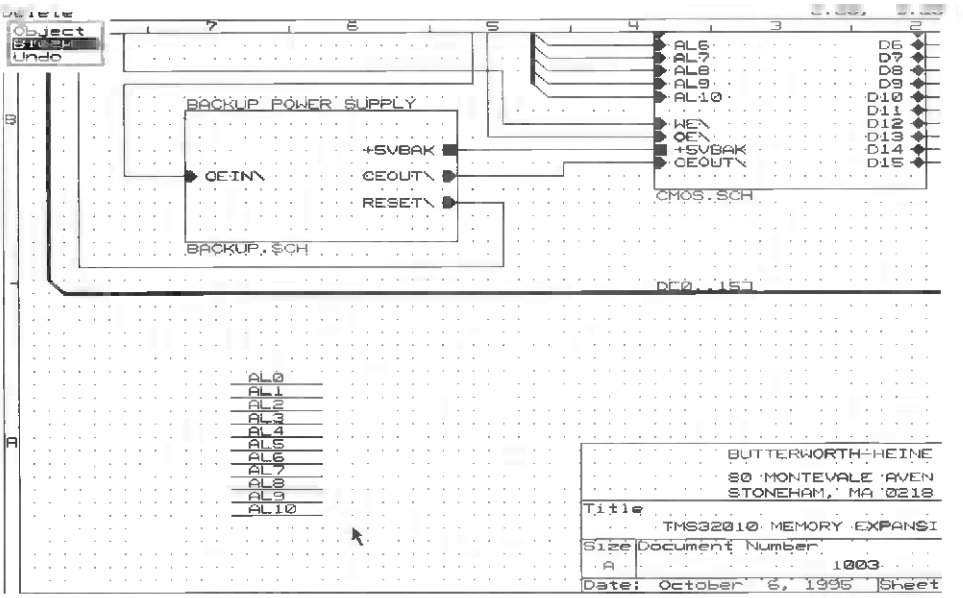


Figure 7-7 Selecting the Delete Block Command

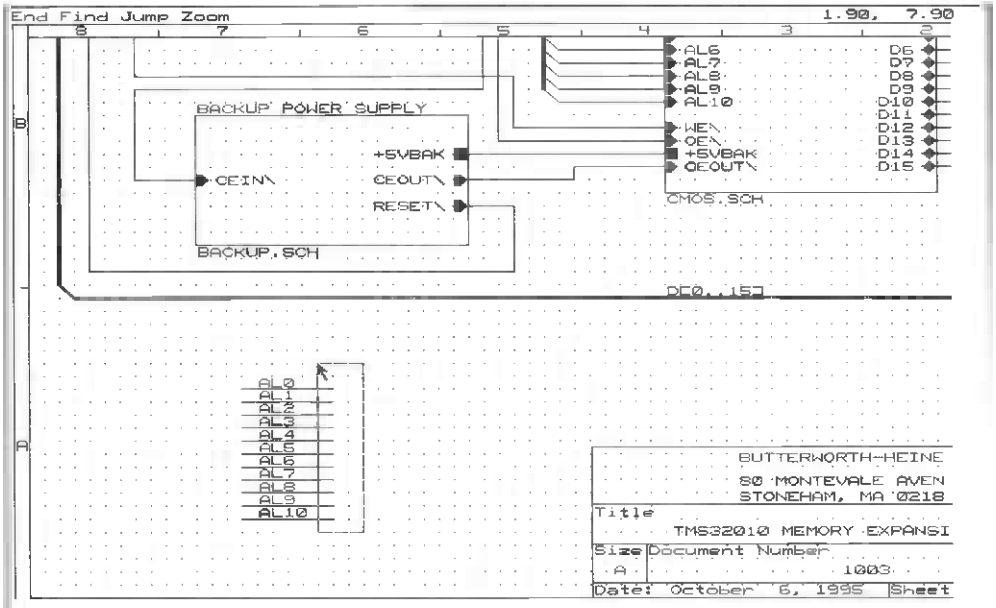


Figure 7-8 Deleting the Extraneous Wires

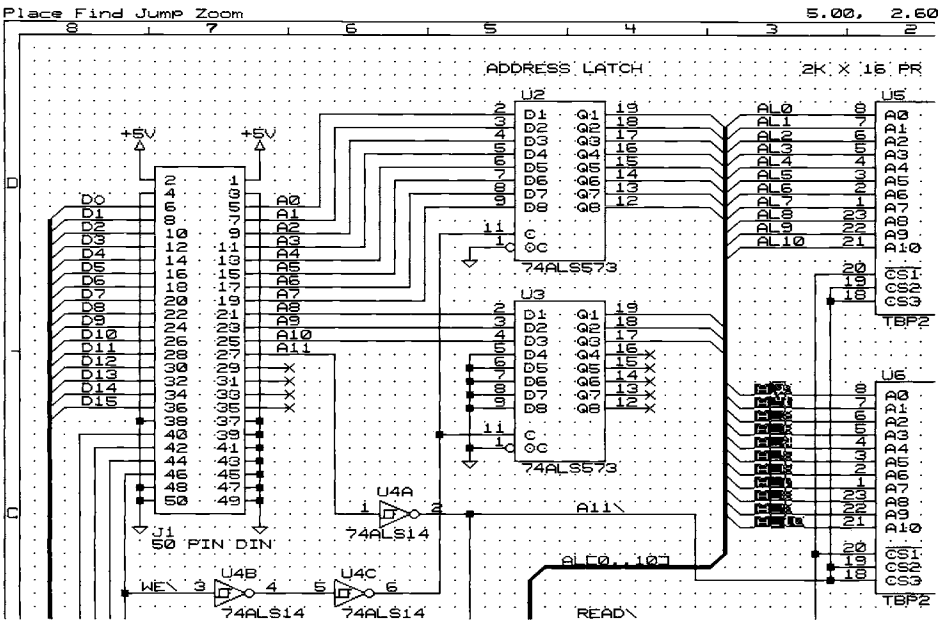


Figure 7-9 Address Labels after Deletion of Wires

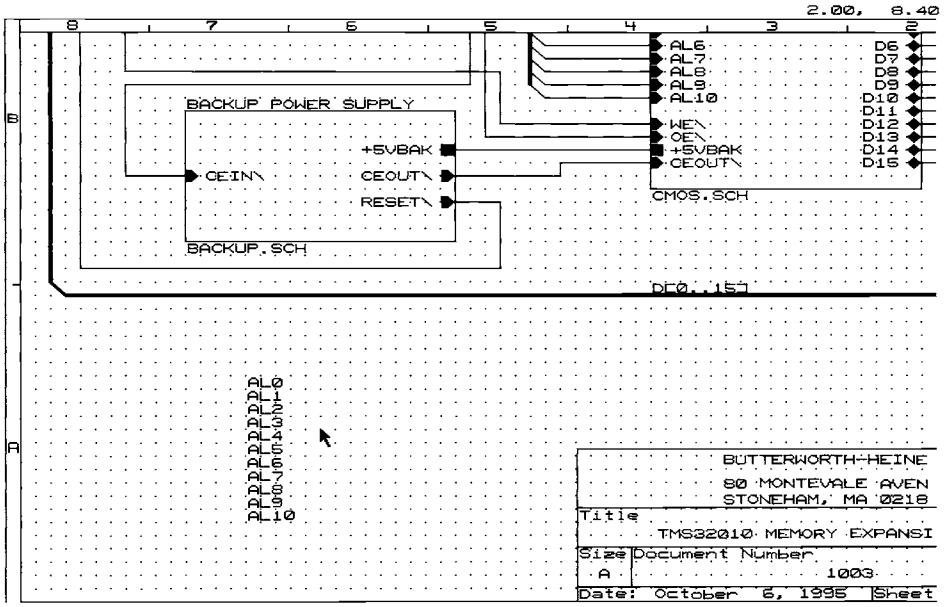


Figure 7-10 Copying the Labels to the New Location

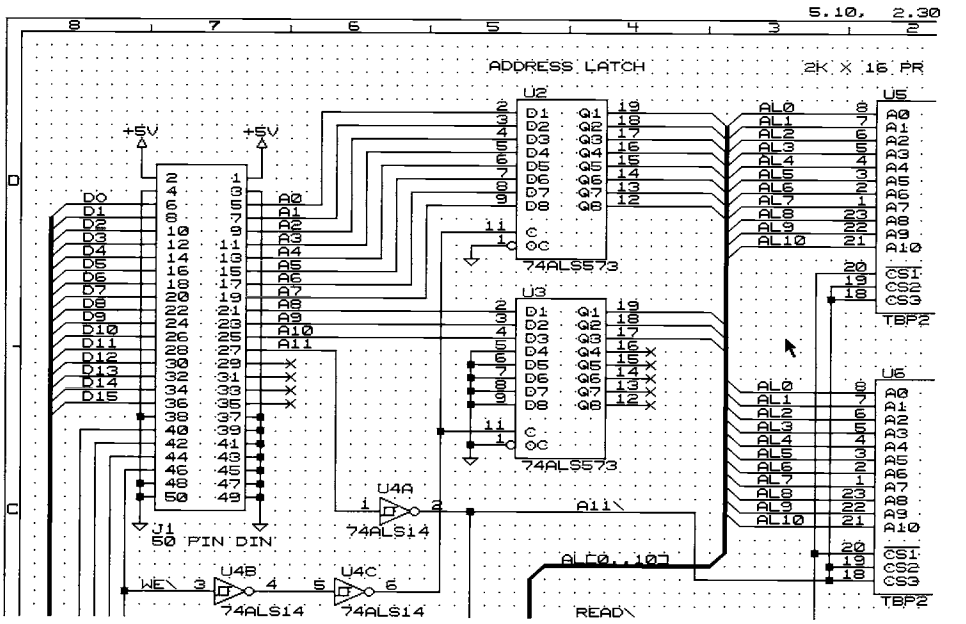


Figure 7-11 Address Labels in Final Position

Using the Block Save and Block Get Commands

Once the first sheet has been completed, you can use the Block Save and Block Get commands to copy the section of the address bus going to U5 and U6 and then use this section on sheet two. Define the selection box so that it includes the bus entries, labels, and wires at U5.

Once you start working on sheet two, you can draw U9 and the data wires. Then copy and edit these objects for use as U10.

Using the Block ASCII Import and Text Export Commands

Sheet two provides an opportunity to try using the Block ASCII Import and Text Export commands. Most companies have standard notes that appear on all schematics, often referred to as "boilerplate." A given schematic will usually require at least some minor changes or additions to the standard notes.

Creating extensive text notes in OrCAD is a somewhat cumbersome process. The text does not appear on the screen until an entire line has been typed in. Most users find that judging the precise length of a line is difficult. Common features found in even the most rudimentary text editor or word processor, such as word wrap at the end of lines, automatic paragraph formatting, and spell checking are

not available in OrCAD. Using a text editor to create and edit standard notes and then importing these notes into OrCAD can save a considerable amount of time and frustration.

As part of the tutorial exercise, exit to DOS and use your text editor to prepare the notes shown on sheet two (Figure 7-2). Enter the notes exactly as shown. Use your text editor in nondocument mode. There should not be any formatting characters except carriage return and linefeed at the end of each line. Print the notes to an ASCII text file. Use a file name such as NOTES.TXT. Copy the ASCII text file to your TUTOR5 design subdirectory. Then go back to Draft, enter sheet two, and use the Block ASCII Import command to import and place the text notes.

Next, try going in the opposite direction. Use the Block Text Export command to export the text notes to another ASCII text file. Use a different filename, such as NOTES1.TXT. Then exit back to DOS and try loading the new ASCII text file into your text editor.

If an existing schematic requires extensive editing of text notes, exporting the notes to an ASCII file, using a text editor, and then importing the edited file back into OrCAD is usually the most efficient approach.

Using the Text Editor Supplied with OrCAD

Up to this point, the assumption has been made that most readers have some level of DOS and PC experience and that they already have their own favorite text editor or word processor. In the context of this book, the terms text editor and word processor are interchangeable. OrCAD includes a text editing tool that can be launched from the main SDT screen.

OrCAD supplies the Stony Brook M2EDIT text editor. The author's experience is that most OrCAD users prefer their own familiar text editor and rarely use Stony Brook. One of the reasons is that the Stony Brook editor requires memorizing and entering keyboard commands for other than the most rudimentary operations. Needless to say, that is somewhat of an anachronism in today's software environment. Another reason is that OrCAD can be configured to directly access up to four user programs via buttons on the main SDT screen. Most users configure buttons for their favorite text editor and frequently used DOS utilities, such as LIST.COM. You will learn how to configure user buttons in the next tutorial.

Let's take a quick look at the Stony Brook M2EDIT text editor. At the main SDT screen, double click on Edit File to launch the editor.

Unlike other OrCAD tools, no local configuration options exist. The file selection screen shown in Figure 7-12 appears. OrCAD uses *.* as the wildcard entry box default. This causes all files in the current design subdirectory to be displayed in the file selection scroll box. Click on the NOTES1.TXT file you created in the last section using the Block Text Export command. The file name appears in the File To Edit box. Then click on OK to enter the text editor. The edit screen appears as shown in Figure 7-13.

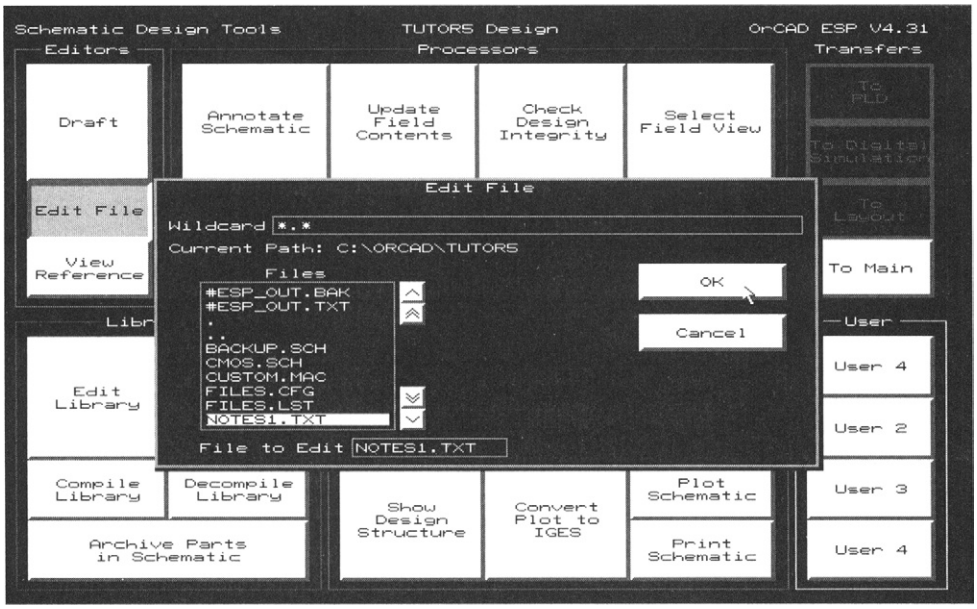


Figure 7-12 Launching Edit File (Stony Brook M2EDIT)

The Stony Brook text editor provides only limited mouse support. You can use the mouse to move the cursor. You can also use the mouse to move a scroll bar on the right side of the screen, however, the response is jumpy on fast 486 and Pentium class PCs. The arrow keys on the keyboard can be used for cursor control and scrolling, but the Page Down key is not supported.

A main menu appears at the top of the screen as shown in Figure 7-13. To select a command, position the mouse pointer on the appropriate command and click the left mouse button once. Note that the right mouse button, which works the same as <ESC> (escape key) in OrCAD, brings up on-line help in the Stony Brook editor. To escape from a menu selection or command, you must press <ESC> on the keyboard.

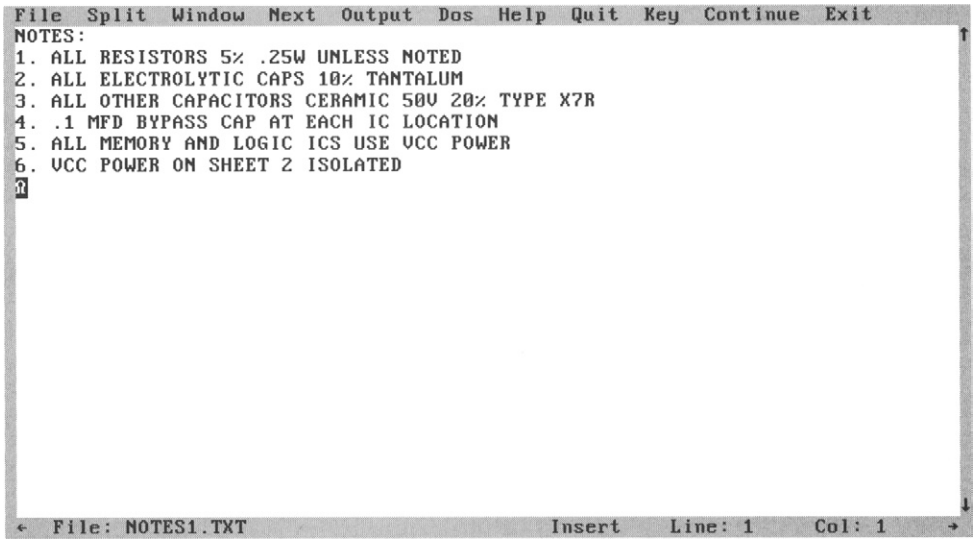


Figure 7-13 NOTES1.TXT File Loaded into M2EDIT

Click on the Help and Key menus to learn more about the Stony Brook editor and the required keyboard sequences for the various commands. To return to OrCAD, without saving any changes, click on Quit. Click on Exit to save any changed files and return to OrCAD.

The Stony Brook editor provides no direct means for printing. However, an easy workaround is to use the DOS command and then copy the desired file to the printer. For example, to print the NOTES1.TXT file, click the mouse on the DOS command and then enter:

COPY NOTES1.TXT LPT1:<ENTER>

Note that the editor loads the file into memory and then manipulates the data stored in memory. If any changes have been made, you must update the file by using the Output command before the changes can be printed.

Viewing OrCAD Reference Materials

OrCAD also uses the Stony Brook M2EDIT text editor to view on-line reference material. The reference material consists of various files written to the C:\ORCADESP\SDT subdirectory during the installation process. Some of these files contain useful information and updates not included in the OrCAD manuals.

While this is somewhat removed from the main thread of the tutorial, let's take a quick look at the View Reference tool. At the main SDT screen, double click on View Reference. As with the Edit File tool, no local configuration options exist. The file selection screen shown in Figure 7-14 appears. This screen is identical to the Edit File screen shown in Figure 7-12, except that the current path now points to the subdirectory where OrCAD loads reference files during installation. A list of available reference materials appears in the scroll box.

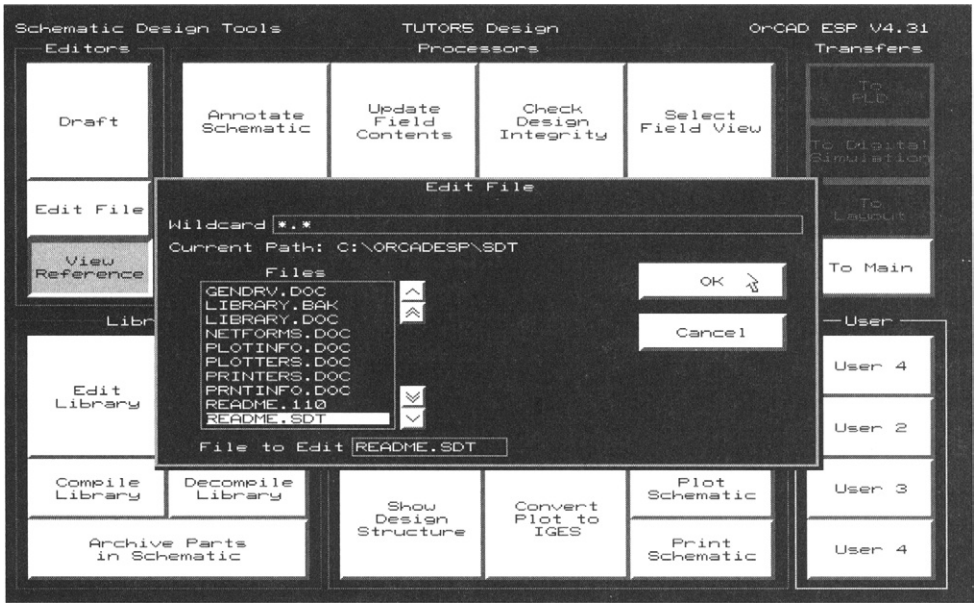


Figure 7-14 Launching the View Reference Tool

Click on the README.SDT file and then click on OK. The file contents appear as shown in Figure 7-15. You may want to spend 10-15 minutes familiarizing yourself with the various reference files. You can also use the DOS technique shown in the previous section for printing file contents.

Use of the Stony Brook M2EDIT text editor to view reference materials leaves much to be desired and is not a suggested practice. In the next session, you will learn how to install a user button that quickly launches the LIST.COM utility. LIST.COM provides efficient 43 line VGA text display, normal scrolling, and direct printing without the hassle of first issuing a DOS command.

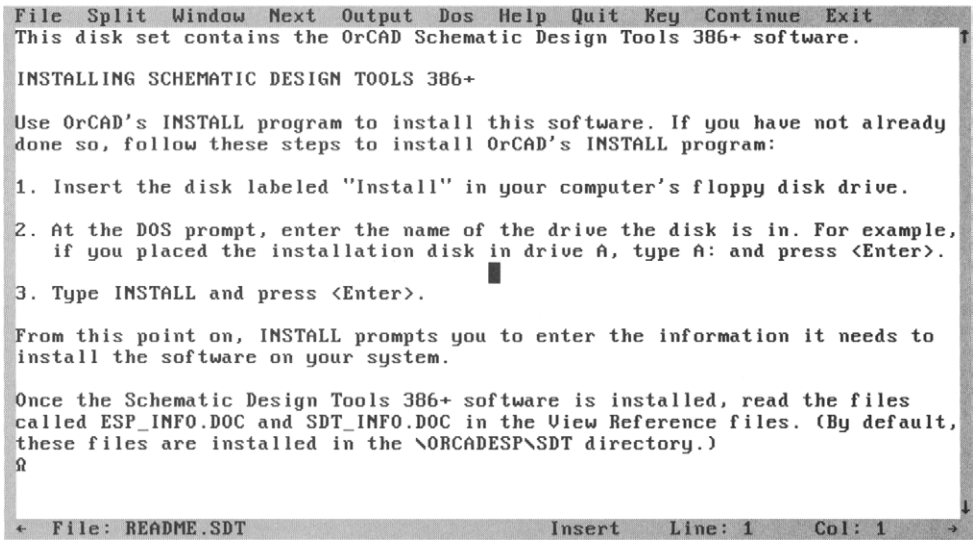


Figure 7-15 Displaying README.SDT Reference Material

Isolated Power Supplies and Invisible Power Pins

One of the main points of this tutorial is to learn how to create a design that isolates power to one section of the circuitry. The requirement for isolated power most commonly occurs with CMOS memory arrays that are supplied with backup power from a battery. Similar situations occur in designs using a combination of standard 5 volt and low voltage (2.7-3.3 volt) logic. Most OrCAD supplied library parts use invisible power pins. OrCAD automatically connects all invisible power pins to predefined power and ground planes, such as VCC and GND. In the context of this tutorial, the terms power supply and power plane have identical meaning. Engineers think in terms of power supplies; PC designers think in terms of power planes.

Fortunately, OrCAD provides a simple technique that allows isolating power to a particular circuit block. The only significant limitation is that the circuit block with isolated power must be placed on a separate sheet. In the tutorial, sheet two (Figure 7-2) uses isolated +5VBAK power from a battery backup supply on sheet three (Figure 7-3). The isolated +5VBAK power is brought onto the second sheet via an unspecified type module port. This module port is tied to a VCC power object.

OrCAD recognizes an unspecified type module port connected to a power object as a special instruction to create an isolated power plane on the particular sheet. All other power objects and invisible power pins with the same name on the

particular sheet will become part of the isolated power plane. Power objects and invisible power pins on other sheets are not affected.

The same technique can be used to create multiple isolated power planes on a given sheet. Multiple sheets can have isolated power planes, by repeating the technique on every affected sheet.

The same isolation technique can also be used with ground objects and invisible ground pins. Note that the three OrCAD grounds available (signal, power, and earth) are **not** isolated from one another since they all use the same pin name GND. If multiple isolated grounds are required on a given sheet, the simplest technique is to just use a power object with downward orientation and edit the name to something appropriate for a ground.

If logic circuitry, requiring multiple or isolated power supplies, must be shown on a single sheet, one must create custom parts with visible power pins. The power pins can then be individually connected to power objects named after the appropriate supply voltages.

One caveat is that the special technique OrCAD uses to create isolated power supplies is by no means a universally accepted standard. At the very least, you must include an explanatory text note on the schematic. Even with an appropriate note, the special OrCAD isolation technique may cause confusion. Ask yourself whether or not all those who will read the schematic during the product's life cycle are likely to fully understand what you had in mind. If you have any doubts, spend the extra time to create new parts with visible power pins.

Completing the Schematic

Complete all three schematic sheets. When you are done, run the Annotate Schematic tool to annotate any remaining reference designators. Make sure you have added the additional part descriptions for BT1 and L1 in the 1st part field via the Edit Command. Run Print Schematic to generate preliminary hardcopy of the schematic. Backup your design subdirectory to floppy disk before proceeding.

Running the Check Design Integrity Tool

The Check Design Integrity tool combines three individual tools that you have already used in previous tutorials: Cleanup Schematic, Cross Reference Parts, and Check Electrical Rules. Figure 7-16 shows the process and information flow. The rationale for the Check Design Integrity tool is convenience. Once you have learned to use Check Design Integrity, no reason exists to go back to running the individual tools.

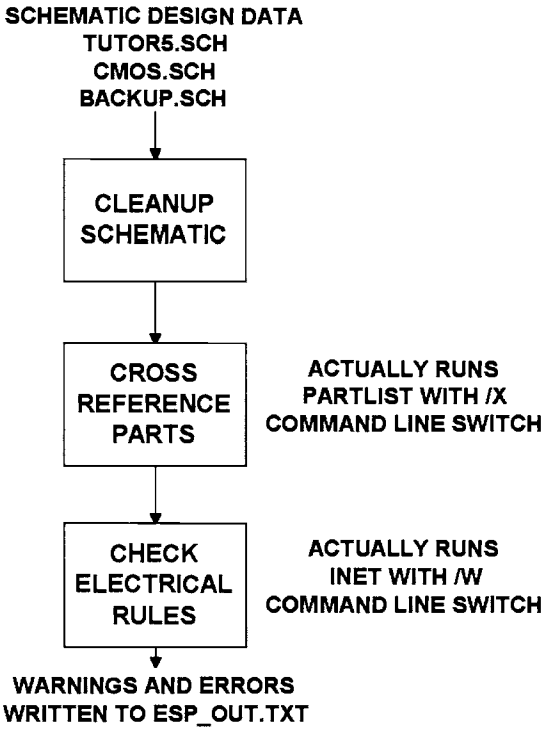


Figure 7-16 The Check Design Integrity Process

Some minor differences exist between running Check Design Integrity and the individual tools. Check Design Integrity has three local configurations, one for each of the three tools. The configuration screens are similar to those you have previously encountered for the individual tools, but they are independent configurations and must be properly set up. Check Design Integrity runs each tool in sequence, a process that programmers refer to as "chaining." This term also appears on some of the OrCAD status messages, such as "Chain Did Not Complete Successfully."

For technical reasons that relate to the chaining of multiple tools, somewhat different routines are run to accomplish the Cross Reference Parts and Check Electrical Rules tasks than would be the case if these tools were launched by themselves as you have done in previous tutorials. This is transparent to the user, except that the configuration options and final report are slightly different.

When OrCAD chains tools, the process terminates if an error or warning is encountered. In most cases this is not desirable. It can be avoided by setting a configuration option to ignore warnings.

Prior to launching Check Design Integrity, you must set up the local configurations. At the main SDT screen, click on Check Design Integrity and then click on Local Configuration. This brings up the second menu level shown in Figure 7-17. Click on Configure CLEANUP.

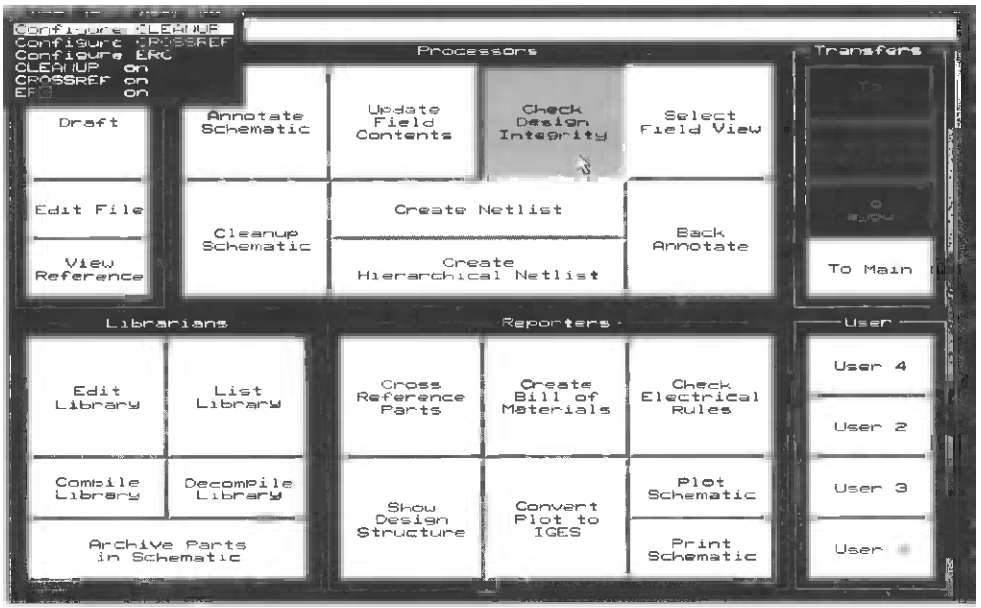


Figure 7-17 Check Design Integrity Configuration

Verify that your local configuration matches that shown in Figure 7-18 and make any required changes. Note that the Ignore Warnings option is selected.

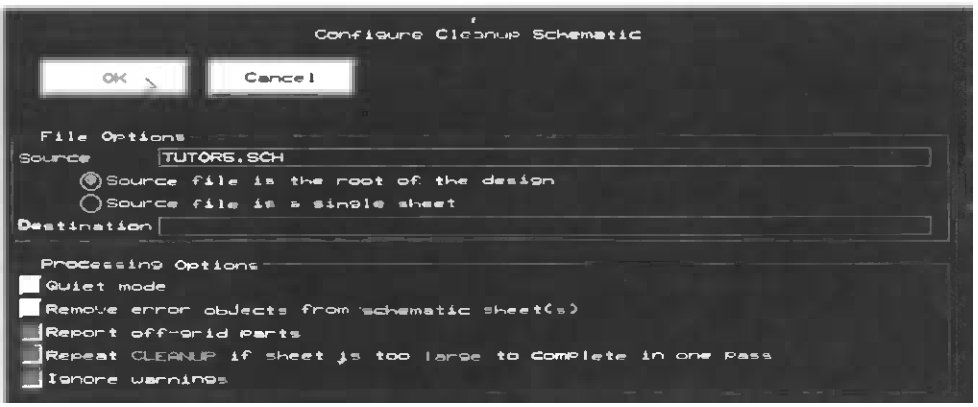


Figure 7-18 Cleanup Configuration Options

After you have verified the Cleanup configuration options, click on OK to exit. Go back to the Check Design Integrity configuration screen shown in Figure 7-17 and click on Configure CROSSREF. Verify that your local configuration matches that shown in Figure 7-19 and make any required changes.

The main reason for running the Cross Reference Parts tools is to detect parts with type mismatch and duplicate reference designators. The detailed cross reference report that you ran as part of the third tutorial exercise is only useful in special situations. Consequently, the local configuration options for Cross Reference Parts are set to eliminate this extraneous information. As with Cleanup, the Ignore Warnings option is selected.

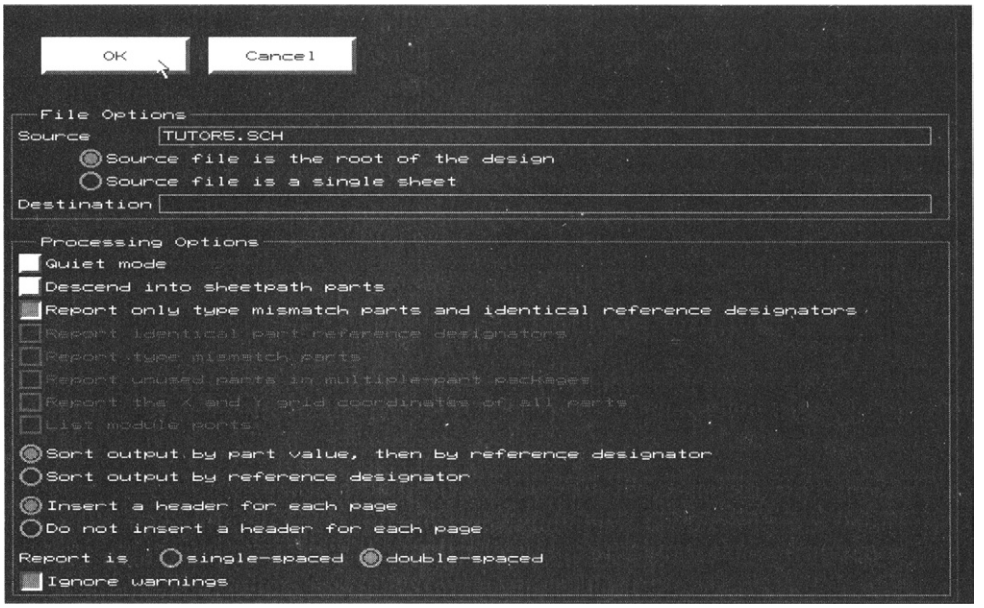


Figure 7-19 Cross Reference Parts Configuration Options

After you have verified the Cross Reference Parts configuration options, click on OK to exit. Go back to the Check Design Integrity configuration screen shown in Figure 7-17 and click on Configure ERC (Check Electrical Rules). Verify that your local configuration matches that shown in Figure 7-20 and make any required changes. After you have verified the ERC configuration options, click on OK to exit.

Note that OrCAD is inconsistent in naming the Check Electrical Rules tool. On the screen in Figure 7-20 and on the main SDT screen, it is called Check Electrical Rules. The abbreviation ERC appears on the screen in Figure 7-17.

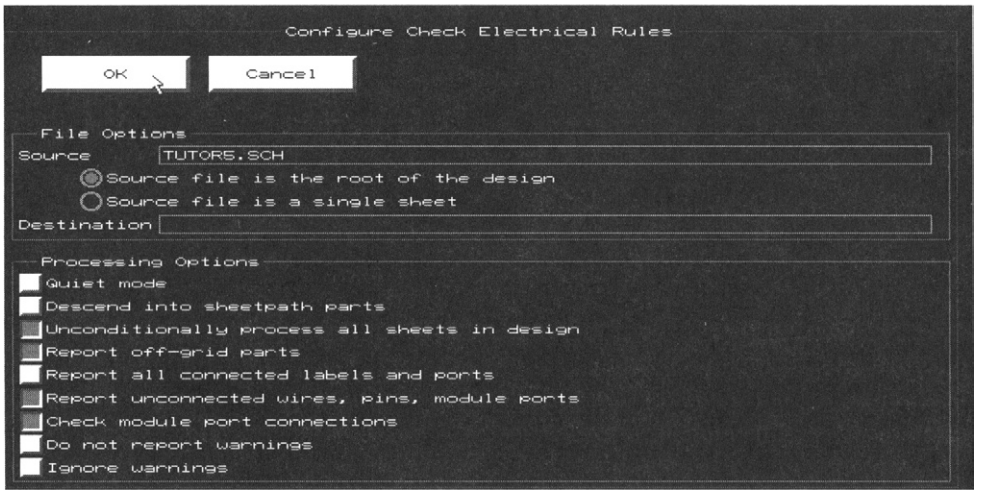


Figure 7-20 Check Electrical Rules Configuration

The focus is finding design rule violations (errors) and potential problem areas (warnings). Consequently, all the local configurations shown in Figures 7-18 through 7-20 have only those options selected that are relevant to finding errors and warnings. Note that selecting the Ignore Warnings option does not prevent the reporting of warnings. This option only prevents the chain from terminating if a warning occurs.

You will also note that the File Options Destination box is left blank in all of the local configurations. This causes each tool to write its report to the default location, the ESP_OUT.TXT file. Most users find that writing all the reports to a single file and then printing this file out is the most convenient approach.

Now that you have completed the local configurations, double click on Check Design Integrity to launch the tool. Most likely your design will have at least some minor error or warning. A status message indicating that the program chain did not complete may appear. This is normal.

Exit to DOS and use the LIST.COM utility or your text editor to examine the ESP_OUT.TXT file. Go back to Draft and fix any problems. Then rerun Check Design Integrity and again examine the ESP_OUT.TXT file. Repeat this cycle until all problems have been corrected.

Note that as in previous tutorials, you will continue to get a warning about +5V and VCC power supplies being tied together. Because the supplies were tied together intentionally, you can ignore this warning.

Wrapping up the First Session

At this point, the schematic has been completed and you have learned to use a new tool, Check Design Integrity. To wrap up the first session, run the Create Bill of Materials tool. Check for any incorrect part descriptions or other discrepancies. Next, print out the schematics using the Print Schematic tool. Carefully look over the schematic hard copy and, if necessary, make any final corrections. Then run the Archive Parts in Schematic tool to create an archive library. Use the name TUTOR5.LIB. Finally, backup your design subdirectory to floppy disk.

Second Session

In the second session you will learn about another type of OrCAD schematic structure, referred to as a flat structure, which resembles traditional multiple page schematics. You will convert the TUTOR5 design that you created in the previous session to a flat structure. You will also modify the CMOS memories on the second sheet (static RAM array) to use visible power pins. This approach clarifies the use of isolated power on the second sheet.

The second session also includes a more in-depth look at Design Management tools and the remaining SDT tools.

Overview of the Flat Design Structure

Up to now, all the schematics you have created in the tutorial exercises have been either single sheet or hierarchical structure. Single sheet schematics are limited to small, simple designs. With the trend to A size (8.5 x 11 inch) hard copy generated on laser printers, large and complex designs are best represented using a hierarchical structure. OrCAD allows a third option, referred to as a "flat" structure, which is ideal for moderate designs of three or four pages.

As mentioned above, the flat structure results in an appearance resembling that which industry has traditionally used for multiple page schematics. Sheet objects do not appear in a flat structure. Signals are routed between sheets using module ports. No direct means exist for identifying which sheet a given module port signal originates from or is routed to. This is a major limitation of the flat structure. Workarounds include modifying signal names to include a sheet number suffix or adding text notes. Even with such notations added, following signal paths on flat designs with more than three or four sheets becomes a daunting task.

Understanding Linked Files and Pipe Commands

In hierarchical designs, OrCAD provides a simple and direct means of navigating between sheets. Likewise, OrCAD automatically creates the files for new sheets when sheet objects are defined. Not so in flat designs.

Flat designs are based on the concept of linked files. The first sheet contains pipe commands that identify the file names of the remaining sheets. Pipe commands start with the special pipe character (|). This is the same character used for the DOS pipe command and appears as a broken vertical bar on most PC keyboards. On enhanced 104 key AT keyboards, the pipe character is at the top of the backslash key.

OrCAD's use of the term pipe command sometimes causes confusion. The term originated in the context of computer operating systems. In DOS, pipe commands are used to redirect the input or output of one program or command to another program or command. A common example most DOS users have encountered is the command sequence **TYPE filename.ext |MORE <ENTER>**, which lists the specified file to the screen one page at a time.

OrCAD uses the term pipe command to refer to special command sequences embedded in the schematic that are used by other tools or external programs. OrCAD pipe commands are placed on the schematic as text strings using the Place Text command. Two pipe commands reviewed in this book include |LINK and |SPICE. The |LINK command establishes links to additional sheets for multiple sheet flat designs. The |SPICE command is used when creating netlists for SPICE circuit simulation programs. Additional lines of text immediately below the |SPICE command allow passing simulation parameters and commands to the simulation program via the netlist. Note that the pipe character (|), must precede every line of text associated with an OrCAD pipe command.

The |LINK command is followed by lines of text listing the file names for the remaining sheets in the flat design. To restructure the TUTOR5 design as a flat design, you would place the following lines of text on the first sheet (do not do this now as it is covered later on in the tutorial):

```
|LINK
|CMOS.SCH
|BACKUP.SCH
```

Other OrCAD tools, such as Annotate Schematic, Check Design Integrity, Create Bill of Materials, Create Netlist, and Print Schematic utilize the |LINK command to identify the design as a flat design and to locate the files for the second and subsequent sheets.

Navigating Between Sheets in a Flat Design

OrCAD does not provide a direct "point and click" means of navigating between sheets in a flat design. Nor does OrCAD automatically create the files for a new sheet in a flat design. You must use the Quit Initialize command for both functions. If you click once on Quit Initialize, OrCAD prompts for a file name. If you double click on Quit Initialize, OrCAD lists all schematic files in the design and provides an additional option for creating a new schematic. Remember that just as in a single sheet or hierarchical design, you must use the Quit Update File command to save any edits to disk before navigating to another sheet.

Creating a CMOS Memory Part with Visible Power Pins

This session includes restructuring the TUTOR5 design as a flat design and redrawing the CMOS static RAM memory with visible power pins to clarify the use of isolated battery backup power. The logical first step in any OrCAD schematic capture process is to create any required library parts.

When you use the library editor, the selected design is not important, except that the local configuration for the Edit Library tool may change. For this reason, you should avoid using the default TEMPLATE design. You can select the TUTOR5 design. Launch OrCAD SDT, configure the Edit Library tool for the MEMORY.LIB library, and then launch Edit Library.

Use the Get Part command to retrieve the 7C128 CMOS static RAM. Then use the Export command to export the part to a temporary file. You can use a descriptive file name such as 7C128.PRT. Quit the library editor, change the local configuration to the CUSTOM.LIB library, and then launch Edit Library again.

Use the Set command to set visible power pins and grid dots. Note that setting visible power pins only affects the library editor display. The power pins are not changed to a visible pin type with this command. The pin type must be edited in order for the power pins to appear as visible pins in Draft.

Next, use the Import command to import the 7C128 part. Use the Pin command to move the pins and to edit the pin type. Refer to Figure 7-21. Use the Pin Move command to center pin 12 (GND) and pin 24 (VCC) as shown. Then use the Pin Type command to change the power pins to passive type.

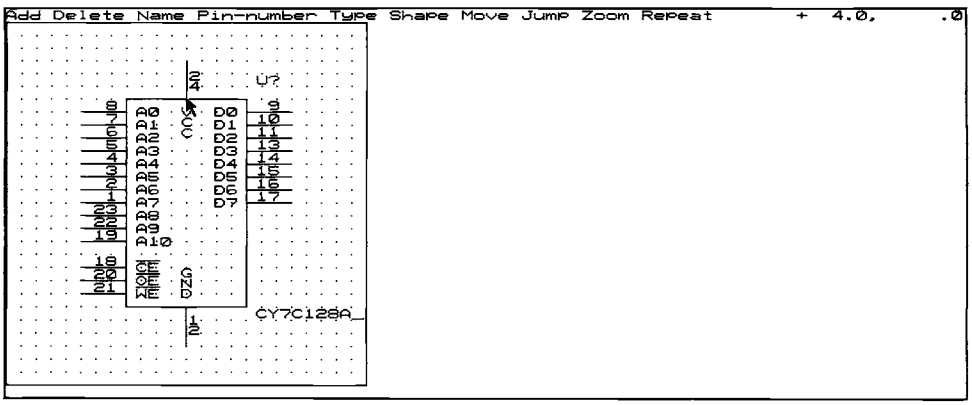


Figure 7-21 CMOS RAM with Visible Power Pins

After editing the power pins, use the Name Delete command to delete all existing names for the CMOS static RAM. You will have to repeatedly use the command to delete the existing names one by one. Then use the Name Add command to add the name CY7C128A_VIS. The VIS suffix is suggested as a means of identifying the part as a special part with visible power pin.

To save the part edits, use the Library Update command and then the Quit Update File command. The edited part with visible power pins is now saved into your CUSTOM.LIB library. Quit the library editor and exit SDT.

Because you have changed CUSTOM.LIB, good practice dictates making a new backup copy on floppy disk.

Using Design Management Tools to Copy a Design

You will copy the TUTOR5 design and then restructure the copy as a flat design. Design Management Tools provides a convenient means of copying an existing design. Launch Design Management Tools and click on Copy Design as shown in Figure 7-22. The screen shown in Figure 7-23 appears.

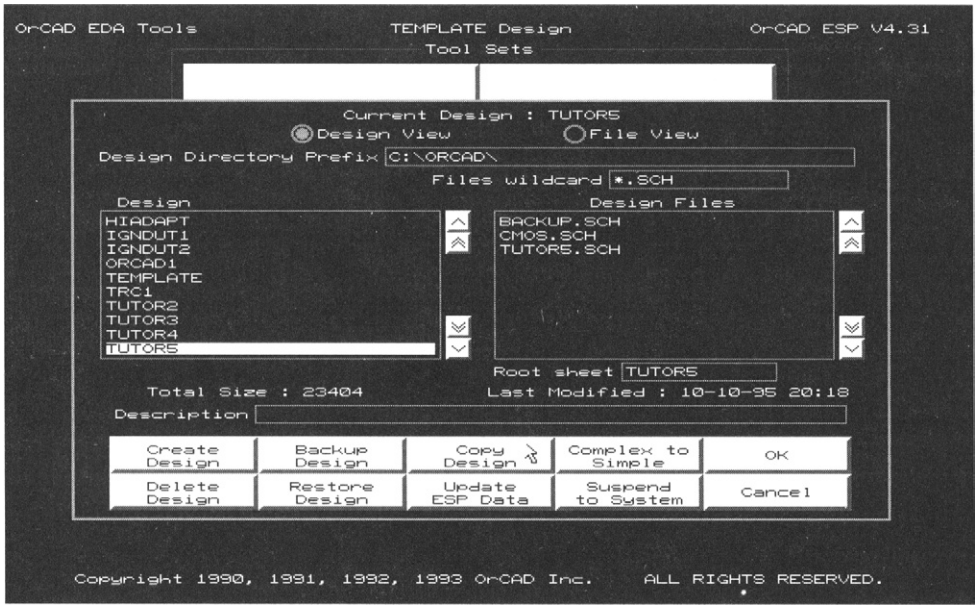


Figure 7-22 Selecting the Copy Design Command

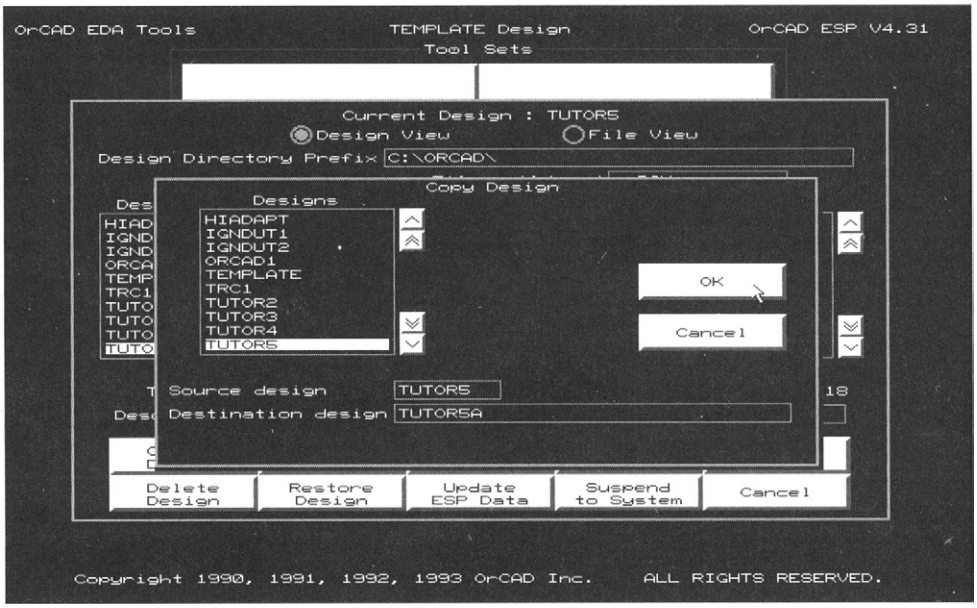


Figure 7-23 Entering Parameters for Design Copy

Using the Design Copy command is relatively straightforward. As shown in Figure 7-23, a scroll box appears for selecting the source design. Select the TUTOR5 design as the source. Then enter the name of the destination design. Use the name TUTOR5A. Click on OK to initiate the copy process. OrCAD automatically creates a new subdirectory named TUTOR5A and copies all files from the source subdirectory TUTOR5. OrCAD also renames the root sheet TUTOR5A.SCH.

At this point, it is appropriate to take a quick look at some of the other functions available in Design Management Tools.

Overview of Design Management Tools

Commands available in Design Management Tools appear as two ribbon bars at the bottom of the screen as shown in Figure 7-22. The commands include:

Create Design Creates a new design using the TEMPLATE design as a model. All configuration files and macro files are copied into the new design.

Backup Design Creates a special backup file containing all design information on a floppy or other media. Reviewed in detail in a subsequent section.

Copy Design	Copies an existing design. Creates a new subdirectory and copies all files from the specified source design. Also renames the root sheet to match the new design name.
Complex To Simple	Specialized command used to change a complex hierarchy to a simple hierarchy. Complex hierarchies are used to represent repeated circuitry in large gate arrays or ASICs (application specific ICs). The subject is beyond the scope of this book.
Delete Design	Deletes all files and the subdirectory associated with an existing design. Reviewed in detail in a subsequent section.
Restore Design	Restores a special design backup file created with the Backup Design command. Reviewed in detail in a subsequent section.
Update ESP Data	Updates configuration files based on the TEMPLATE design. Useful for updating existing designs if new drivers have been defined in the TEMPLATE design. Reviewed in detail in a subsequent section.
Suspend To System	Exits to DOS. Allows the user to run DOS commands. OrCAD remains in the background. After the DOS commands have been completed, type **EXIT <ENTER>** at the DOS prompt to return to OrCAD. Do not load memory resident utilities or attempt to run programs other than basic DOS commands while OrCAD is suspended.
Cancel	Cancels Design Management Tools and returns to ESP main screen without selecting a new design directory. Does not cancel or undo other Design Management Tools commands.
OK	Returns to ESP main screen with the current design selection. Normal exit from Design Management Tools.

In the following three sections, the Backup Design, Restore Design, Delete Design, and Update ESP Data commands will be explored in more detail.

Using Design Management Tools to Backup a Design

Clicking on the Backup Design command brings up the screen shown in Figure 7-24. A scroll box allows selecting the design to backup. Backup is normally done to floppy disk. The Destination prefix box allows entering the drive letter and path for the backup file.

Buttons for the destination disk type appear on the right side of the screen. The buttons are primarily informative. When a destination such as drive A: is entered, the program reads the media type and highlights the appropriate button. This action informs the user as to what type of media must be inserted into the drive. Overriding the selection by clicking on a different button does not seem to affect the operation of the program. In one test, the program misread a 3.5 inch 720KB floppy as being 5.25 inch 1.2MB, but still copied the backup file correctly. Note that the program requires formatted floppy disks. Disks cannot be formatted on the fly.

Backup files are named DESIGNNAME.N where N is a number starting at one. If the backup is too big to fit onto a single disk, you are prompted to insert additional disks. The backup files on subsequent disks are sequentially numbered.

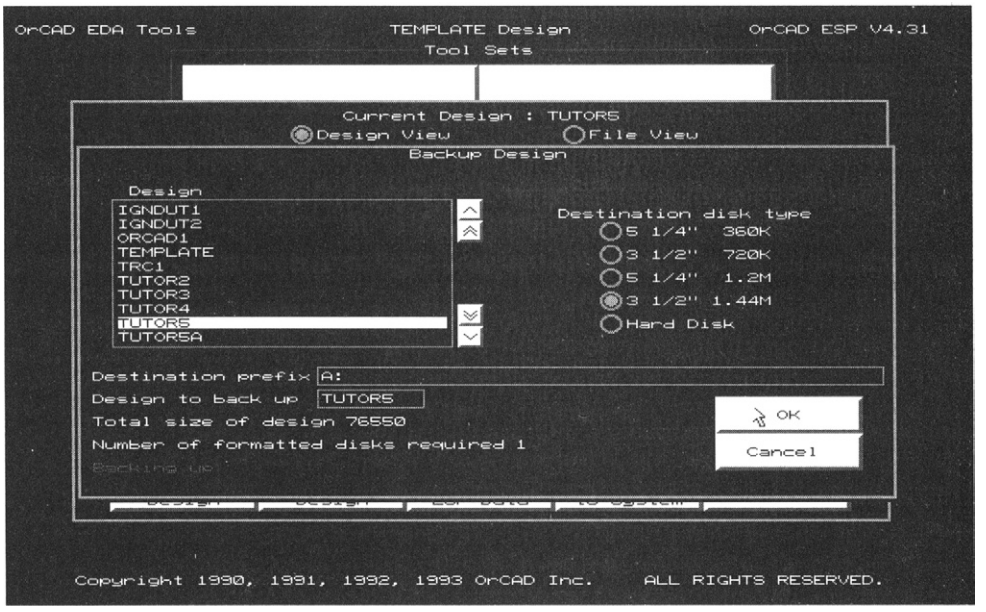


Figure 7-24 The Backup Design Screen

A disadvantage of the Backup Design command is that all data in the design directory is written to a single backup file. The only way to later access this data

is to use the Restore Design command. In practical situations, users often find that a particular file has been corrupted. Perhaps a design was edited and objects were inadvertently lost on one sheet. The OrCAD Backup and Restore tools provide no means of selective access to a particular file. Most users find that using the DOS Copy commands to copy files onto a backup floppy offers far more flexibility.

Using Design Management Tools to Restore a Design

Clicking on the Restore Design command brings up the screen shown in Figure 7-25. The Source prefix box allows entering the drive letter and path for the backup file that is to be restored. A scroll box allows selecting the design to restore if more than one design is found. Click on the design to be restored. The name then appears in the Designs to restore to box. You can edit the name in this box. This has the effect of restoring the selected design and changing the name in one operation. If you enter a different design name to restore to, the root schematic is also renamed.

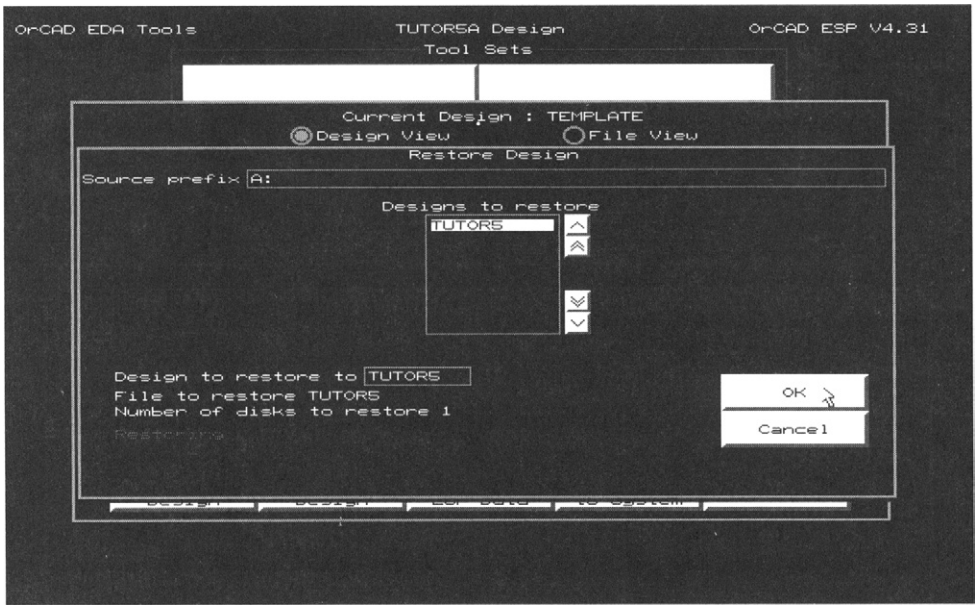


Figure 7-25 The Restore Design Screen

When a design is restored, the backup file is expanded and the individual design files are written to a subdirectory under C:\ORCAD. The restore operation will

abort with an error message if the selected design already exists under C:\ORCAD. Restore Design does not have any provision for overwriting an existing design.

While Restore Design does not allow selective restoration of files, the capability of changing the design name offers at least a partial workaround. If you require access to a particular file, restore the design using a new name and then exit to DOS and copy the required file.

Using Design Management Tools to Delete a Design

Clicking on the Delete Design command brings up the screen shown in Figure 7-26. Select the design to be deleted in the scroll box. Once you select a design and click on OK, a warning screen appears as shown in the figure. You must click on OK again to delete the selected design. OrCAD then deletes all files in the design subdirectory and the subdirectory itself.

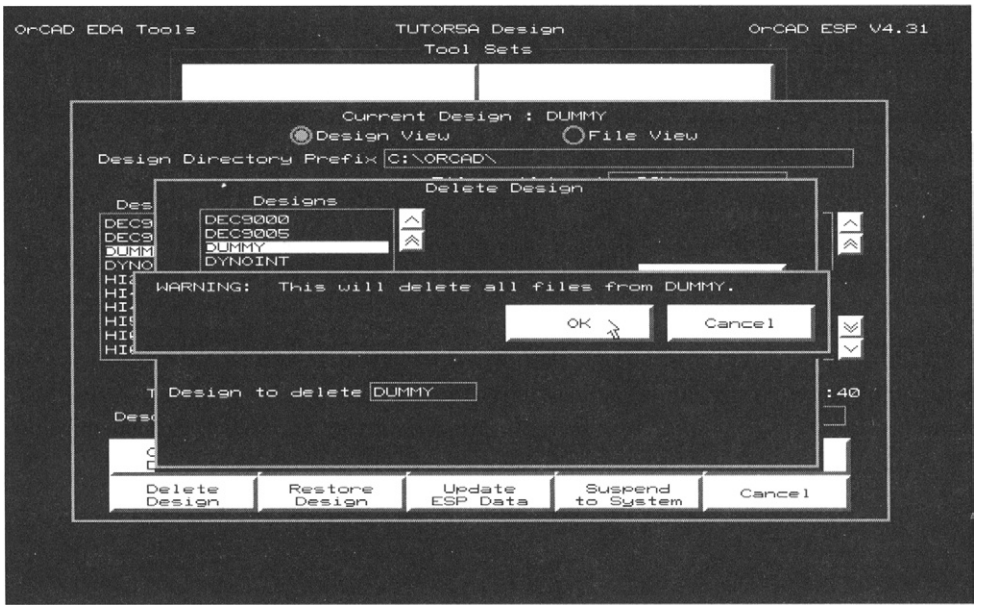


Figure 7-26 The Delete Design Screen

Exercise caution when using the Delete Design command because no provision exists to undo the action. DOS 6.0 or higher and the Norton Utilities provide tools for undeleting files. The Norton Utilities Unerase tool can sometimes

recover a deleted subdirectory, but the overall prognosis is generally poor when a mistake has been made using the Delete Design tool. Using DOS commands to delete designs and always keeping a backup on floppy disk is your best assurance against costly mistakes.

Using Design Management Tools to Update ESP Data

The Update ESP Data command works on the current design selected on the main Design Management tools screen. Clicking on the Update ESP Data command brings up the confirmation screen shown in Figure 7-27. You must click on OK to update the selected design. OrCAD then copies all configuration files from the TEMPLATE design to the current design.

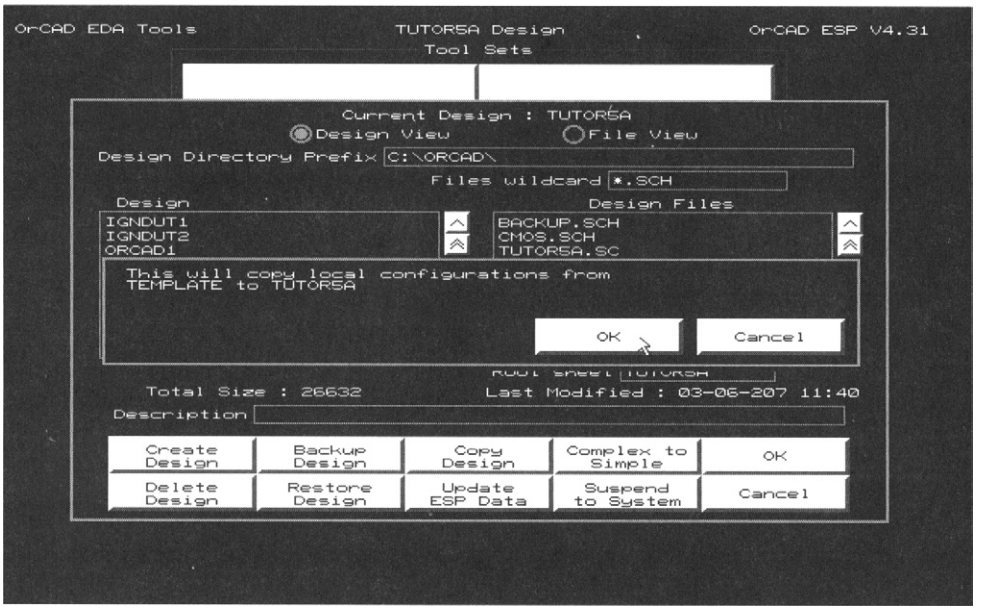


Figure 7-27 The Update ESP Data Screen

The Update ESP Data command is especially useful when new hardware has been installed and video, printer, or plotter drivers must be changed in existing designs. You can make the changes to the TEMPLATE design and then use the Update ESP Data command to replicate the changes. A minor limitation is that macro files are not updated.

Converting to a Flat Structure

Now that you have explored Design Management Tools, let's get back to the schematic. So far you have created a new CMOS static RAM part with visible power pins and copied the TUTOR5 design to a new design named TUTOR5A. From here on, you will work on the TUTOR5A design and convert it into a flat structure.

Select the TUTOR5A design and launch Draft. The model for the flat structure version of the design is shown in Figures 7-28 through 7-30.

The first step is to delete the sheet objects on the first sheet. Take a careful look at Figure 7-28. The address, data, and control signals that were previously routed to sheet objects now go to module ports. The module ports carry these signals to the other two sheets. You can use Block commands to rearrange the first sheet. Reroute the address bus and data bus as shown in the figure. Then add the module ports and then route the remaining signal lines.

The final step on sheet one is to add the |LINK pipe command. Use the Place Text command to place the four lines of text shown near the bottom of the sheet. This establishes the link to the remaining sheets.

Sheet two and sheet three do not contain any sheet objects. These sheets already have all the module ports required to carry signals to and from sheet one. At this point, you have converted the design to a flat structure. Text notes are added near the module ports to provide signal routing information. These notes have no effect on the electrical connectivity database and are not required by OrCAD. Their only purpose is to aid the reader in understanding the signal routing.

Clarifying Isolated Power Usage via Visible Power Pins

This step is independent of the design structure. The two possible approaches, invisible power pins with VCC tied to an unspecified module port or using visible power pins directly tied to the isolated power supply, can be employed with either a hierarchical or a flat design structure.

No changes are required on sheet one. On sheet two, delete U9 and U10. Then get the CY7C128A_VIS part from CUSTOM.LIB and place it where U9 and U10 were located. Edit the reference designator and part descriptions. Use the Block Drag command to move the two ICs further apart so that you can add power and ground objects. Remove the +5VBAK unspecified module ports from sheets two and three. Then add the +5VBAK power objects as shown in Figures 7-29 and 7-30.

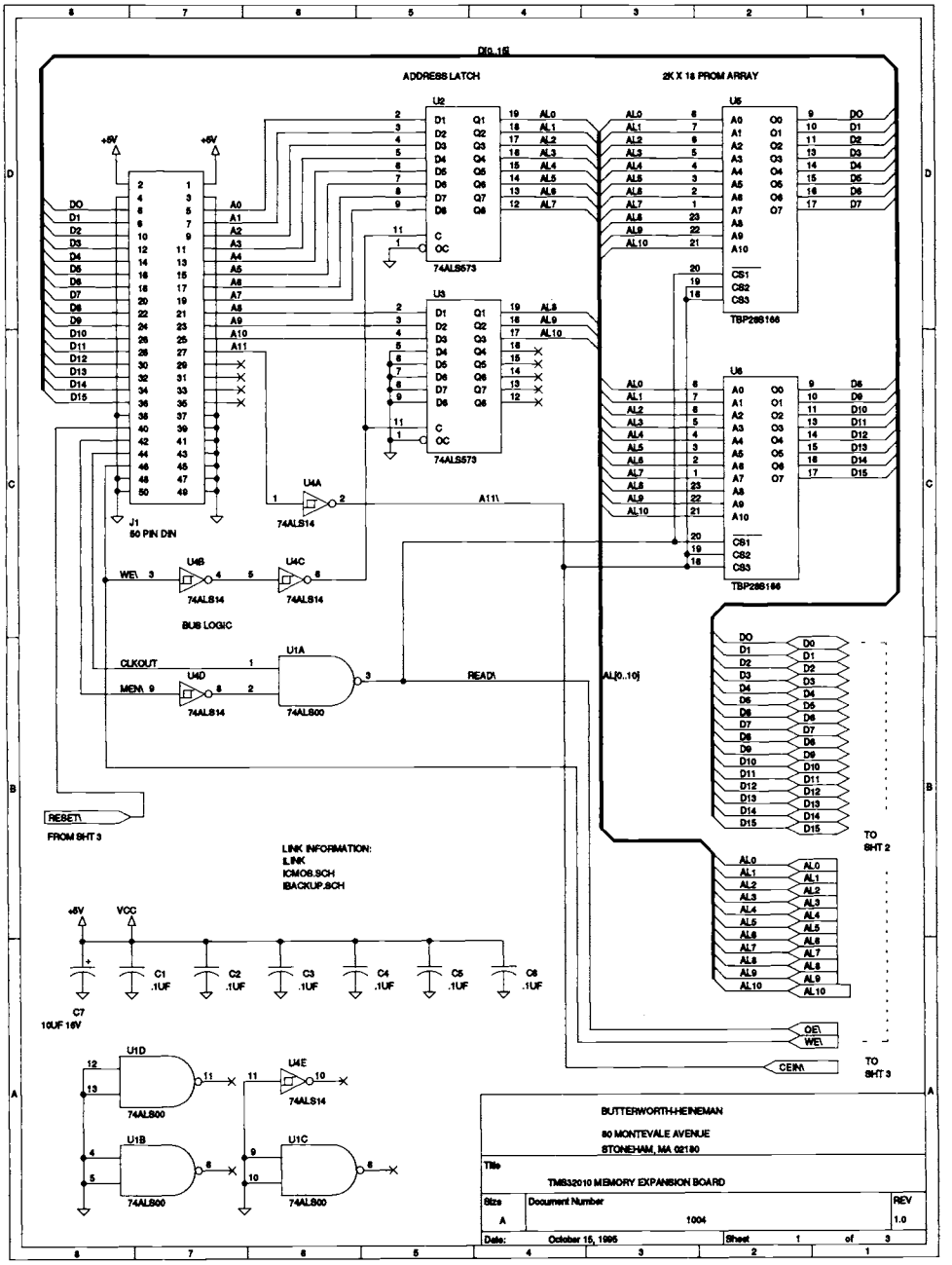


Figure 7-28 Revised Sheet One

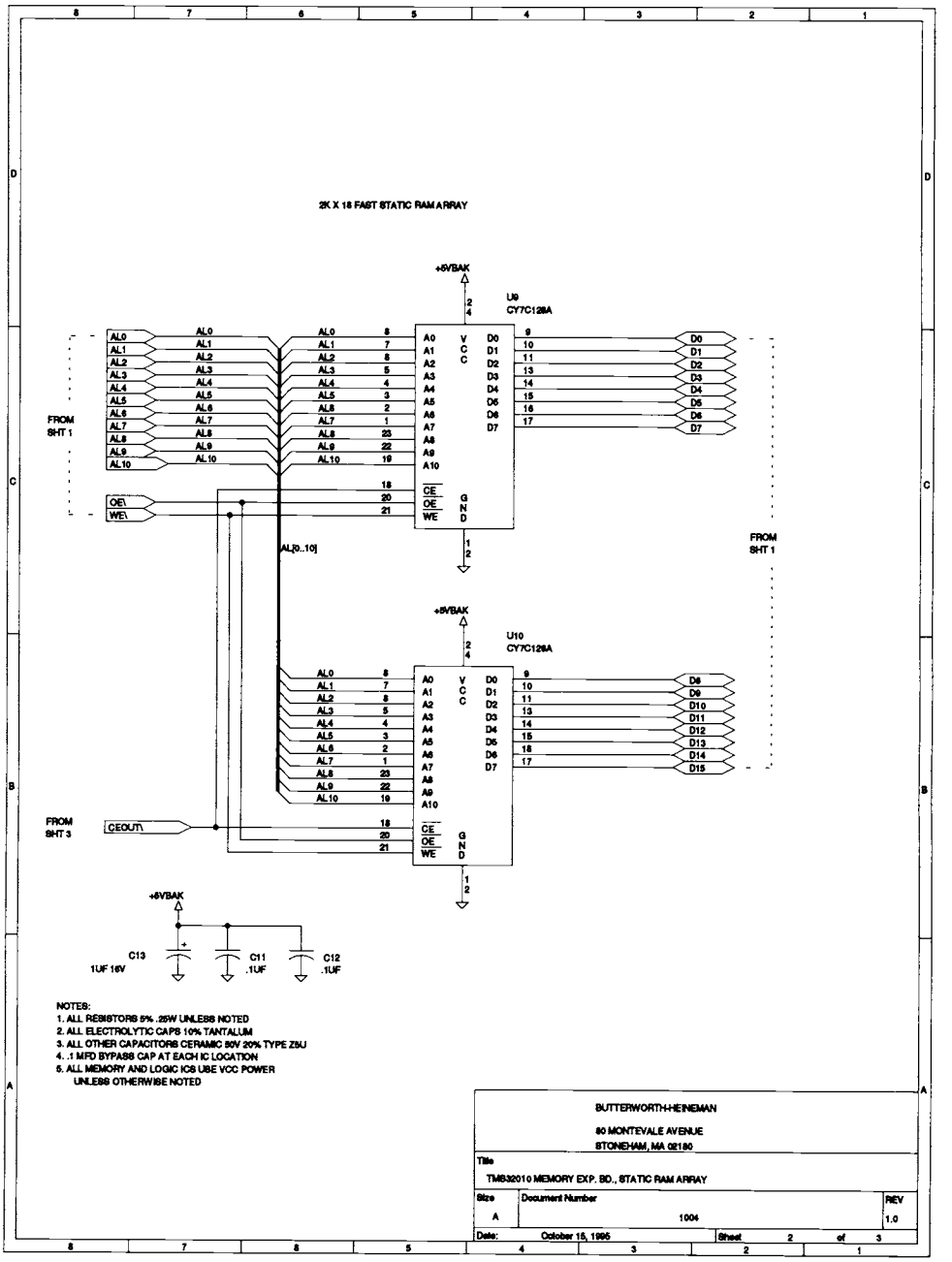


Figure 7-29 Revised Sheet Two

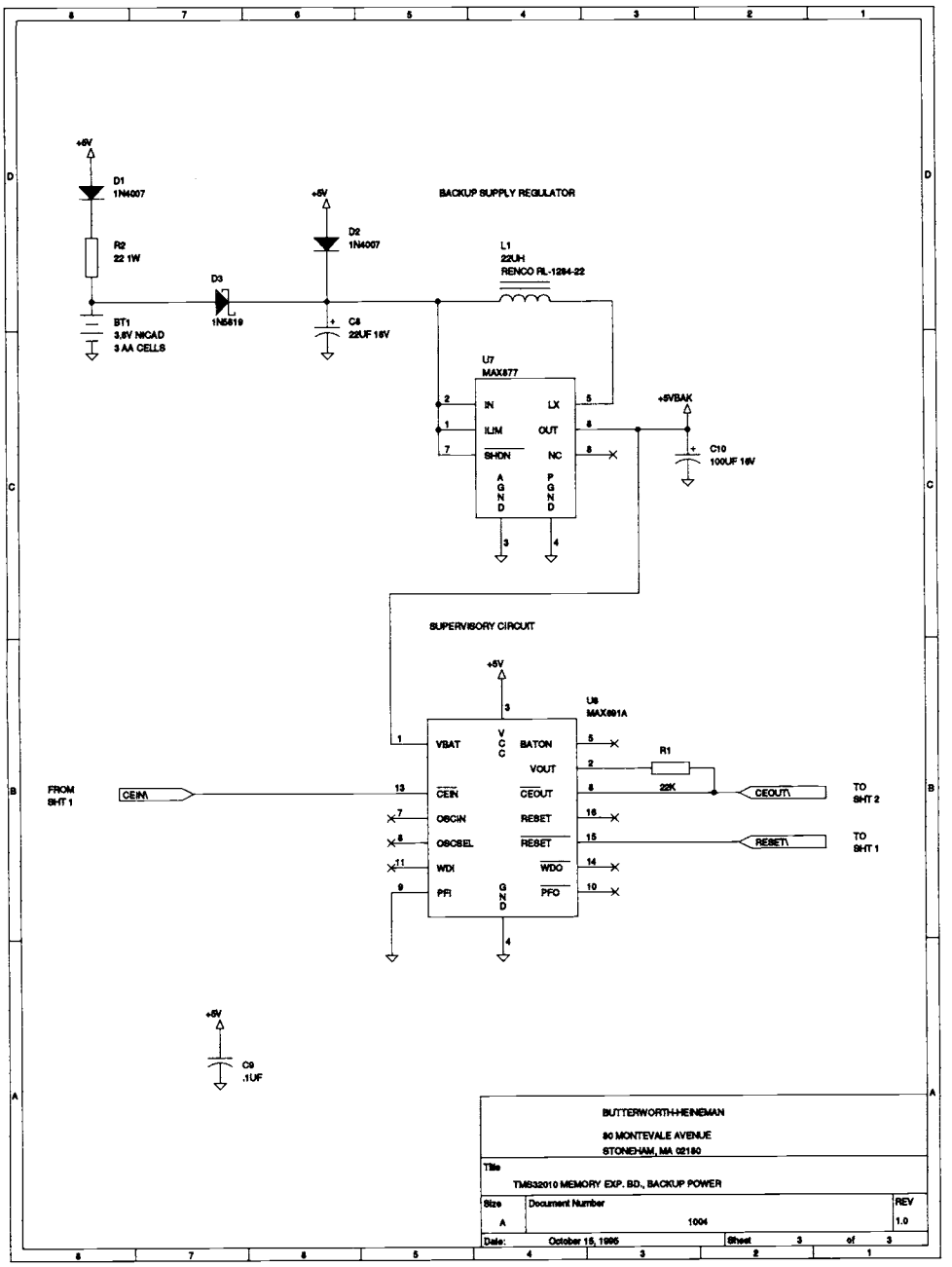


Figure 7-30 Revised Sheet Three

Wrapping up the Second Session

You have now converted the schematic to a flat structure and clarified the use of isolated power via visible power pins on the CMOS static RAM array. Your completed schematic should match the model in Figures 7-28 through 7-30.

Before proceeding, check for errors by running the Check Design Integrity Tool. After you have corrected any errors, print out the schematic and backup the design on floppy disk.

Third Session

The third session covers the remaining OrCAD tools that have not been discussed in previous tutorials, including the Update Field Contents, Select Field View, and Back Annotate tools. You will also learn advanced techniques for use with the Create Bill of Materials and Create Netlist tools. The session starts with a discussion of key fields. After reading the OrCAD documentation, many users abandon key fields as a hopelessly complex and confusing subject. You will learn to use key fields via a hands-on approach as you work through the remaining exercises in the tutorial.

Introduction to Key Fields

Postprocessing tools such as Create Bill of Materials and Create Netlist extract parts information from one or more of the ten part fields. Recall that the ten part fields include two preassigned fields for reference designator and part value and eight additional user definable fields. Key fields set in the main SDT configuration define how information stored in the ten part fields is combined or compared during postprocessing operations. The key field section of the SDT configuration screen in shown in Figure 7-31.

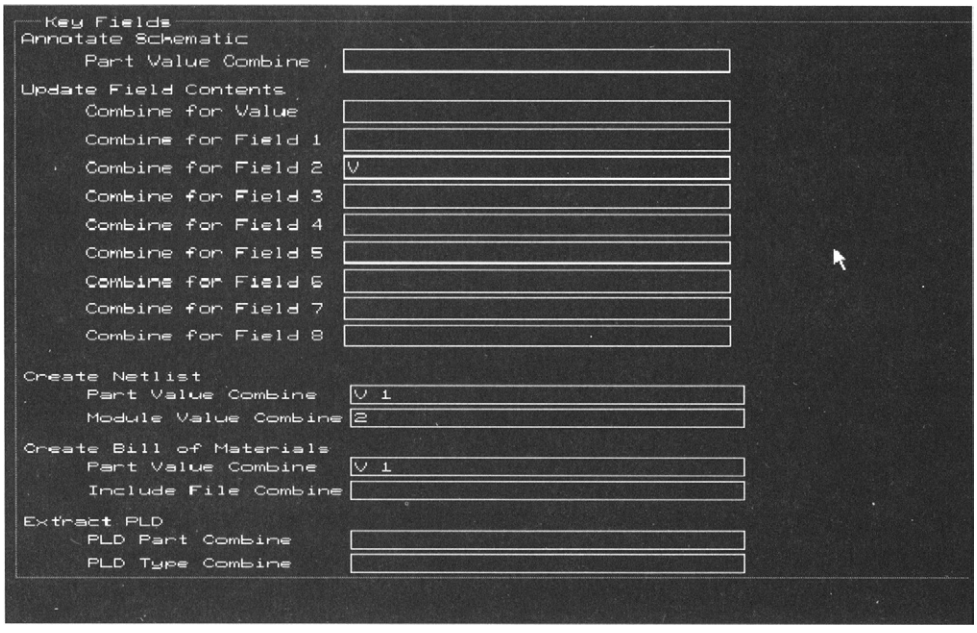


Figure 7-31 Key Fields Configuration

The characters entered in the key fields are used to reference one of the ten part fields. The following characters can appear in a key field:

R	The character R represents the reference designator field.
V	The character V represents the part value field.
1-8	The numerals 1-8 represent user definable part fields 1 through 8.

There is a limitation of 127 characters per key field or part field. The interpretation of the key field depends on the particular postprocessing routine, as summarized in the following sections. Extract PLD is skipped because this tool involves material beyond the scope of the book.

Note that if no entry is made in the key fields, certain defaults apply, which are also discussed below.

Using Key Fields with Annotate Schematic

Recall that Annotate Schematic assigns values to reference designators. Many ICs come in multiple gate packages. Annotate Schematic uses the part value to determine how to group gates together for the purpose of assigning reference designators. For example, an 74HCT00 contains four NAND gates. If eight NAND gates appeared on the schematic, Annotate Schematic would assign reference designators such as U1A, U1B, U1C, U1D, U2A, U2B, U2C, and U2D. The sequence would depend on the order the gates were placed onto the schematic and would tend to be somewhat random. In many cases, the user prefers to group certain gates into a particular package, perhaps due to concerns about signal propagation delays.

The Annotate Schematic Part Value Combine key field shown in Figure 7-31 can be used to control how Annotate Schematic groups IC gates. One of the eight user definable part fields is used to store a package ID. For example, you could use part field three for the package ID. Then use the Edit command in Draft to assign values such as PACK1 or PACK2 to gates that you want to group together in a given IC package. The entry V3 in the Part Value Combine key field then instructs Annotate Schematic to combine the information in the preassigned part value field (V) and user defined part field 3 for the purpose of grouping gates when assigning reference designators. Note that other parts, or IC gates without any entry in part field 3, are unaffected.

If no entry appears in the Part Value Combine key field, the field defaults to V, and Annotate Schematic just uses the part value.

While the use of a key field to control Annotate Schematic appears to be a powerful feature, its practical value is limited. In most cases, only a few parts have multiple gates where grouping is critical to circuit performance. Using Draft to edit a few reference designators on the schematic is often the quickest and easiest solution.

For analog boards, grouping of multiple "gate" packages such as the ubiquitous LM324 quad opamps and LM339 quad comparators is often based on PCB layout considerations. The grouping typically changes during a phase of the PCB component placement process known as gate swapping. Gate swapping reduces the overall length and complexity of circuit trace between the gates and associated critical components. After gate swapping, the user must update the schematic, a process referred to as back annotation. OrCAD provides the Back Annotate Schematic tool to facilitate this process. The point of this discussion is that the user should look at the overall picture. Undue emphasis on grouping gates during the initial schematic capture phase may be misguided because changes during PCB layout are almost inevitable.

Using Key Fields with Create Bill of Materials

As shown in Figure 7-31, OrCAD supports two key fields for the Create Bill of Materials tool. The Part Value Combine key field functions in a somewhat similar manner as explained above for Annotate Schematic. The Include File Combine field is used for specialized bill of materials processing functions. If no entries appear, OrCAD uses the default V (for part value) for both of these key fields.

The entry V 1 for the Part Value Combine key field shown in Figure 7-31, instructs Create Bill of Materials to combine the information in the preassigned part value field (V) and user defined part field 1 for the part description reported on the bill of materials. Several of the parts on the tutorial schematic contain additional descriptive information in part field 1. When you run Create Bill of Materials later on in this session, all this information will be included in the report.

The second key field instructs Create Bill of Materials how to scan an "include file." An include file is an ASCII text file created with a text editor that contains additional descriptive information to be "included" into the bill of materials report. The effect is somewhat similar to using additional parts fields to build a longer part description, except the information is merged from an external file.

The following example illustrates the include file format:

' '	DESCRIPTION
'.1UF'	50V 20% Z5U CERAMIC DISK
'22K'	.25W 5% CARBON FILM RESISTOR

The first line of the include file is the header. The header must begin with two single quotes with no characters or spaces in between. The remaining lines consist of a part value enclosed in single quotes along with the additional descriptive information to be included in the bill of materials.

When OrCAD creates a bill of materials with the include file option selected, the combined part value (as configured in the key field) for each part on the schematic is compared to each line in the include file. When a match occurs, the additional descriptive text on that line is written to the bill of materials.

The final bill of materials report generally requires more detailed information than the basic part descriptions that appear on the schematic hard copy. Three options exist for handling this requirement:

- **Editing the bill of materials**. The schematic database contains only the basic part descriptions, such as .1UF capacitor or 1K resistor. A text editor is used to edit the bill of materials and to add the additional information. This option is the most cost effective if your schematics contain a low percentage of common parts, and if engineering changes throughout the product life cycle are expected to be minimal.
- **Defining additional part fields**. One or more of the eight available user defined part fields are used for additional part descriptions. The Part Value Combine key field is appropriately configured. This option results in a bill of materials report that requires only minimal editing. However, if the schematic contains many parts with identical values, a substantial additional effort is required in Draft to properly edit every part field.
- **Using an include file**. The information in the include file is merged onto the bill of materials. You are trading-off time spent editing and updating the include file for time that you would otherwise spend editing the bill of materials. This option is very cost effective if your schematics contain a high percentage of common parts or if frequent, extensive engineering changes are expected. Perhaps due to the additional time required up front, few OrCAD users use include files.

Using Key Fields with Create Netlist

OrCAD supports two key fields for the Create Netlist tool. The Part Value Combine key field functions in an identical manner to that explained above for

Create Bill of Materials. The other field is referred to as the Module Value Combine key field. Recall that "module value" is the term OrCAD uses for a physical part layout description on the netlist. If no entries appear, OrCAD uses the default V (part value) for both of these key fields.

The entry V 1 for the Part Value Combine key field shown in Figure 7-31, instructs Create Netlist to combine the information in the preassigned part value field (V) and user defined part field 1 for the part description on the netlist. In most cases, you should use the same Part Value Combine key field entry for both the Create Bill of Materials and Create Netlist tools.

The entry 2 for the Module Value Combine key field shown in Figure 7-31, instructs Create Netlist to use the information in user defined part field 2 for the netlist module value. Later on in this session, you will merge module value information into the second part field and then extract it onto a netlist.

Using Key Fields with Update Field Contents

The Update Field Contents tool is a new tool that will be explored later in this session. Update Field Contents allows a selected part field to be updated using data from a "stuff" file. Only one selected part field can be updated each time you run the tool, as determined by the tool's local configuration. The reference designator field is protected and cannot be updated.

The stuff file is an ASCII text file created with a text editor. The following example illustrates the stuff file format:

```
'.1UF'      'CAP\40LS'
'22K'       'R\40LS'
'74ALS14'   'DIP14'
```

Each line of the stuff file contains two ASCII strings enclosed in single quotes. The first string is the match string against which the schematic part database is searched. The second string is the update string that will be stuffed into the selected part field when a match is found. In the example above, the match string is part value and the update string is the module value to be used in the netlist.

OrCAD supports nine key fields for the Update Field Contents Tool. As shown in Figure 7-31, these key fields are referred to as Combine For Value and Combine For Field 1 through 8. Note that fields 1 through 8 refer to the eight user definable part fields. Each key field functions in the same manner. No defaults exist. If a key field is to be used, an entry is required.

When the Update Field Contents tool is launched, OrCAD reads the local configuration to determine which field is to be updated. The combined value (an

ASCII string constructed based on the corresponding key field) for each part on the schematic is compared to the match string on each line of the stuff file. When a match occurs, the update string on that line is written to the selected part field.

While the description of this process sounds terribly complicated, the Update Field Contents tool is relatively easy to use. The most common usage is for adding module value data (physical layout descriptions) to a schematic prior to running Create Netlist.

Continuing the Tutorial Exercise - Configuring Key Fields

The remainder of this session covers OrCAD tools that utilize key fields. Enter the OrCAD SDT Configuration screen and setup your key fields as shown in Figure 7-31. Note that the entry V 1 on the Part Value Combine key field for Create Bill of Materials and Create Netlist has a space between the characters V and 1. This causes the tools to add a space between the two strings when combining part value and part field 1.

Running the Bill of Materials

Run the Create Bill of Materials tool on the TUTOR5A design using the same technique you learned in previous tutorials. Then use a text editor to examine and print out the bill of materials listing. If you find any errors, go back and correct the design. Remember to always backup all design files onto a floppy disk before making major changes or before going on to the next postprocessing step. A bill of materials listing corresponding to the tutorial model is shown in Figure 7-32. Note that your reference designators may be different depending on the order in which you placed parts into the design.

As specified in the SDT key fields configuration, part value and part field 1 have been combined on the bill of materials part description. This can be seen clearly on BT1 and L1.

```
MS32010 MEMORY EXPANSION BOARD                    Revised:    October 15, 1995
004                                               Revision: 1.0
ill Of Materials                 October   19, 1995      18:29:51      Page    1

 Item  Quantity  Reference                    Part
-------------------------------------------------------------------------------

    1         1  BT1                          3.6V NICAD 3 AA CELLS
    2         9  C1,C2,C3,C4,C5,C6,C9,C11,    .1UF
                 C12
    3         1  C7                           10UF 16V
    4         1  C8                           22UF 16V
    5         1  C10                          100UF 16V
    6         1  C13                          1UF 16V
    7         2  D1,D2                        1N4007
    8         1  D3                           1N5819
    9         1  J1                           50 PIN DIN
   10         1  L1                           22UH RENCO RL-1284-22
   11         1  R1                           22K
   12         1  R2                           22 1W
   13         1  U1                           74ALS00
   14         2  U2,U3                        74ALS573
   15         1  U4                           74ALS14
   16         2  U5,U6                        TBP28S166
   17         1  U7                           MAX877
   18         1  U8                           MAX691A
   19         2  U9,U10                       CY7C128A

Command:
```

Figure 7-32 TUTOR5A.BOM Bill of Materials Listing

Running Update Field Contents

The next part of the tutorial involves using Update Field Contents to add module values to the schematic. The process requires a stuff file that contains match strings and module value strings, as discussed on page 290. You can create a stuff file such as the one shown in Figure 7-33 by using a text editor.

```
'3.6V NICAD 3 AA CELLS'          'BAT\3AA'
'.01UF'                          'CAP\40LS'
'.022UF'                         'CAP\40LS'
'.047UF'                         'CAP\40LS'
'.1UF'                           'CAP\40LS'
'1UF 16V'                        'ECAP\10LS20'
'2.2UF 16V'                      'ECAP\10LS20'
'4.7UF 16V'                      'ECAP\10LS20'
'10UF 16V'                       'ECAP\10LS25'
'22UF 16V'                       'ECAP\10LS25'
'47UF 16V'                       'ECAP\20LS35'
'100UF 16V'                      'ECAP\20LS35'
'1N4007'                         'D\40LS'
'1N4148'                         'D\40LS'
'1N5819'                         'D\40LS'
'25 PIN FEMALE DB25'             'CONN\25DB\FEMALE'
'50 PIN DIN'                     'CONN\50DIN'
'22UH RENCO RL-1284-22'          'L\RENCO\1'
'1Ω 1W'                          'R\60LS'
'22 1W'                          'R\60LS'
'10 2W'                          'R\80LS'
'22 2W'                          'R\80LS'
'10K'                            'R\40LS'
'22K'                            'R\40LS'
'47K'                            'R\40LS'
'100K'                           'R\40LS'
'74ALS00'                        'DIP14'
'74ALS04'                        'DIP14'
'74ALS14'                        'DIP14'
'74ALS244'                       'DIP20'
'74ALS573'                       'DIP20'
'TBP28S166'                      'DIP24'
'MAX877'                         'DIP8'
'MAX691A'                        'DIP16'
'CY7C128A'                       'DIP24'
```

Figure 7-33 PARTLIST.STF Update File Listing

For convenience, the completed stuff file, PARTLIST.STF, is included in the TUTOR5A subdirectory on the disk supplied with this book.

Each line of the file consists of a part value match string followed by a module value update string. The module values used in this example are physical layout descriptions for PADS Work, a leading PCB design program that the author uses. Some examples of module value names include: R/40LS for a axial lead resistor with .40 inch lead spacing, ECAP/10LS20 for a radial lead tantalum capacitor with .10 inch lead spacing and .20 inch diameter, and DIP14 for a 14 pin DIP package.

The module value associated with each part will be added to part field 2. You have already configured the key field required for the part value match string. The last step before running Update Field Contents is to set the local configuration.

At the main SDT screen, click on Update Field Contents and then click on local configuration as shown in Figure 7-34. This brings up the configuration screen. Verify that your local configuration matches that shown in Figure 7-35 and make any required changes.

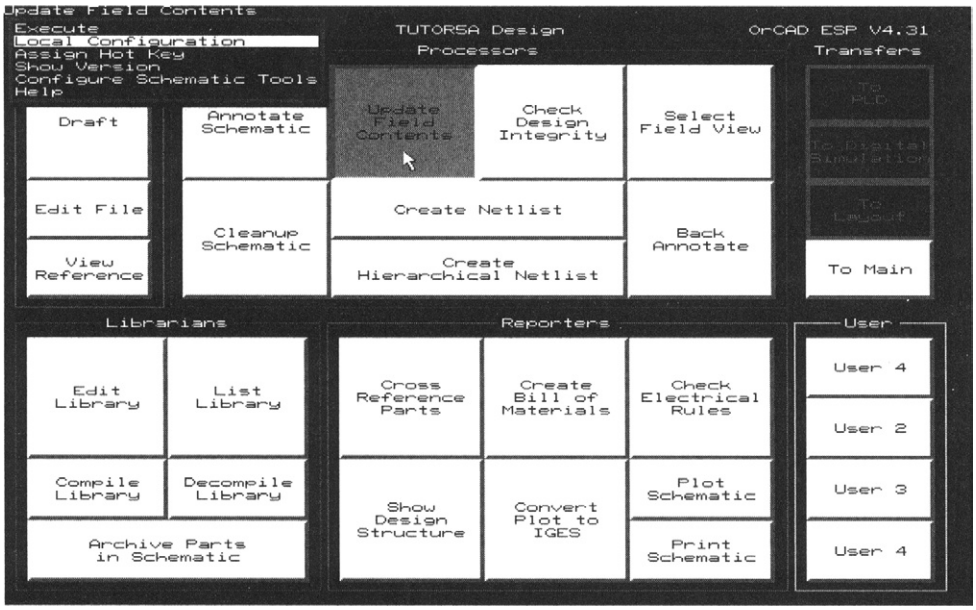


Figure 7-34 Selecting Update Field Contents Configuration

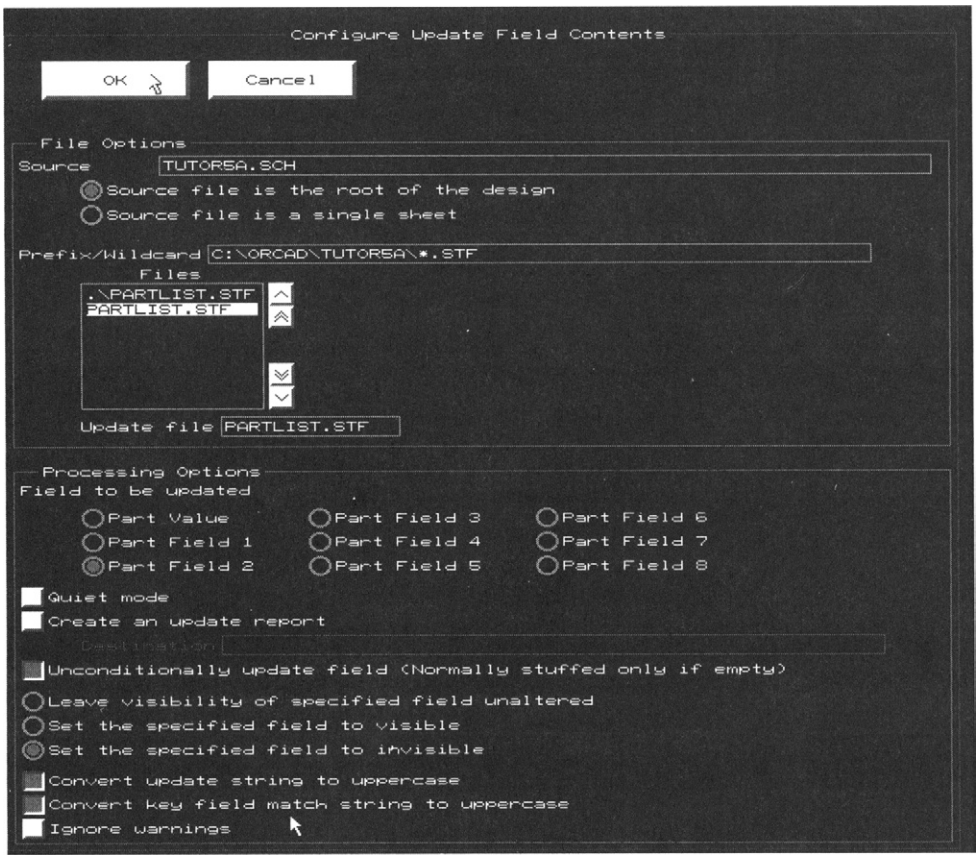


Figure 7-35 Update Field Contents Configuration Options

As with most other OrCAD tools, the Update Field Contents local configuration screen has the familiar file options section. A scroll box is used to select the update (stuff) file. Other processing options are explained below:

Field to be updated. Only one part field can be updated during each pass of the tool. The reference designator field is protected and cannot be updated. Note that an entry must be made in the key field (SDT configuration) corresponding to the selected part field. Users often forget this.

Quiet mode. Suppresses display of text messages. Leave deactivated so that status messages will be displayed.

Create an update report. Causes the program to generate a report that lists all parts and fields that have been updated. If this option is selected,

you can use the Destination entry box to specify the filename for the report.

Unconditionally update field. Unless this option is selected, the selected field is only stuffed if it was originally empty. Normally you do want Update Field Contents to write over any existing data, so select this option.

Part field visibility options. Three mutually exclusive options are available: leave specified field unaltered, set to visible, or set to invisible. Because module values are usually not displayed, select the "set invisible" option. Note that all of the eight user defined part fields default to visible unless set otherwise.

Convert update string to uppercase. Useful in certain situations. Normally all characters on a schematic are uppercase. Selecting this option as a default avoids potential problems if a lower case character is encountered.

Convert key field match string to uppercase. Useful in certain situations. Selecting this option as a default avoids potential problems if a lower-case character is encountered. Note that the match string comparison is case sensitive unless this option is selected.

Ignore warnings. Prevents the program from terminating if an error occurs. Because an error could corrupt the schematic database by updating a field with improper data, selecting this option is not recommended.

After you have verified the Update Field Contents configuration options, click on OK to exit. Launch the tool by double clicking on Update Field Contents. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appears during execution, there is usually no reason to examine this file.

Examining the Updated Field Contents

Go back to Draft and examine the changes to the schematic. Because you set part field 2 used for module values to invisible, you must use the edit command to examine individual parts. Figure 7-36 shows the entry in part field 2 of U2. Note that depending on how your schematic was annotated, the upper 74ALS573 address latch may have a different reference designator.

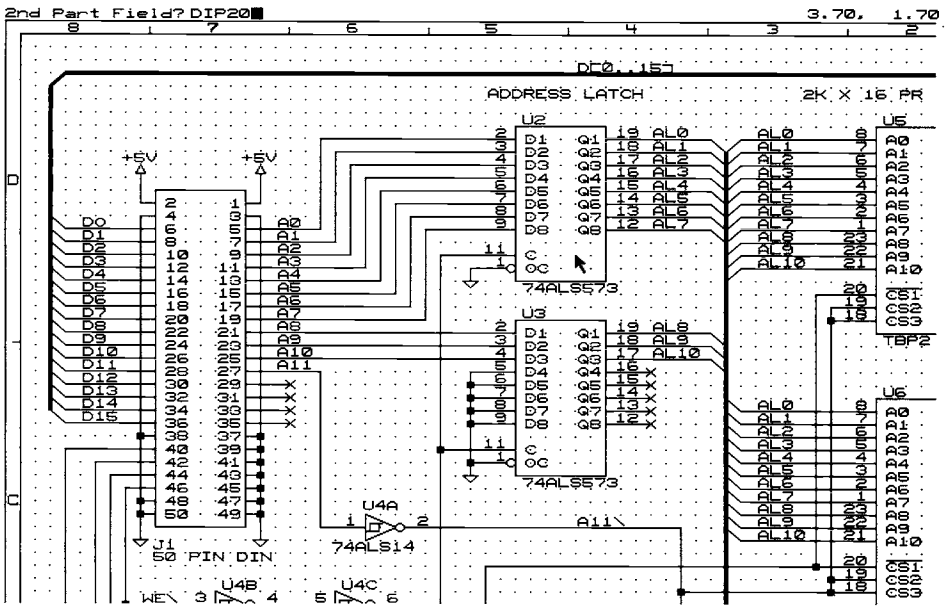


Figure 7-36 Using the Edit Command to Examine U2

Creating a Netlist with the Updated Module Values

Your TUTOR5A design now has updated module values. Configure and launch the Create Netlist tool to create a FutureNet netlist. Use the local configurations given in the FutureNet example in chapter 5. Use the name TUTOR5A.NET for the netlist. After completing the netlist, exit to DOS and examine the netlist using a text editor or the LIST.COM utility. Figure 7-37 shows the first section of the netlist.

Refer back to page 201 for an overview of the FutureNet format. Compare Figure 7-37 to the netlist shown in Figure 5-36 on page 202. Note that actual PCB layout module values are shown in Figure 7-37. When you ran Create Netlist as part of Tutorial 3, no Module Value Combine key field was defined. Recall that Create Netlist defaults to using part values for module port values, as shown in Figure 5-36.

Creating an update file that cross references part values and module port values and then running the Update Field Contents tool prior to running Create Netlist can save time when editing the netlist later on. Using this technique results in a significant time savings if the design has many parts with the same part value.

```
LIST      1                10-19-95 19:12 ♦ C:\ORCAD\TUTOR5A\TUTOR5A.NET
NETLIST,2
(DRAWING,ORCAD.NET,1-1
DATA,50,TMS32010 MEMORY EXPANSION BOARD
DATA,51,1004
DATA,52,1.0
DATA,54,  October 15, 1995
)
(SYM,1-1,1
DATA,2,C1
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
)
(SYM,1-1,2
DATA,2,C2
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
)
(SYM,1-1,3
DATA,2,C3
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
)
(SYM,1-1,4
DATA,2,C4
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
)
(SYM,1-1,5
DATA,2,C5
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
)
(SYM,1-1,6
DATA,2,C6
DATA,3,.1UF
DATA,4,CAP\40LS
DATA,23,1
DATA,23,2
Command▶     *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 7-37 TUTOR5A.NET Netlist with Updated Module Values

Using the Select Field View Tool

While you can use the Edit command to set visibility attributes for part fields associated with individual parts, the need sometimes arises for setting visibility attributes on a global basis throughout the entire design. You can use the Select Field View tool for this purpose. Select Field View sets the visibility attribute of the selected part field for all parts in the design. Note that only one part field visibility attribute can be set during each pass of the tool.

As part of the tutorial exercise, set the visibility attribute of the module value field (part field 2) to visible using the Select Field View tool, examine the results with Draft, and then set the visibility attribute back to invisible.

At the main SDT screen, click on Select Field View and then click on local configuration as shown in Figure 7-38. This brings up the configuration screen. Verify that your local configuration matches that shown in Figure 7-39 and make any required changes so that part field 2 will become visible.

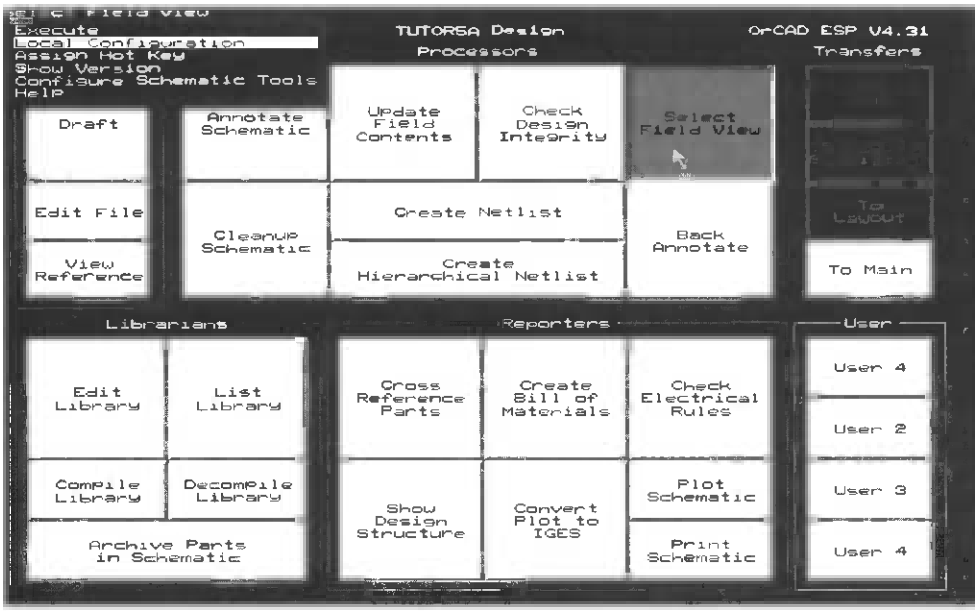


Figure 7-38 Selecting Field View Configuration

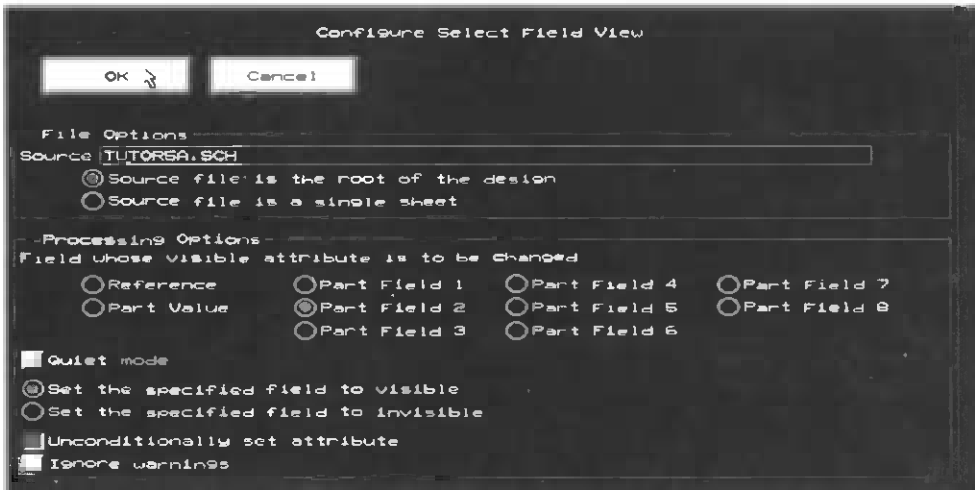


Figure 7-39 Field View Configuration Options

As with most other OrCAD tools, the Select Field View local configuration screen has the familiar file options section and additional processing options as explained on the next page:

Field whose visible attribute is to be changed. Selects the part field. Only one part field visibility attribute can be changed during each pass. Select part field 2, since this field is used for module values in the design.

Quiet mode. Suppresses display of text messages. Leave deactivated so that status messages will be displayed.

Field visibility options. Two options are available: set to visible or set to invisible. For the first pass, select "set to visible."

Unconditionally set attribute. Unless this option is selected, only part fields with nonblank entries are affected. Otherwise, blank part fields are left with the visibility attribute unchanged. Normally you do want Select Field View to unconditionally set all attributes, so select this option.

Ignore warnings. Prevents the program from terminating if an error occurs. Because an error could corrupt the schematic database by updating a field with improper data, selecting this option is not recommended.

After you have verified the Select Field View configuration options, click on OK to exit. Launch the tool by double clicking on Select Field View. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appears during execution, there is usually no reason to examine this file.

Next, go to Draft and examine the changes to the schematic. Part field 2, which contains module values, should be visible on all parts. Since you have not used the Edit command to adjust the locations, many of the module values will appear on top of other fields.

This part of the exercise illustrates the use of the Select Field View tool. In most cases module values are not displayed or printed, because they tend to clutter the schematic. Go back to the Select Field View local configuration. Set the visibility attribute for part field 2 to invisible and rerun the tool. Then go back to Draft and verify that the module values are invisible again.

Overview of the Back Annotation Process

Back Annotation refers to the process of updating reference designators that have changed during the PCB design phase. Reference designators change for two possible reasons:

- Gates are swapped on multiple parts per package devices. Swapping can occur between gates on the same IC or between gates on ICs of the same type.

- After parts are placed on the PCB layout, reference designators are renumbered in sequence. Neatly sequenced reference designators reduce the difficulty of manually locating components for assembly, test, and rework. In today's age of automated assembly and test, spatial sequencing of reference designators is no longer viewed as an absolute requirement.

OrCAD provides a back annotation capability via the appropriately named Back Annotate tool. Back Annotate uses data in a "was-is" file as a basis for updating reference designators. The was-is file can be created with an ASCII text editor. Each line consists of the original reference designator (the "was") and the new reference designator (the "is").

Using the Back Annotate Tool

Back Annotate is the last of the OrCAD tools covered in this tutorial exercise. Before using Back Annotate, you must create the was-is file for updating reference designators. Let's assume that the TUTOR5A netlist you previously created has been used to design a PCB. After the PCB design phase, you have a list of updated reference designators that must be back annotated into the schematic. Use the list shown in Figure 7-40 to create the was-is file. For convenience, the completed was-is file TUTOR5A.WAS is included in the TUTOR5A subdirectory on the disk supplied with this book.

```
LIST       1       7      10-19-95 22:02 ♦ C:\ORCAD\TUTOR5A\TUTOR5A.WAS
U1         U6
U2         U5
U3         U4
U4         U3
U5         U2
U6         U1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 7-40 TUTOR5A.WAS Was-Is File

Note that unlike the OrCAD update file shown in Figure 7-33, the was-is file format does not require single quotes around the reference designator text strings. The two reference designators on each line must be separated by one or more spaces.

At the main SDT screen, click on Back Annotate and then click on local configuration as shown in Figure 7-41. This brings up the configuration screen. Verify that your local configuration matches that shown in Figure 7-42 and make any required changes.

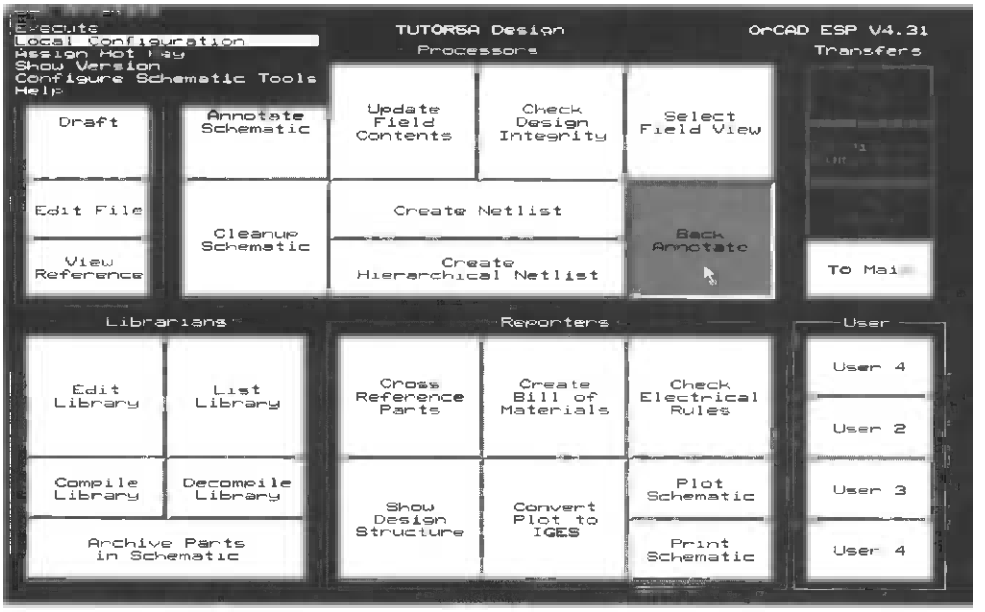


Figure 7-41 Selecting Back Annotate Configuration

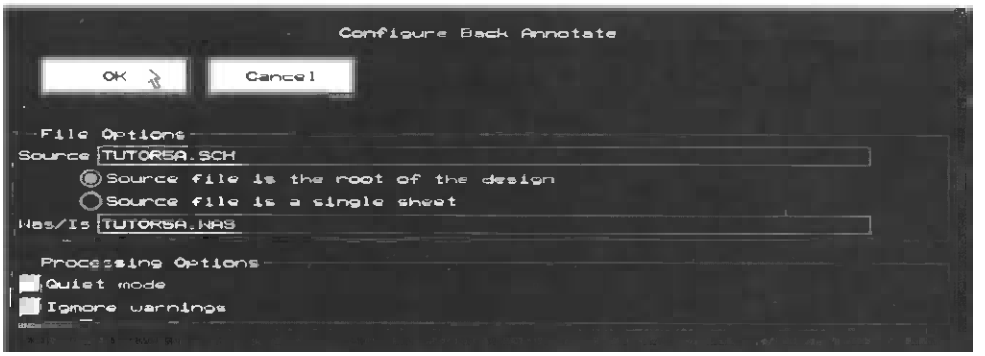


Figure 7-42 Back Annotate Configuration Options

Back Annotate has only minimal local configuration options. The file options section includes entry boxes for the name of the schematic design and the was-is filename. The two processing options function in an identical manner as with other OrCAD tools and do not require further discussion.

After you have verified the Back Annotate configuration options, click on OK to exit. Launch the tool by double clicking on Back Annotate. As with other OrCAD tools, status messages are written to the #ESP_OUT.TXT file. Unless a screen message warning of abnormal termination appears during execution, there is usually no reason to examine this file.

Next, go to Draft and examine the changes to the schematic. The reference designators for U1 through U5 should have been updated in accordance with the was-is file. The resultant changes for sheet one are shown in Figure 7-43. As mentioned before, your schematic may appear different depending on how reference designators were originally annotated.

Wrapping up the Third Session

You have now learned how to use all the OrCAD tools covered in this book. To wrap up the third session, run a final hard copy of your schematic, archive the parts in the schematic, and then backup all data onto a floppy disk.

Conclusion

Congratulations. You have endured through five increasingly difficult and challenging tutorials and learned the ins and outs of capturing schematics with OrCAD. Additional specialized information is provided in the following chapters:

Tutorial 6 in chapter 8 covers user buttons and custom macros. An understanding of these useful features will enhance your OrCAD productivity.

Tutorial 7 in chapter 9 covers using OrCAD to create netlists for SPICE circuit simulation. If you do not use SPICE, you can skip this specialized tutorial.

Tutorial 8 in chapter 10 covers specialized topics related to editing netlists for PCB design.

Tutorial 9 in chapter 11 covers editing bill of materials, including the use of a special sort utility to sort parts by part value

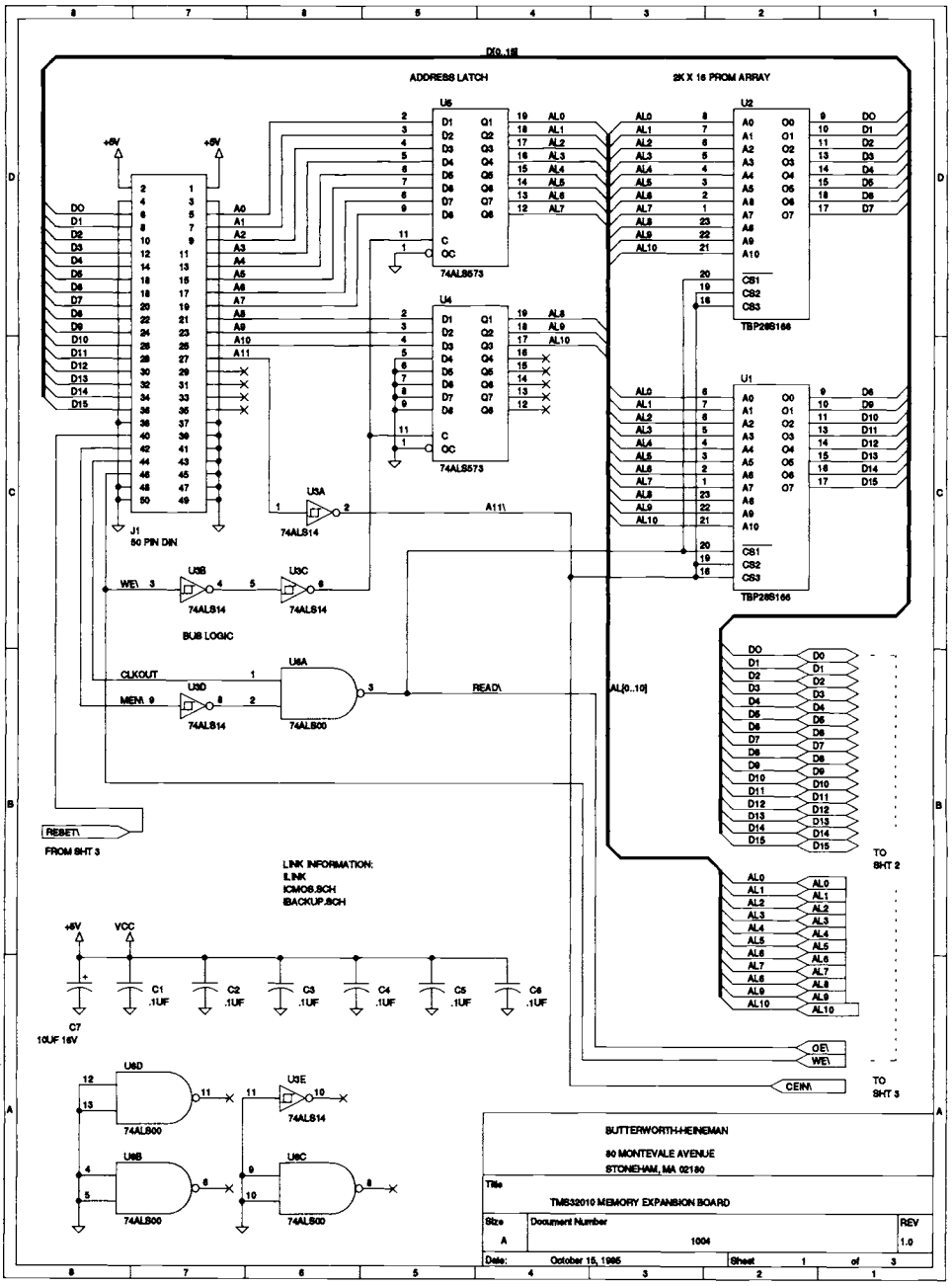


Figure 7-43 Updated Reference Designators on Sheet One

8

Tutorial 6 - User Buttons and Macros

The sixth tutorial is a short exercise that covers two OrCAD features that enhance productivity: user buttons and macro routines. OrCAD SDT 386+ provides four user buttons on the main screen. You can define these buttons to launch DOS commands or external programs. Typical uses include launching utility programs or a text editor. In the first part of this tutorial, you will learn how to define a user button to launch the LIST.COM utility supplied on disk with the book.

You have already learned to macro commands to speed schematic capture during the previous tutorials, via the CUSTOM.MAC macro. In the second part of this tutorial, you will learn how to define a new macro command and generate a new macro file.

Defining a User Button for the List Utility

At the main SDT screen, click on the User 1 button and then click on Modify as shown in Figure 8-1. This brings up the user button setup screen shown in Figure 8-2. The setup screen provides entry boxes for the following parameters:

Button Name. Two lines are provided for the button name. This name will appear on the screen when you have completed the user button definition. For this exercise, use List for the first line and Utility for the second line.

Help File Name. You can enter the name of an optional help file that provides additional information about the function of the user button. You can create the help file as an ASCII text file with a text editor. The text must be formatted into lines of 64 or less characters and the file is limited to 16 or less lines. The help file must reside in the ORCADESP directory.

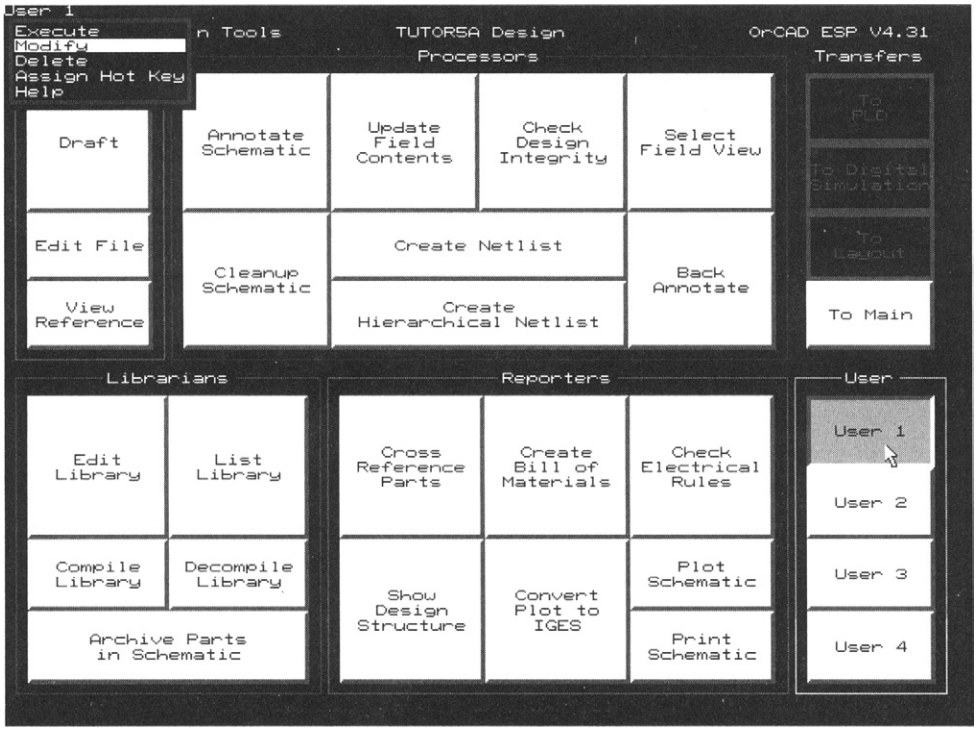


Figure 8-1 Defining a User Button

Command. This is the command string that you would type at the DOS prompt in order to launch the desired DOS command or program. Good practice is to include a drive letter and full path. For this exercise, the assumption is made that LIST.COM resides in the C:\UTILS subdirectory. A user button can also run multiple commands if you enter the name of a batch file.

Parameters. Used for DOS commands or programs that require parameters following the initial command. For example, if you wanted to pipe a directory listing through the MORE filter, the command entry would be DIR and the parameter entry would be |MORE.

After you have completed the setup, click on OK to exit. The name List Utility now appears on the top user button. Launch the utility by double clicking on the user button. This brings up the LIST.COM file selection screen shown in Figure 8-3. You can use the cursor controls to select any of the files in the current design directory. Then press the Enter key to display the contents of the highlighted file.

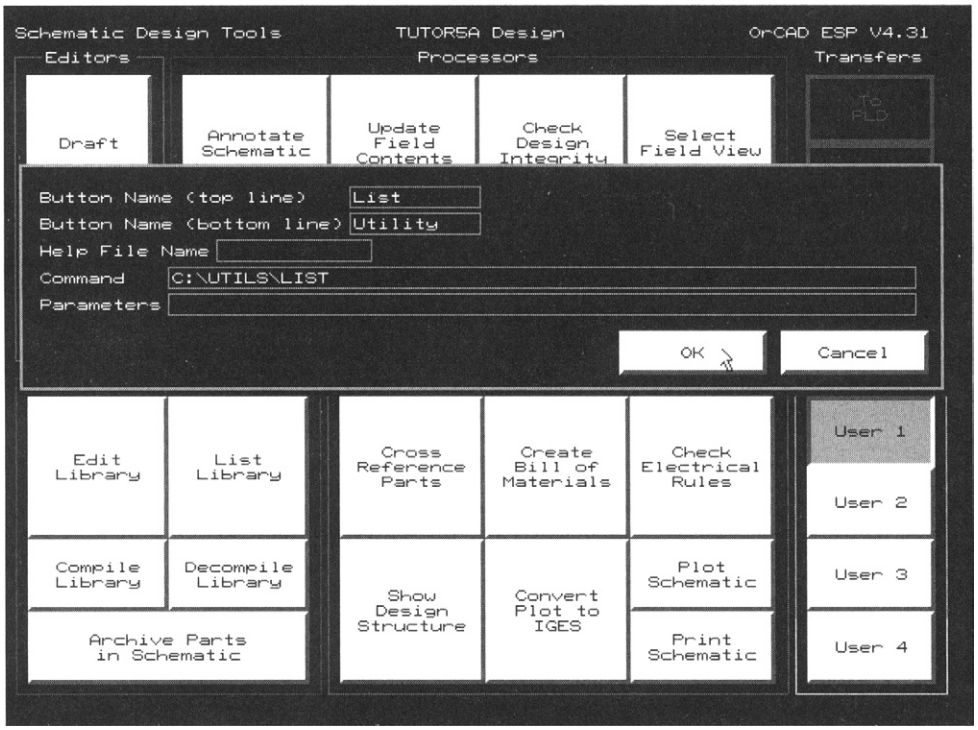


Figure 8-2 Entering Parameters for the User Button

You can use the menu shown in Figure 8-1 to modify or delete previously defined user buttons or to access the help file.

The user button feature has some limitations. Do not try to load memory resident programs. Also, the LIST.COM utility cannot access the #ESP_OUT.TXT file in the current design directory, presumably because the file has not been closed by OrCAD. An interesting note is that the author's text editor of choice, WordStar, has no difficulty accessing the #ESP_OUT.TXT file when launched from a user button.

Figure 8-3 LIST.COM File Selection Menu

Macro Command Overview

The Draft and Edit Library tools support macro commands. Macro commands improve productivity by reducing the number of keystrokes and mouse clicks required to carry out common, repetitive tasks. Macro commands are stored in a macro file on disk, generally with the filename extension .MAC. The macro file is an ASCII text file. You can use a text editor to create a macro file. Both Draft and Edit Library also provide a Macro command that allows you to define new macros and manage macro files.

Due to the nature of the tools, Draft and Edit Library generally use different macro commands, even though the syntax is identical. Each tool can load a specified macro file and execute an initial macro command on startup as set in the tool's configuration. The initial macro command typically sets the environment, such as visible grid dots and grid references.

Once a tool is launched, you can load new macro files containing different macro commands. You can only load one macro file at a time. A macro buffer in RAM memory stores the macro commands. The default size of 8192 bytes suffices for a large macro file with dozens of commands. OrCAD displays a warning message if the macro buffer fills up. If this occurs, you can increase the size of the buffer set in the tool's configuration.

Macro commands are ASCII text strings that consist of a name and the macro script. The macro script can contain keyboard commands, mouse movements and clicks, and text. A macro script can call other macro commands or even load another macro file. However, the user is cautioned against creating an excessive level of complexity because OrCAD does not provide any direct indication of which macro commands are active.

Macro Command Syntax

Use the List Utility user button to examine the CUSTOM.MAC macro file in one of your design subdirectories. The file contents appear in Figure 8-4.

```
LIST       1       40       10-25-95 19:13 ♦ C:\ORCAD\TUTOR5A\CUSTOM.MAC
CUSTOM.MAC CUSTOM MACRO

BEGIN WIRE
{F1}=PWB{}

PLACE JUNCTION
{F2}=PJP{ESC}{}

PLACE +5V POWER
{F3}=PPV{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}+5V{ENTER}TAP{ESC}{}

PLACE +12V POWER
{F4}=PPV{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}+12V{ENTER}TAP{ESC}{}

PLACE GROUND
{F5}=GGND SIGNAL{ENTER}{L}{ENTER}{ENTER}{ESC}{R}{}

GET RESISTOR
{F6}=GR{ENTER}{}

GET NON-POLARIZED CAPACITOR
{F7}=GCAP{ENTER}{}

GET POLARIZED CAPACITOR
{F8}=GCAPACITOR POL{ENTER}{}

GET DIODE
{F9}=GDIODE{ENTER}{}

GET NPN TRANSISTOR
{F10}=GNPN{ENTER}{}

INITIAL MACRO
   DRAG BUSES
   LEFT MOUSE BUTTON ENTER ON RELEASE
   SHOW COORDINATES
   GRID REFERENCES
   VISIBLE GRID DOTS
{\F10}={ENTER}{ESC}SD{ENTER}SL{ENTER}SX{ENTER}SGG{ENTER}SGU{ENTER}{}
→→→→→→→→

Command►     *** End-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 8-4 CUSTOM.MAC File Listing

The CUSTOM.MAC file was created using a text editor. A comment precedes each macro definition. The format for the macro definition is:

{Macro name} = Macro script {}

The macro name is a valid macro name key or key combination (see below) which must be enclosed in curly brackets. The equal sign (=) indicates the beginning of the macro script. The macro script consist of the actual commands run by the macro. The two curly brackets ({}) signify the end of the macro definition.

Several additional rules apply to macro files:

- Any line of text that does not start with a macro name enclosed in curly brackets is considered a comment line and is ignored. Curly brackets or the equal sign must not appear anywhere in a comment line.
- Macro scripts may contain only valid OrCAD commands, text, or filenames. A single character is used to represent the command from each menu level. The character is usually the first letter of the command or the highlighted letter shown on the menu command. For example PW represents Place Wire. Certain command keystrokes must be translated according to Table 8-1.
- Macro scripts may contain spaces, but not line breaks or paragraph breaks.
- Initial macros should begin with the script {ENTER}{ESC} to bypass the file selection box when Draft or Edit Library are launched.
- OrCAD provides macro support for three button mouse pointing devices. Refer to the OrCAD SDT Reference Guide for details.

Table 8-1 Macro Command Key Translations

Command Key	Macro Translation
Alt	\
Ctrl	^
Shift	SHIFT
Tab	{^I)
Shift Tab	{BACKTAB}
Enter	{ENTER}
Backspace	{RUBOUT}
Escape	{ESC}
Up	{U} or {UP}
Down	{D} or {DN}
Left	{L} or {LEFT}

Table 8-1 Macro Command Key Translations (Cont'd)

Command Key	Macro Translation
Right	{R} or {RIGHT}
Page Up	{PGUP}
Page Down	{PGDN}
Home	{HOME}
Ctrl-Home	{^HOME}
End	{END}
Insert	{INS}
Delete	{DEL}

Valid Macro Name Keys

OrCAD allows a large number of different macro name keys and key combinations:

- **F1 to F10 function keys**. Use the function keys for your primary macros. In addition to the function keys, you can use the <ALT>, <CTRL>, and <SHIFT> keys in combination with the ten function keys. Although this is an undocumented feature, OrCAD also appears to support the F11 and F12 function keys found on enhanced 104 key AT keyboards.
- **<ALT> in combination with A to Z and 0 to 9**.
- **<CTRL in combination with A to Z** (except <CTRL>C, <CTRL>H, and <CTRL>M are reserved).

Several additional key combinations are valid. Refer to the OrCAD SDT Reference Guide for details.

Most OrCAD users are engineers who also use many other computer programs. Unless you are lucky enough to possess a photographic memory, you are not likely to remember more than about a dozen macros for any given program. If you need to define more macros, use the function keys F1 to F10 in combination with the <ALT>, <CTRL>, and <SHIFT> keys and make a template that fits on the keyboard. You can purchase template blanks from Global Computer Supplies (1-800-GLOBAL).

Overview of the Macro Menu

Let's take a detailed look at the Macro menu. Select one of your designs from a previous tutorial, such as the TUTOR5A design. Launch Draft. Then click on the Macro command from the main menu. The Macro menu appears as shown in Figure 8-5.

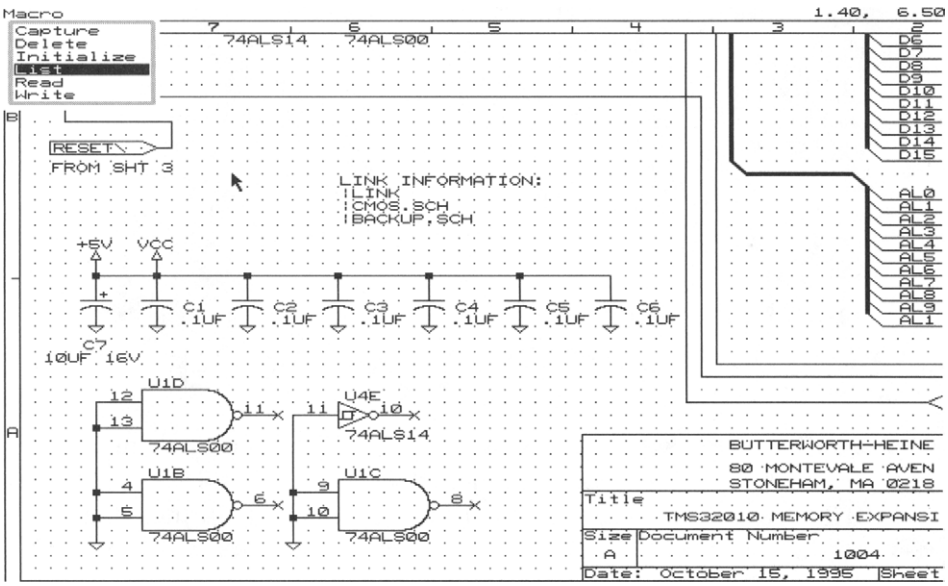


Figure 8-5 Macro Menu Options

The following options are available on the Macro menu:

Capture Used to capture (define) a macro command from within OrCAD. Prompts for the macro name followed by the sequence of commands that make up the macro. Any valid sequence of keystrokes and mouse movements and clicks is recorded. Press <CTRL> <END> to stop the capture command. The macro command is written to the macro buffer.

Delete Used to delete a macro command from the macro buffer. Prompts for the macro name key or key combination.

Initialize Clears the macro buffer.

List Lists the names of all macro commands available in the macro buffer as shown in Figure 8-6. The List option does not display any information about the macros.

Read Loads a new macro file into the buffer. Prompts for the macro filename.

Write Saves the macro buffer contents to a file. Prompts for the filename. The macro buffer only stores macro definitions, not comments. If you use Macro Read to read a macro file (or OrCAD loads an initial macro at startup) and then use Macro Write, any comments are lost.

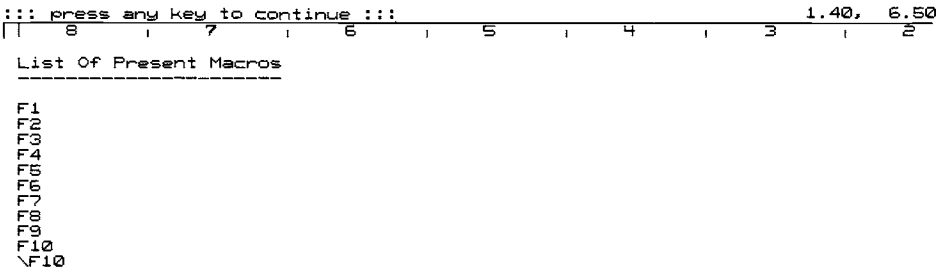


Figure 8-6 Output of Macro List Command

Capturing a New Macro

Assume that you want to modify your CUSTOM.MAC macro. F10 presently places an NPN transistor. With the advent of new technology, you now use N channel MOSFETs more often than NPN transistors, so you are going to modify F10 to place a MOSFET. Because OrCAD does not provide a means of editing macro commands from within Draft, you must capture a new definition for F10.

Click on the Macro Capture command. You are prompted for the macro name. Press **F10<ENTER>** (press the actual function key, do not enter F 1 0). The screen appears as shown in Figure 8-7.

Next, enter the macro script:

G MOSFET N <ENTER>

The screen echoes the actual command, Get? MOSFET N as you type it in (Figure 8-8). The part appears on the screen. Press **<CTRL><END>** to end the macro. Then press **<ESC>** to clear the screen. The new F10 macro has now been stored.

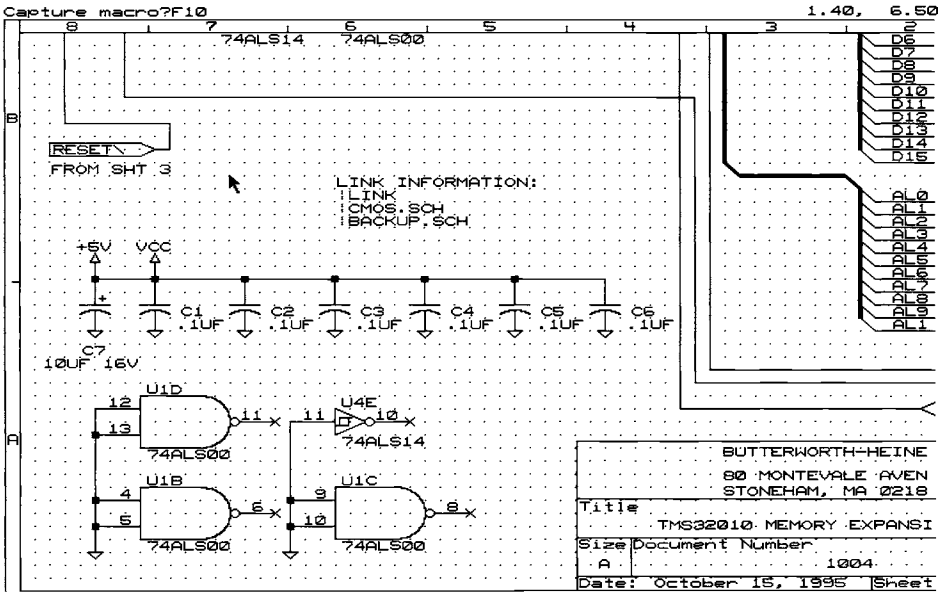


Figure 8-7 Macro Capture Name Entry

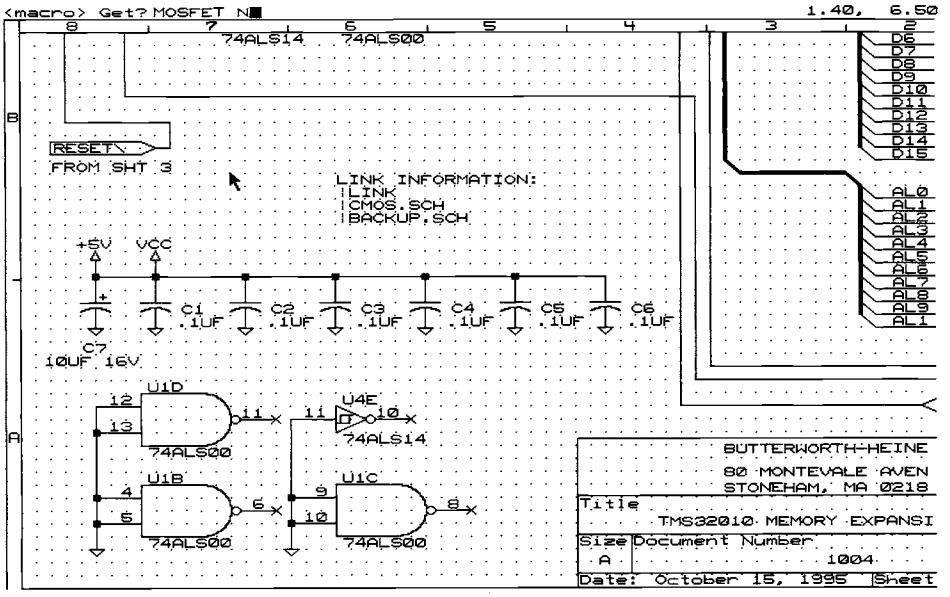


Figure 8-8 Macro Capture Script Entry

Save your revised set of macro commands to a file. Use the Macro Write command and filename NEW.MAC. Then exit Draft. Use the List Utility user button to examine the NEW.MAC file. The file contents should appear as shown in Figure 8-9

```
LIST       1       11       10-25-95 16:22 ♦ C:\ORCAD\TUTOR5A\NEW.MAC
{F1}=PWB{}
{F2}=PJP{ESC}{}
{F3}=PPV{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}+5V{ENTER}TAP{ESC}{}
{F4}=PPV{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}{RUBOUT}+12V{ENTER}TAP{ESC}{}
{F5}=GGND SIGNAL{ENTER}{L}{ENTER}{ENTER}{ESC}{R}{}
{F6}=GR{ENTER}{}
{F7}=GCAP{ENTER}{}
{F8}=GCAPACITOR POL{ENTER}{}
{F9}=GDIODE{ENTER}{}
{\F10}=SD{ENTER}SL{ENTER}SX{ENTER}SGG{ENTER}SGV{ENTER}{}
{F10}=GMOSFET N{ENTER}{}

Command▶     *** End-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 8-9 NEW.MAC File Listing

Note that the Macro Write command has deleted all comments included in the original CUSTOM.MAC macro file. This is one of the drawbacks to using the Draft Macro commands to directly create or edit macro files.

Macro Tips and Techniques

The following is a list of suggested tips and techniques for creating, editing, and debugging OrCAD macros:

- Use keyboard commands instead of mouse movements and clicks. Macros containing mouse movements and clicks consume an inordinate amount of buffer space and are very difficult to debug and to edit.
- Use the Macro Capture command to define new macro commands, write the results to a temporary file, and then use an ASCII text editor to add comments or to merge the new commands into existing macro files.
- Always keep a hard copy printout and backup copy on a floppy disk of all your macro files.

- For debugging, you can use the hard copy printout of your macro file as a script for manually entering the command sequence.
- Use <CTRL><BREAK> to abort a macro or stop a macro from running in an endless loop.
- Macro commands can call other macros. Embed the called macro name within curly brackets.
- A macro can pause and wait for user input. Insert the string {MACROBREAK} into the definition at the point where you want to pause.

9

Tutorial 7 - Creating SPICE Netlists

The seventh tutorial covers creating netlists for SPICE circuit simulation. SPICE stands for Simulation Program with Integrated Circuit Emphasis. The University Of California, Berkeley, developed the original SPICE program during the mid 1970s. Since then, many commercial versions have been developed. One of the most popular is PSpice, which is sold by MicroSim Corporation (Irvine, California). OrCAD supports PSpice with a customized component library and netlist format. With some limitations, the same library and netlist format is also compatible with most other SPICE programs.

SPICE evolved as a result of the problems associated with designing integrated circuits. By the 1970s, ICs had reached a state of miniaturization and complexity where breadboarding a prototype was no longer feasible. SPICE allowed the designers to simulate the circuit prior to running a wafer through a fab line and testing an actual part. While SPICE simulation had its roots in IC design, the concept is now widely used throughout the electronics industry.

In this tutorial, you will draw a schematic and then create a SPICE netlist for a small analog circuit, a magnetic pickup phonograph amplifier. The assumption is made that the reader has some level of familiarity with SPICE.

SPICE Netlist Overview

SPICE simulation programs, such as PSpice, require a circuit file as input. The circuit file consists of:

- **Title line and comments**. PSpice expects the first line to be the title line. Additional comment lines starting with an asterisk (*) character usually appear after the title line. The OrCAD netlist formatter automatically extracts information from the title block and creates the title line and additional comment lines.
- **Circuit device declarations list**. Each device declaration appears on a separate line. Device declarations include: device name (reference

designator), node names, and part/parameter value(s). OrCAD extracts data for circuit devices directly from the schematic. An example of a device declaration is:

```
R1   INPUT   0   47K ; COMMENTS
```

where R1 is the device name, INPUT and 0 are the two nodes names and 47K is the part value. PSpice accepts alphanumeric node names up to 131 characters in length. Ground is always assigned node 0. The semicolon is treated as an end-of-line character and any text after the semicolon is treated as a comment. Node names are established by labels or module ports. If neither is present, OrCAD generates an arbitrary node name.

- **SPICE commands**. These are program commands for the SPICE simulator. All SPICE commands start with a period (.) character. The exact syntax and SPICE-supported commands varies somewhat between the different commercial versions. OrCAD can extract SPICE commands, comments, and device declarations from OrCAD pipe commands placed on the schematic. For example:

```
|SPICE
|* ADDITIONAL COMMENT LINE
|.DC  LIN  I2  5MA  2MA  1MA  ; DC SWEEP STATEMENT
|KXFORM  L1  L2  .99  ; COUPLED INDUCTOR DECLARATION
```

  Pipe commands are special lines of text grouped together on the schematic. Each line of text must be preceded by the pipe character | (broken vertical bar on PC keyboard). The |SPICE statement on the first line instructs OrCAD to place the remaining lines of text at the top of the netlist. The pipe characters are stripped off the text placed on the netlist.

The order of comments, device declarations, and commands in the circuit file is generally not important, except that the first line must be the title and the .END statement must appear on the last line.

SPICE programs use the term "circuit input file." OrCAD uses the term "SPICE netlist." The term "SPICE netlist" is used in a very broad sense, because unlike a netlist for PCB design, no actual listing of circuit nets is created.

Most of the PC based SPICE circuit simulation programs, such as PSpice, offer schematic capture. So why use OrCAD? If simulating a circuit is the only requirement, using the simulation program's built-in schematic capture capability may be the best approach. However, if PCB design is a requirement after the simulation is completed, using OrCAD from the start is more effective.

PSpice Parts Library

OrCAD provides a customized parts library, PSPICE.LIB, for use with PSpice. To avoid problems, you must use parts exclusively from this library. PSpice, and other SPICE versions as well, have certain special requirements and considerations as far as parts:

- **Special reference designators**. Capacitors, inductors, resistors, diodes, and bipolar transistors use standard reference designators. JFETS use the letter J and MOSFETS use M. Various controlled current and voltage sources use the letters E, F, G, and H. Transformers are modeled as coupled inductors; SPICE uses T to represent a transmission line.
- **Device values and units**. SPICE uses units of volts, amperes, seconds, meters, ohms, henries, and farads with the multiplier suffixes in Table 9-1.

Table 9-1 SPICE Multiplier Suffixes

SPICE SUFFIX	UNIT	MULTIPLIER
F	femto	10^{-15}
P	pico	10^{-12}
N	nano	10^{-9}
U	micro	10^{-6}
M	milli	10^{-3}
K	kilo	10^{3}
MEG	mega	10^{6}
G	giga	10^{9}
T	tera	10^{12}
MIL	.001 inch	2.54×10^{-6}

Note that a 10 megohm resistor would become 10MEG not 10M.

- **Device models**. Other than simple passive devices such as capacitors, inductors, and resistors, most semiconductor devices are based on models with specified parameters. PSpice comes with its own parts library that contains device models used during simulation (not to be confused with OrCAD's PSPICE.LIB library). You can only use parts for which models exist in the simulator's library. All the parts in OrCAD's PSPICE.LIB library are supported. Parts in other OrCAD libraries are most likely not supported.

- **Device pin names.** SPICE device declarations require nodes to be in a specified sequence. For example, the sequence for bipolar transistor device nodes is collector, base, and emitter. OrCAD's SPICE netlist formatter generates the node sequence based on the numerical order of the pin names. Bipolar transistor pins are named 1, 2, and 3 corresponding to collector, base, and emitter. All the parts in the PSPICE.LIB library have pin names with the proper numerical order for SPICE.

Circuit Considerations for SPICE Simulation

A detailed discussion of circuit considerations for SPICE simulation is beyond the scope of this book. Some basic considerations relevant to this tutorial and OrCAD are listed below. For additional information, the reader should consult the reference given at the end of this chapter.

- **Circuit simplification**. Circuits for SPICE simulation can generally be simplified. For example, voltage sources used for power supplies are ideal, with zero source impedance. In most cases, you can delete bypass capacitors and power supply circuitry without affecting simulation results.
- **Ideal circuit devices**. Simple passive circuit devices such as resistors and capacitors are modeled as ideal devices without parasitic elements. Parameters such as resistor wattage or capacitor polarity and rated voltage are not applicable to the ideal devices modeled in SPICE.
- **Initial conditions**. In addition to part value, initial conditions can be specified for capacitor voltage and inductor current. The initial condition statement follows the device value. For example:

1U IC=5V	(1 microfarad capacitor charged to 5 volts)
5MH IC=1A	(5 millihenry inductor with 1 amp current)

- **Ground**. The OrCAD PSPICE.LIB library includes a special ground part. Note that SPICE considers ground as a node, not as a circuit device. SPICE requires the name "0" for the ground node. OrCAD's SPICE netlist formatter correctly interprets the standard OrCAD ground symbols, even though these use GND for the pin name.
- **Floating nodes prohibited**. SPICE does not allow isolated terminals or floating nodes. All devices must have at least two terminal nodes. Every terminal must be connected to another terminal or ground. Every SPICE circuit must have a ground node and every other node in the circuit must have a DC path to ground. If necessary, high value resistors must be added in

parallel with circuit devices such as capacitors to meet the DC path to ground requirement.

- **Power objects**. You can use OrCAD power objects if the value is correctly named. OrCAD's SPICE netlist formatter translates power objects into SPICE voltage source device declarations. For example, OrCAD does the following translation:

```
VCC  +5V              (OrCAD power object value)
VCC  VCC  0  +5V      (SPICE voltage source declaration)
```

  Note that the name of the voltage source, VCC, is also used as the name for the first node. OrCAD automatically assigns the second node to ground. The first letter of the name must be V, because this is the prefix used by SPICE for voltage sources. The "+" and "V" characters associated with the +5V value are optional.

- **Inputs and outputs**. SPICE circuits must be completely self-contained. Inputs are excited via a voltage or current source with an associated waveform definition. Outputs are measured across a load impedance, such as R9 in Figure 9-1. The OUTPUT module port is merely used for convenience, to establish the node name.

Starting the SPICE Netlist Tutorial

Use the circuit shown in Figure 9-1 as the model for this tutorial. Start by creating a new design called TUTOR7, based on the standard template. Use the OrCAD Design Management tools as for the previous tutorials. Then select TUTOR7 and launch OrCAD SDT.

Configuring SDT for the PSPICE.LIB Library

The next step is to add the PSPICE.LIB library to the SDT configuration. Click on Draft and then click on Configure SDT. Scroll down to the library options section of the configuration screen shown in Figure 9-2.

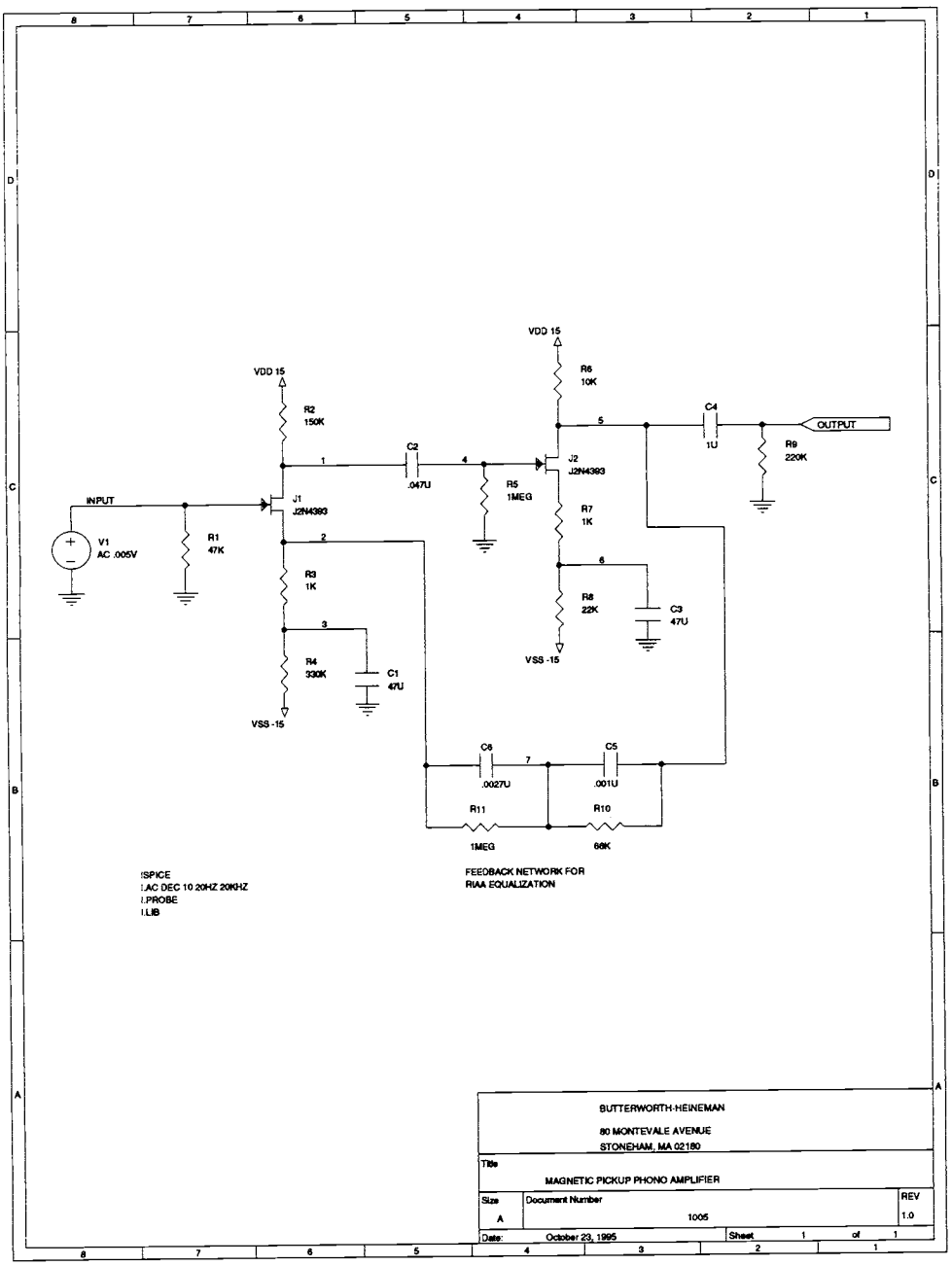


Figure 9-1 Amplifier Circuit for SPICE Simulation

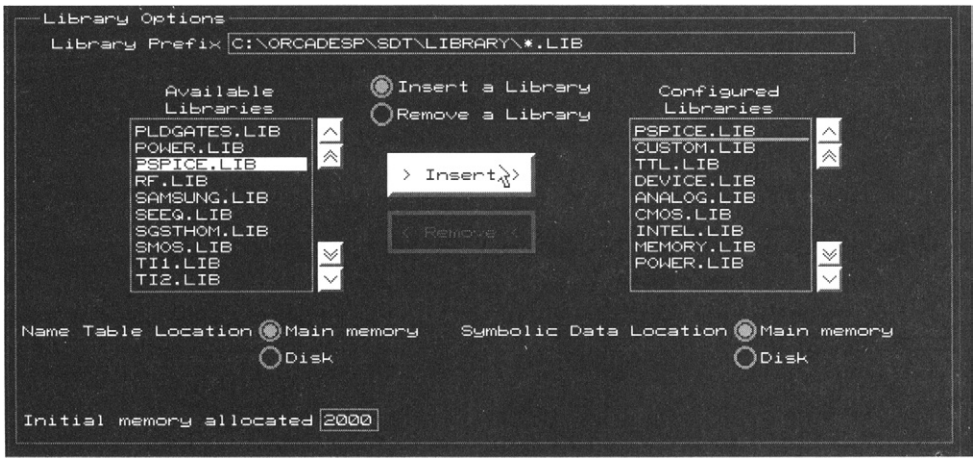


Figure 9-2 Inserting the PSPICE.LIB Library

Click on the Insert a Library button. Move the green bar on the configured library scroll box to the very top by clicking the left mouse button near the top of the scroll box. Recall that the green bar indicates the position at which the next library will be inserted.

Scroll down the list of available libraries and click on the PSPICE.LIB library. Click on the Insert button to insert the new library. Then click on OK at the top of the SDT configuration screen.

PSPICE.LIB must be the very first library on the list of configured libraries. You might even consider removing all other libraries, because the parts in other libraries are not generally compatible with SPICE.

OrCAD has an idiosyncrasy in the way that parts are retrieved from libraries. Recall that the Get command brings up a list of libraries and you can select a library and scroll through the list of parts in the selected library. However, you are only scrolling through part names. When you click on the selected part, Draft scans the libraries in sequence and retrieves the first occurrence of the part name. If a given part name occurs in two libraries, the part is retrieved from the first library regardless of where you originally selected the part. Because the PSPICE.LIB library has SPICE versions of parts that may appear in other libraries, PSPICE.LIB **must be** the first library.

Parts retrieved from a library are not stored in the design database, only the part occurrence is stored. The process of searching through the libraries for parts is repeated every time a design is loaded into Draft. If the library order is changed, problems can occur the next time the design is loaded.

Drafting the Schematic

Complete the schematic shown in Figure 9-1. You must retrieve all the parts from the PSPICE.LIB library. Up to now, you have probably used the CUSTOM.MAC macro commands to place common objects and discrete parts. Not all of the macros will work since some of the SPICE part names are different. If you plan on creating many designs requiring SPICE simulation, you may want to write a special SPICE macro.

Listed below are hints on the parts and objects used in the tutorial design:

- **Wires, junctions, and power objects**: use the familiar CUSTOM.MAC macro commands as explained in chapter 3. You can use either F3 or F4 for power objects and then edit the name.
- **Ground**: use GND from PSPICE.LIB.
- **Capacitors**: use macro F7 since the SPICE capacitor has the same name as the standard part.
- **Resistors**: use RES from PSPICE.LIB.
- **J2N4393**: use JFET N from PSPICE.LIB. Edit the name. Note that SPICE requires the reference designator J for JFETS. J2N4393 is the PSpice name for a 2N4393 JFET.
- **Voltage source V1**: use V SRC from PSPICE.LIB. Edit the name.

Additional objects that appear on the schematic include the node labels 1 through 7, the OUTPUT module port, text comments, and the SPICE pipe commands.

Because the schematic has few components, manually edit the reference designator. This will assure that your schematic matches the tutorial model and will facilitate checking the netlist.

After completing the schematic, run the Check Design Integrity tool and correct any errors. Run Print Schematic to generate hard copy. Backup your design subdirectory to a floppy disk before proceeding.

Creating the SPICE Netlist

The first step is configuring the IFORM netlist formatter for SPICE. At the main SDT screen click on Create Netlist and then click on Configure IFORM. Modify your IFORM configuration to match that shown in Figure 9-3.

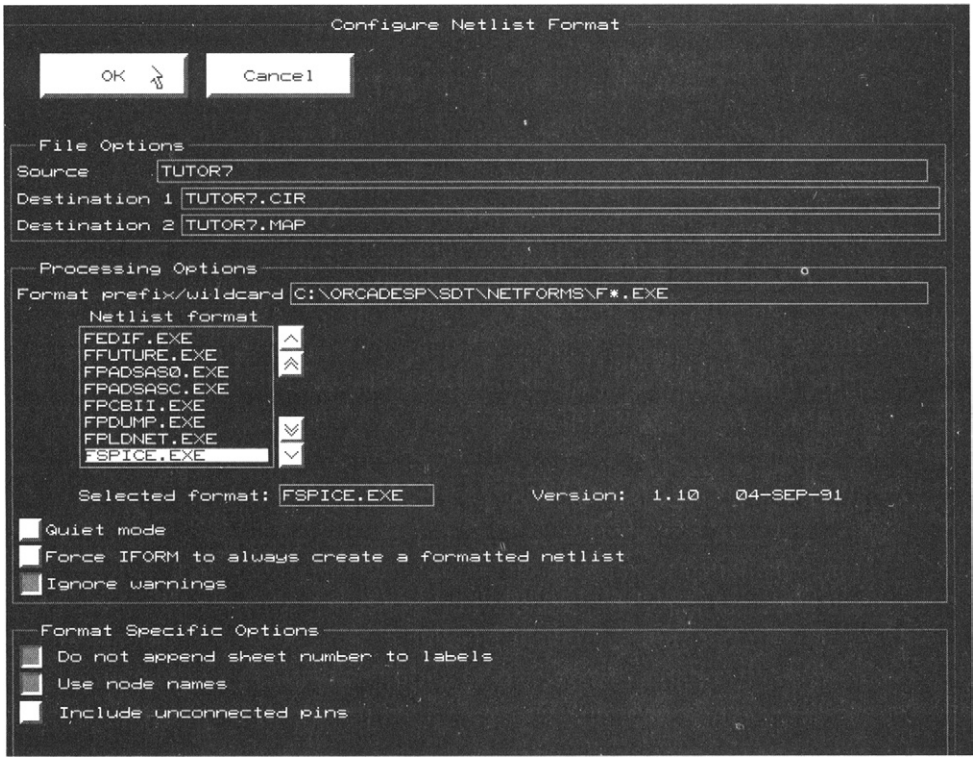


Figure 9-3 IFORM Configuration for SPICE

Two destinations appear in the File Options section. The formatter writes netlist data to the first destination. A special "map" file that cross-references labels with node names is written to the second destination. PSpice expects the circuit file (netlist) to have a .CIR filename extension.

When you select the FSPICE (flat SPICE) netlist format, several SPICE-specific processing options appear:

Do not append a sheet number to labels. A sheet number is normally appended to labels, which in turn are used for node names.

Use node names. Causes label and module port names to be used as node names. Some SPICE versions do not accept alphanumeric node nodes. If this option is not selected, the netlist formatter assigns arbitrary numerical node names starting with 10000. The map file cross-references the labels and module ports on the schematic to the assigned node names.

Include unconnected pins. SPICE never allows unconnected (floating) pins. This option will cause SPICE to generate an error message if unconnected pins appear on the schematic.

After you have verified the IFORM configuration options, click on OK to exit. Then run the Create Netlist tool.

If you installed the LIST.COM utility as a user button in the preceding tutorial, click on the utility and examine the TUTOR7.CIR netlist file. The file should appear similar to that shown in Figure 9-4. Statements may appear in a different order, depending on the order in which you placed parts onto the schematic.

```
LIST       1              10-23-95 12:53 ♦ C:\ORCAD\TUTOR7\TUTOR7.CIR
* MAGNETIC PICKUP PHONO AMPLIFIER                Revised:   October 23, 1995
* 1005                                           Revision: 1.0
* BUTTERWORTH-HEINEMAN
* 80 MONTEVALE AVENUE
* STONEHAM, MA 02180
*
*
.AC DEC 10 20HZ 20KHZ
.PROBE
.LIB
VDD VDD 0 15
VSS VSS 0 -15
C5 5 7 .001U
C6 7 2 .0027U
C2 1 4 .047U
R6 VDD 5 10K
R2 VDD 1 150K
R3 2 3 1K
R7 10004 6 1K
R5 4 0 1MEG
R11 2 7 1MEG
C4 5 OUTPUT 1U
R9 OUTPUT 0 220K
Command▶     *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 9-4 TUTOR7.CIR SPICE Netlist

PSpice runs the netlist shown in Figure 9-4 without any requirement for editing. Figure 9-5 shows a Bode plot of the amplifier gain and phase shift as a function of frequency over the audio range.

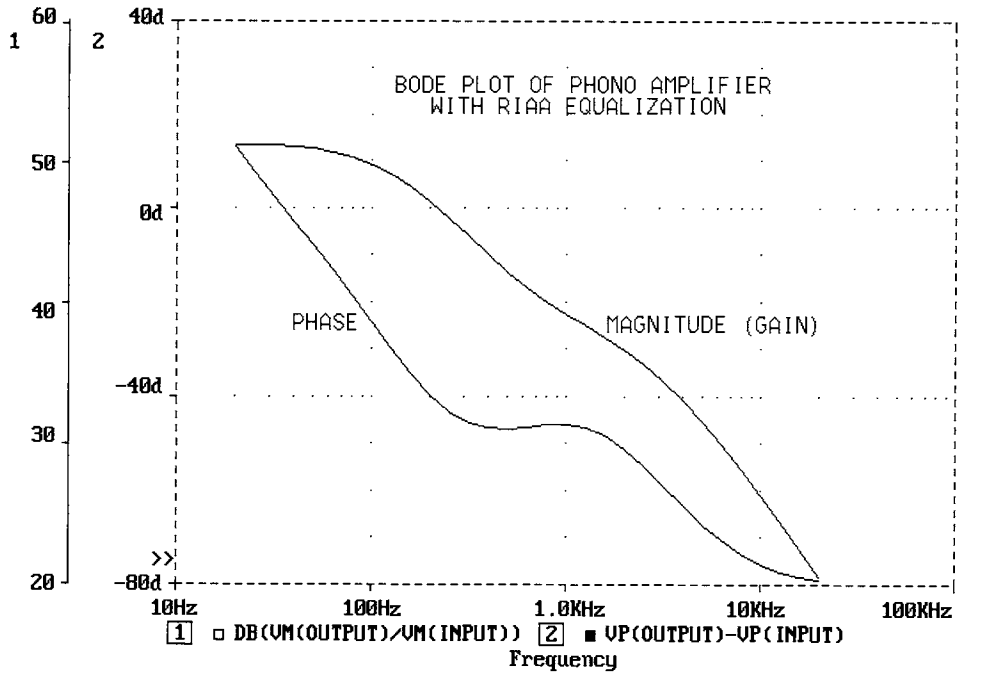


Figure 9-5 Bode Plot from TUTOR7.CIR SPICE Simulation

Conclusion

An excellent source for readers who are interested in learning more about SPICE, and PSpice in particular, is:

> Tuinega, P. *SPICE: A Guide to Circuit Simulation and Analysis Using PSpice*, 2d ed. Eaglewood Cliffs: Prentice Hall, 1992.

The SPICE guide by Tuinega is available with a student version of PSpice on disk which is suitable for running the tutorial circuit.

10

Tutorial 8 - Netlists for PCB Design

Netlists were introduced in chapter 5 with a discussion of netlist concepts and an exercise using the Create Netlist tool to generate a FutureNet netlist. Additional material was covered as part of the key fields discussion in chapter 7, including the use of the Update Field Contents and Back Annotate tools.

Recall from the discussion in chapter 5 that two primary issues arise when preparing a netlist for PCB design:

- PCB design software requires module values to identify the physical layout of parts on the schematic.
- Pin names and the pin sequence used by OrCAD may differ from those used by the PCB design software to represent the physical layout of actual devices. Problems are most prevalent with discrete parts, especially transistors.

The author's experience has shown that using a text editor to edit the netlist is usually the most efficient approach to adding module values and correcting pin names.

In this tutorial, you will prepare a netlist suitable for input into PADS-Work, a leading PCB design software package used by the author. Module values, pin numbers, and pin arrangements are based on the PADS-Work layout parts library. While PADS-Work imposes some specific requirements, the techniques you will learn are applicable to most PCB design software.

The model for this tutorial is the TUTOR2 design you created as part of the exercise in chapter 3. Recall that TUTOR2 is the four sheet hierarchical schematic shown in Figures 1-18A through 1-18D in chapter 1.

Starting the Tutorial Exercise

Start OrCAD. Use Design Management Tools to copy the TUTOR2 design you previously created to a new design. Use the name TUTOR8 for the new design.

Select the TUTOR8 design and then launch SDT. Use the Check Design Integrity tool to check the design for errors. Before launching the tool, verify that the Check Design Integrity local configuration matches that used in chapter 7. After you have run Check Design Integrity, examine the #ESP_OUT.TXT file for errors. If required, correct any errors and rerun Check Design Integrity.

After you have checked the TUTOR8 design for errors, run the Create Bill of Materials tool and the Print Schematic tool. Keep the hard copy of the schematic handy for reference. Exit to DOS and use your text editor to print out the TUTOR8.BOM bill of materials listing. Verify that your printout matches the listing shown in Figure 10-1. Use Draft to make any required corrections to the schematic. Backup all design files to a floppy disk before proceeding.

```
LIST       1       30      10-30-95 21:38 ♦ C:\ORCAD\TUTOR8\TUTOR8.BOM
HIERARCHICAL DESIGN EXAMPLE                    Revised:    August  8, 1995
1001                                           Revision: 1.0
Bill Of Materials              October   30, 1995     21:38:11     Page    1

 Item  Quantity  Reference                    Part
-------------------------------------------------------------

    1         1  C1                          22UF 35V
    2         2  C2,C4                       1UF 16V TANT
    3         1  C3                          .01UF
    4         1  D1                          1N4007
    5         1  PL1                         +12V
    6         1  PL2                         GROUND
    7         1  PL3                         VACUUM
    8         1  PL4                         TACH OUTPUT
    9         1  PL5                         COIL OUTPUT
   10         2  Q1,Q2                       2N4401
   11         1  Q3                          FUJI ET365
   12         1  R1                          3.3K
   13         1  R2                          100K
   14         1  R3                          39K
   15         3  R4,R7,R9                    2.2K
   16         2  R5,R6                       10K
   17         1  R8                          1K .25W
   18         1  R10                         470 1W
   19         1  U1                          HA-640
   20         1  U2                          MIC2951
   21         1  U3                          PIC16C71
   22         1  Y1                          8.00 MHZ
   23         1  ZNR1                        ERZ-CF1MK270

Command▶     *** End-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 10-1 TUTOR8.BOM Bill of Materials Listing

Creating the Netlist

Configure and launch the Create Netlist tool to create a FutureNet netlist. Use the local configurations given in the FutureNet example in chapter 5. Use the name TUTOR8.NET for the netlist. After completing the netlist, exit to DOS and use your text editor to print out the netlist. Figure 10-2 shows a section of the

TUTOR8.NET parts data. Figure 10-3 shows a section of the signal data. Verify that your netlist printout is similar to the listings shown in Figures 10-2 and 10-3. Some minor differences may exist depending on the order in which you placed parts on the schematic. This will affect the sequence in which parts are listed and the assigned net names. Part names, pin names, and signal interconnections should not be affected.

```
LIST       1              10-30-95 21:39 ♦ C:\ORCAD\TUTOR8\TUTOR8.NET
NETLIST,2
(DRAWING,ORCAD.NET,1-1
DATA,50,HIERARCHICAL DESIGN EXAMPLE
DATA,51,1001
DATA,52,1.0
DATA,54,   August  8, 1995
)
(SYM,1-1,1
DATA,2,PL1
DATA,3,+12V
DATA,4,+12V
DATA,23,1
)
(SYM,1-1,2
DATA,2,C3
DATA,3,.01UF
DATA,4,.01UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,3
DATA,2,R2
DATA,3,100K
DATA,4,100K
DATA,23,1
DATA,23,2
)
(SYM,1-1,4
DATA,2,R5
DATA,3,10K
DATA,4,10K
DATA,23,A
DATA,23,B
DATA,23,WIPER
)
(SYM,1-1,5
DATA,2,R6
DATA,3,10K
DATA,4,10K
DATA,23,A
DATA,23,B
DATA,23,WIPER
)
(SYM,1-1,6
DATA,2,R8
DATA,3,1K .25W
DATA,4,1K .25W
DATA,23,1
Command▶    *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 10-2 TUTOR8.NET Netlist Parts Section

Please refer back to the FutureNet netlist format explanation on page 202 in chapter 5. Part values follow FutureNet statements of the form DATA,3. Module values follow FutureNet statements of the form DATA,4. Recall that unless you modify the Create Netlist Module Value Combine key field, OrCAD defaults to using part values for module values. This is apparent in the listing shown in Figure 10-2, since the part values and module values are identical. These dummy module values must be replaced with the names of the actual layout parts used in the PCB design software.

```
LIST      228                10-30-95 21:39 ♦ C:\ORCAD\TUTOR8\TUTOR8.NET
(SIG,,***000001,1-1,5,***000001
PIN,1-1,7,D1,23,ANODE
PIN,1-1,1,PL1,23,1
)
(SIG,,***000002,1-1,5,***000002
PIN,1-1,25,U2,23,7
PIN,1-1,25,U2,23,6
)
(SIG,,***000003,1-1,5,***000003
PIN,1-1,28,PL3,23,1
PIN,1-1,16,R1,23,2
PIN,1-1,3,R2,23,1
)
(SIG,,***000004,1-1,5,***000004
PIN,1-1,4,R5,23,WIPER
PIN,1-1,26,U3,22,17
)
(SIG,,***000005,1-1,5,***000005
PIN,1-1,26,U3,22,18
PIN,1-1,5,R6,23,WIPER
)
(SIG,,***000006,1-1,5,***000006
PIN,1-1,19,Y1,23,2
PIN,1-1,26,U3,22,16
)
(SIG,,***000007,1-1,5,***000007
PIN,1-1,19,Y1,23,1
PIN,1-1,26,U3,22,15
)
(SIG,,***000008,1-1,5,***000008
PIN,1-1,6,R8,23,2
PIN,1-1,27,PL4,23,1
PIN,1-1,14,Q1,23,COLLECTOR
)
(SIG,,***000009,1-1,5,***000009
PIN,1-1,11,R7,23,2
PIN,1-1,14,Q1,23,BASE
)
(SIG,,***000010,1-1,5,***000010
PIN,1-1,20,PL5,23,1
PIN,1-1,22,Q3,23,COLLECTOR
)
(SIG,,***000011,1-1,5,***000011
PIN,1-1,18,R10,23,2
PIN,1-1,22,Q3,23,BASE
PIN,1-1,15,Q2,23,COLLECTOR
)
(SIG,,***000012,1-1,5,***000012
Command►                                  Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 10-3 TUTOR8.NET Netlist Signal Section

The FutureNet specific local configuration for Create Netlist was set to output pin numbers instead of pin names. Most PCB design software expects pin numbers. Pin numbers follow FutureNet statements of the form DATA,23 in the parts section. Pin numbers are also given as the last data on FutureNet statements of the form PIN,1-1 in the signal section. Note that diode D1, transistors Q1-3, and trimpots R5-6 all have pin names instead of pin numbers. These parts, and many others in the DEVICE.LIB library are defined as graphics parts without pin numbers. If Create Netlist encounters a graphics part without pin numbers, the pin names are used instead. The PCB design software uses layout parts defined with pin numbers, so any alphanumeric pin names such as ANODE or BASE must be replaced with appropriate pin numbers.

Stripping out Part Values

The TUTOR8.NET netlist contains FutureNet data statements with part values. Part value information is generally not used in the PCB design process. The only exception occurs with certain consumer electronics applications where part values appear along with the reference designators on the component legend silk screen. Finding space for part value legends on today's dense circuit boards is almost impossible and this practice has become rare.

In most cases, it is convenient to strip out part values from the netlist. A simple utility for this purpose, NETSTRIP.EXE, is included on the disk supplied with this book. Exit to DOS and load the utility onto your hard drive. The ORCADESP directory is the suggested location for OrCAD related utilities. If you use the ORCADESP directory, make sure that you add it to the PATH statement in your AUTOEXEC.BAT file and reboot the computer.

NETSTRIP.EXE strips out all lines in the netlist that contain FutureNet statements beginning with DATA,3. This will remove all part values from the parts section of the netlist. Assuming that you have loaded NETSTRIP.EXE into your ORCADESP directory and added the directory to your path, run the utility by typing:

CD\ORCAD\TUTOR8<ENTER>

NETSTRIP<ENTER>

NETSTRIP prompts for the name of the netlist input file and output file. The input file is read but not modified. Enter TUTOR8.NET for the input filename and TUTOR8A.NET for the output filename. NETSTRIP exits back to DOS after completion.

Use your text editor to print out the stripped netlist TUTOR8A.NET. Figure 10-4 shows a section of the parts data. Verify that your netlist printout is similar to the listings shown in Figure 10-4. Note that all part value statements are stripped out. The module value statements are not affected.

Editing Module Values

The next step is to edit the module values. The dummy module values in the netlist must be replaced with the names of the actual layout parts used in the PCB design software. Layout part names will depend on the PCB design software parts libraries. PCB design software packages generally come with a standard library of common parts. The user then defines additional parts in a custom library. The situation exactly parallels OrCAD.

The usual procedure involves taking the bill of materials for the schematic design and writing the appropriate module values next to each part description.

For the purposes of this tutorial use Table 10-1 which cross-references the TUTOR8 bill of materials with appropriate module values taken from the author's PADS-Work part libraries.

```
LIST       1                10-30-95 21:47 * C:\ORCAD\TUTOR8\TUTOR8A.NET
NETLIST 2
(DRAWING ORCAD.NET 1-1
DATA,50,HIERARCHICAL DESIGN EXAMPLE
DATA,51,1001
DATA,52,1.0
DATA,54,   August  8, 1995
)
(SYM,1-1,1
DATA,2,PL1
DATA,3,+12V
DATA,23,1
)
(SYM,1-1,2
DATA,2,C3
DATA,3,.01UF
DATA,23,1
DATA,23,2
)
(SYM,1-1,3
DATA,2,R2
DATA,3,100K
DATA,23,1
DATA,23,2
)
(SYM,1-1,4
DATA,2,R5
DATA,3,10K
DATA,23,A
DATA,23,B
DATA,23,WIPER
)
(SYM,1-1,5
DATA,2,R6
DATA,3,10K
DATA,23,A
DATA,23,B
DATA,23,WIPER
)
(SYM,1-1,6
DATA,2,R8
DATA,3,1K .25W
DATA,23,1
DATA,23,2
)
(SYM,1-1,7
DATA,2,D1
DATA,3,1N4007
DATA,23,ANODE
Command>    *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 10-4 TUTOR8A.NET Netlist Parts Section

Table 10-1 Module Value Cross Reference

Reference Designator	Part Value	Module Value
C1	22UF 35V	ECAP\30SQ\SMD
C2,C4	1UF 16V TANT	ECAP\3216\SMD
C3	.01UF	C\0805\SMD
D1	1N4007	D\40LS
PL1	+12V	CONN\08PAD
PL2	GROUND	CONN\08PAD
PL3	VACUUM	CONN\08PAD
PL4	TACH OUTPUT	CONN\08PAD
PL5	COIL OUTPUT	CONN\08PAD
Q1,Q2	2N4401	TO-92A
Q3	FUJI ET365	TO-220\UPA
R1	3.3K	R\0805\SMD
R2	100K	R\0805\SMD

Table 10-1 Module Value Cross Reference (Cont'd)

Reference Designator	Part Value	Module Value
R3	39K	R\0805\SMD
R4,R7,R9	2.2K	R\0805\SMD
R5,R6	10K	VRES\T\ADJ\6MM
R8	1K .25W	R\1210\SMD
R10	470 1W	R\2512\SMD
U1	HA-640	HALL\HA640\1
U2	MIC2951	DIP8\SO
U3	PIC16C71	DIP18\SOL
Y1	8.00 MHZ	XTAL\RES\SMD
ZNR1	ERZ-CF1MK270	ZNR\3224\SMD

At first glance, the module value names might appear somewhat strange. The names can be interpreted as follows: ECAP\3216\SMD for 3216 size SMD tantalum capacitors, C\0805\SMD for 0805 size SMD capacitors, D\40LS for axial lead diodes with .40 inch lead spacing, CONN\08PAD for .080 inch diameter pads used for connections to signal wires, VRES\T\ADJ\6MM for 6 mm top adjust trimpots, and DIP8\SO for 8 pin SOICs.

Use your text editor to replace the dummy module value names throughout the entire parts section of the TUTOR8A.NET netlist with the names in Table 10-1.

Search and replace techniques can greatly reduce the editing time for large complex designs containing many common components. For example, if the design used several dozen 1UF tantalum bypass capacitors, you could replace very occurrence of the string 1UF 16V TANT with ECAP\3216\SMD.

Pin Names and Pin Arrangements

Before you can edit pin names on the netlist, you need to know what pin names and pin arrangements are required by your PCB design software. These requirements are determined by part definitions in the parts library.

Two lead nonpolarized components such as ceramic capacitors, inductors, and resistors generally do not cause any problems or require special attention. The same applies to ICs because industry standards exists for pin numbering on most common packages. Polarized capacitors, diodes, LEDs, transistors and other three lead discrete semiconductor devices, trimpots, switches, and transformers are a different situation. You must examine these parts in detail and make sure that the netlist matches the PCB parts library.

Start with the documentation for the PCB parts library. Most PCB design software vendors provide graphic illustrations of standard library parts giving dimensions and pin names. You will also need the manufacturer's data sheets for all parts with unique pin arrangements.

The TUTOR8 design includes the following parts that require close examination of pin names and pin arrangement. Pin information abstracted from manufacturer's data sheets is included for your convenience:

- **C1** (22UF 35V electrolytic capacitor): Refer to Figure 10-5 for the ECAP\30SQ\SMD pin arrangement. Note that OrCAD assigns pin 1 to the positive (+) terminal on polarized capacitors. This is not an industry standard and only matches the PCB part by coincidence. No editing is required.
- **C2,C4** (1UF 16V Tantalum capacitor): Refer to Figure 10-6 for the ECAP\3216\SMD pin arrangement. Same note about pin numbers as C1 above. No editing is required.
- **D1** (1N4007 diode): Refer to Figure 10-7 for the D\40LS pin arrangement. OrCAD uses the pin names CATHODE and ANODE for all diodes and LEDs. Change the pin names to pin numbers 1 and 2 respectively.
- **Q1,Q2** (2N4401 transistor): Refer to Figure 10-8 for the TO-92A pin arrangement. OrCAD uses the pin names EMITTER, BASE, and COLLECTOR for all bipolar transistors. The data sheet for a 2N4401 shows pin sequence EBC corresponding to pin numbers 1, 2, and 3 respectively. Note that not all TO-92 bipolar transistors use this same pinout.
- **Q3** (Fuji ET365 power transistor): Refer to Figure 10-9 for the TO-220\UPA pin arrangement. The Fuji data sheet for the ET365 shows pin sequence BCE corresponding to pin numbers 1, 2, and 3 respectively. Note that most TO-220 bipolar transistors use this same pinout.
- **R5,R6** (10K trimpot, Mepcopal CT6P series): Refer to Figure 10-10 for the VRES\T\ADJ\6MM pin arrangement. OrCAD uses the pin names A, WIPER, and B for all trimpots. Examination of the schematic and netlist signal section shows that pin B is the CW (clockwise) terminal. The part data sheet shows that this corresponds to pin 1 of the PCB layout part in Figure 10-10.
- **Y1** (8.00 MHZ ceramic resonator with built-in capacitors): Refer to Figure 10-11 for the XTAL\RES\SMD pin arrangement. OrCAD uses pin numbers 1, 2, and GND. The PCB layout part uses pin numbers 1 to 6. The part data sheet shows that pins are connected in parallel across the two rows and that pins 2 and 5 are ground.

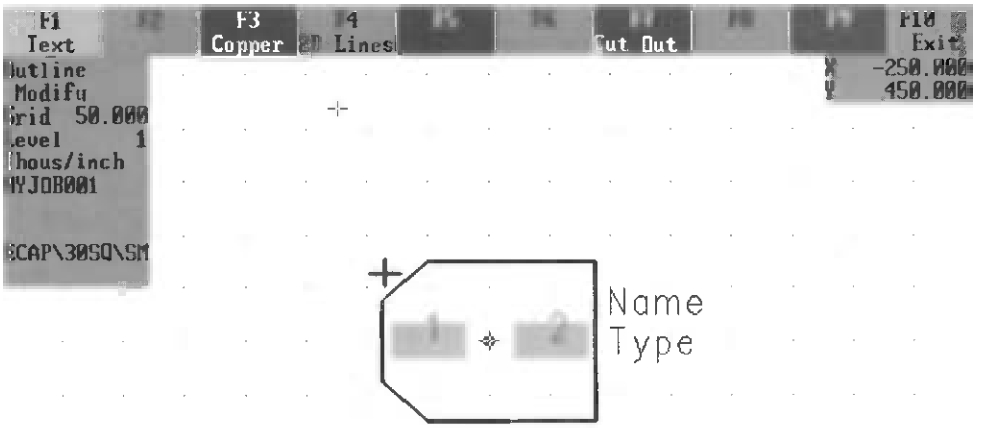


Figure 10-5 ECAP\30SQ\SMD Part Layout

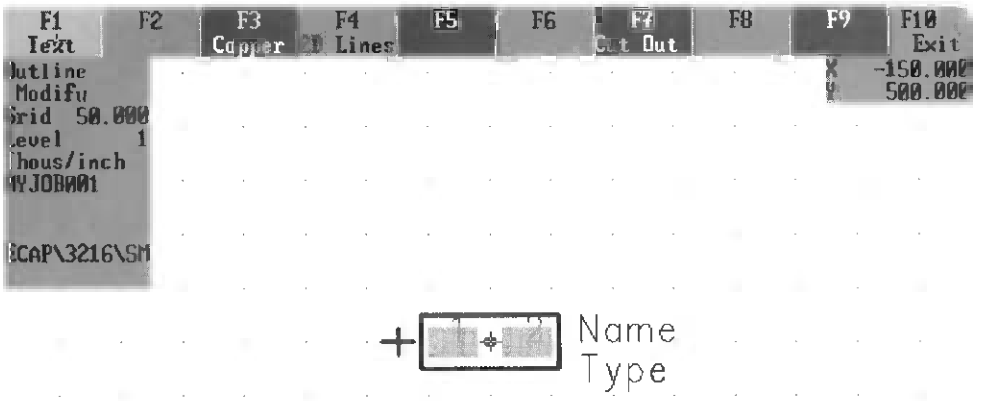


Figure 10-6 ECAP\3216\SMD Part Layout

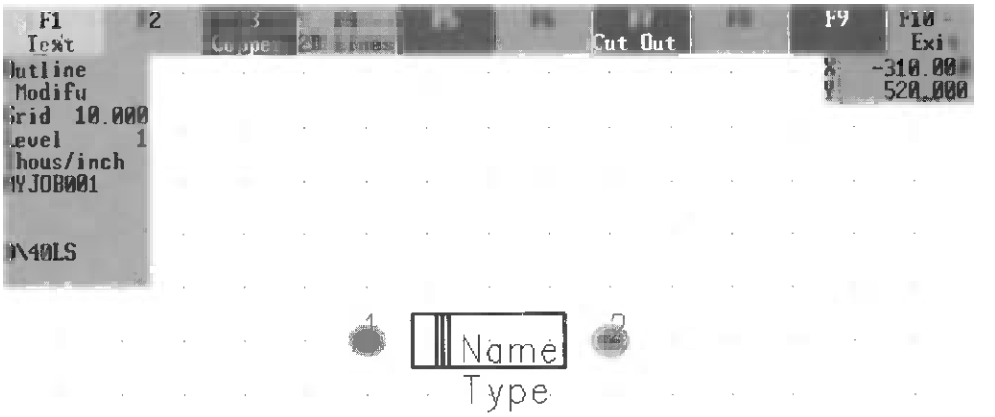


Figure 10-7 D\40LS Part Layout

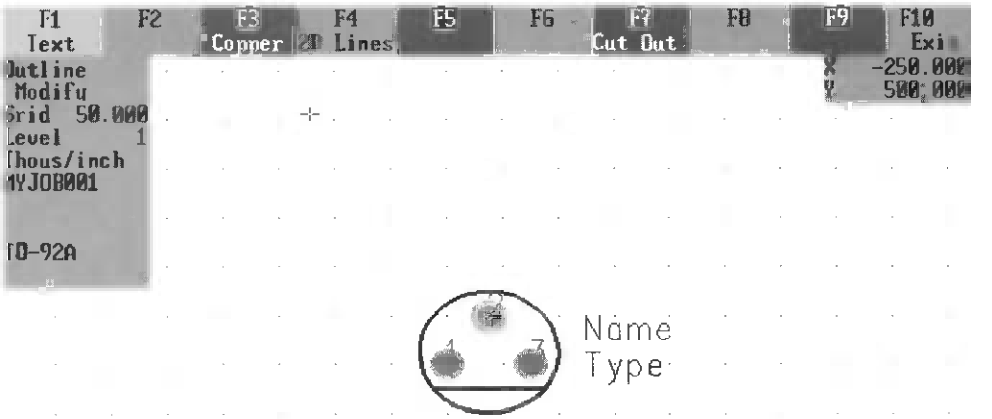


Figure 10-8 TO-92A Part Layout

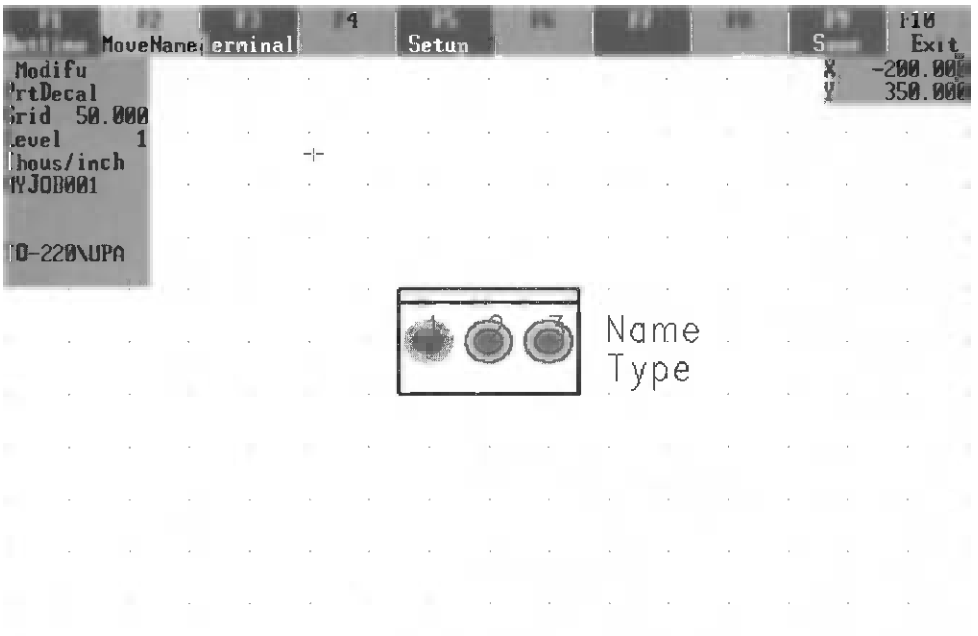


Figure 10-9 TO-220\UPA Part Layout

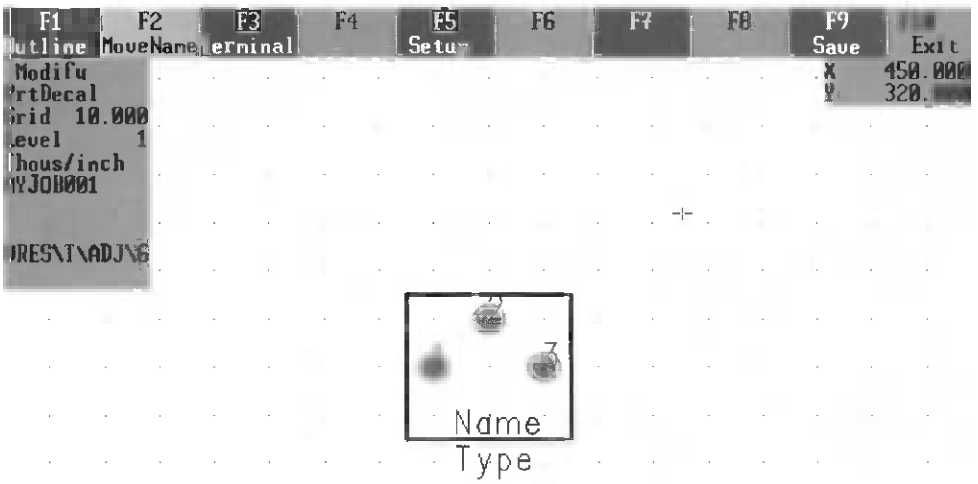


Figure 10-10 VRES\T\ADJ\6MM Part Layout

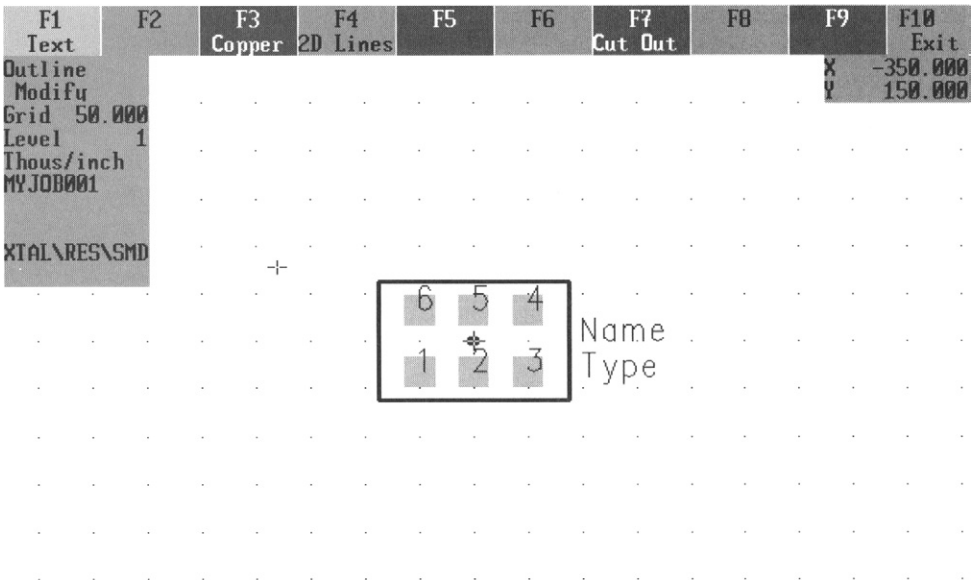


Figure 10-11 XTAL\RES\SMD Part Layout

Editing the Pin Names

Effective editing of pin names requires a text editor with flexible search and replace capability. Each of the parts listed on page 336, with the exception of C1 through C3, require pin name edits. Pin names must be edited in both the parts and signal sections of the netlist.

In some cases, the pin name edits are such that a global search and replace can be used. Diode D1 is an example. You can search and replace the OrCAD pin names CATHODE and ANODE with PCB part pin names 1 and 2 throughout the entire design. Go ahead and try this on the TUTOR8A.NET netlist.

The other parts require more detailed edits because there are no unique 1:1 correspondences between OrCAD and PCB part pin names. Let's look at Q1 as an example. Search TUTOR8A.NET for the first occurrence of Q1. The first occurrence should be in the parts section. Edit the pin names. Use 1, 2, and 3 in place of EMITTER, BASE, and COLLECTOR.

Because Q1 has three terminals, there will be three more occurrences of Q1 in the signal section of the netlist. Search for the next occurrence of Q1. Replace the OrCAD pin name with the required PCB part pin name. Repeat until all occurrences of Q1 have been edited. Then use the same process for the remaining parts on page 336

After completing the edits, print out the netlist. Figures 10-12 and 10-13 show sections of the TUTOR8A.NET netlist after completion of edits. Verify that your netlist printout is similar to these listings. Some minor differences may exist depending on the order in which you originally placed parts of the schematic.

The edited netlist is now ready for input into the PCB design system. Figure 10-14 shows the results of loading the edited TUTOR8A netlist into PADS-Work. The figure shows the PCB design after creation of the board outline and placement of all parts. "Rubberbanded" interconnections appear between pins. The PCB design shown is ready for trace routing.

```
LIST      1              10-30-95 22:20 ♦ C:\ORCAD\TUTOR8\TUTOR8A.NET
NETLIST,2
(DRAWING,ORCAD.NET,1-1
DATA,50,HIERARCHICAL DESIGN EXAMPLE
DATA,51,1001
DATA,52,1.0
DATA,54,   August  8, 1995
)
(SYM,1-1,1
DATA,2,P1
DATA,3,CONN\08PAD
DATA,23,1
)
(SYM,1-1,2
DATA,2,C3
DATA,3,C\0805\SMD
DATA,23,1
DATA,23,2
)
(SYM,1-1,3
DATA,2,R2
DATA,3,R\0805\SMD
DATA,23,1
DATA,23,2
)
(SYM,1-1,4
DATA,2,R5
DATA,3,VRES\T\ADJ\6MM
DATA,23,1
DATA,23,3
DATA,23,2
)
(SYM,1-1,5
DATA,2,R6
DATA,3,VRES\T\ADJ\6MM
DATA,23,1
DATA,23,3
DATA,23,2
)
(SYM,1-1,6
DATA,2,R8
DATA,3,R\1210\SMD
DATA,23,1
DATA,23,2
)
(SYM,1-1,7
DATA,2,D1
DATA,3,D\40LS
DATA,23,2
Command►    *** Top-of-file ***          Keys: ↑↓→← PgUp PgDn F10=exit F1=Help
```

Figure 10-12 TUTOR8A.NET Parts Section after Edits

```
SIG,,***000001,1-1,5,***000001
IN,1-1,7,PL1,23,2
IN,1-1,1,PL1,23,1

SIG,,***000002,1-1,5,***000002
IN,1-1,25,U2,23,7
IN,1-1,25,U2,23,6

SIG,,***000003,1-1,5,***000003
IN,1-1,28,PL3,23,1
IN,1-1,16,R1,23,2
IN,1-1,3,R2,23,1

SIG,,***000004,1-1,5,***000004
IN,1-1,4,R5,23,2
IN,1-1,26,U3,22,17

SIG,,***000005,1-1,5,***000005
IN,1-1,26,U3,22,18
IN,1-1,5,R6,23,2

SIG,,***000006,1-1,5,***000006
IN,1-1,19,Y1,23,3
IN,1-1,26,U3,22,16

SIG,,***000007,1-1,5,***000007
IN,1-1,19,Y1,23,1
IN,1-1,26,U3,22,15

SIG,,***000008,1-1,5,***000008
IN,1-1,6,R8,23,2
IN,1-1,27,PL4,23,1
IN,1-1,14,Q1,23,3

SIG,,***000009,1-1,5,***000009
IN,1-1,11,R7,23,2
IN,1-1,14,Q1,23,2

SIG,,***000010,1-1,5,***000010
IN,1-1,20,PL5,23,1
IN,1-1,22,Q3,23,2

SIG,,***000011,1-1,5,***000011
IN,1-1,18,R10,23,2
IN,1-1,22,Q3,23,1
IN,1-1,15,Q2,23,3

SIG,,***000012,1-1,5,***000012
```

Figure 10-13 TUTOR8A.NET Signal Section after Edits

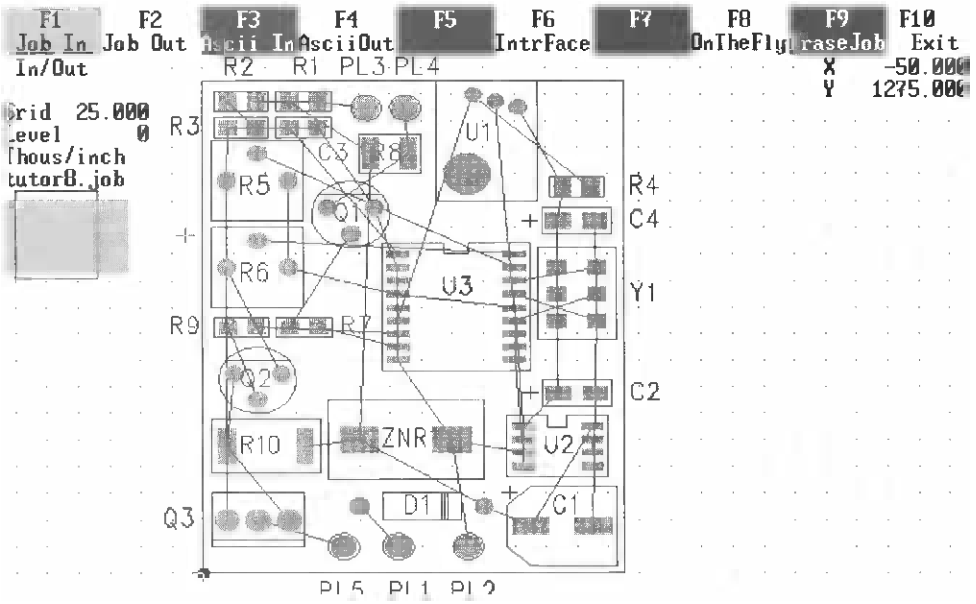


Figure 10-14 TUTOR8A.NET Loaded into PADS-Work

PCB Design Netlist Tips and Techniques

The following is a list of suggested tips and techniques related to netlists for PCB design:

- Before starting on netlist edits, make a file with all the required documentation: schematics, bill of materials and unedited netlist printouts, data sheets for all devices that may require edits, and any available data from the PCB layout part libraries. Keep copies of all this information in a permanent file for the design.
- OrCAD and some PCB design software packages share a common limitation. The programs do not include sufficient documentation for library parts, such as pin names for discrete devices. Often the only solution is to use the library editor to examine parts in question. The author uses a screen capture utility to generate hard copy and keeps a binder containing information on all the parts for future reference. The part layout graphics in Figures 10-5 through 10-11 were captured as TIFF (tagged image file format) files and then printed out. A recommended screen capture utility that appears compatible with most EDA and CAD software is Screen Thief by Nildram Software/FormGen Inc. (Scottsdale, Arizona, Phone: 905-857-0022). Both Windows and DOS versions are available. A shareware version can be downloaded from CompuServe.
- Keep copies of both the original and final edited netlists on disk. If a mistake or discrepancy occurs, these files may become invaluable.
- In the author's experience, the most troublesome parts as far as netlist errors propagating into the PCB design are transistors, trimpots (also panel pots), and switches. For any given case style, transistors may have several different pin arrangements. For example, TO-92 transistors are usually EBC, but some parts are CEB. A corollary of Murphy's Law states that trimpots will always have the clockwise and counterclockwise terminals reversed on the first PCB prototype. Likewise, designers frequently forget that toggle switches are constructed such that the toggle points in the direction opposite to the closed contacts.

11

Tutorial 9 - Editing Bill of Materials

Bill of materials were first discussed in chapter 4 when the Create Bill of Materials tool was introduced. Additional bill of materials concepts were covered as part of the key fields discussion in chapter 7. The discussion in chapter 7 focused on the use of additional part fields and the use of an include file. Recall that the include file provided a means of automatically merging parts information into the bill of materials.

Most users find that simply editing the basic OrCAD generated bill of materials is the most effective approach. Even with the use of additional parts fields or an include file to complete the part descriptions, bill of materials still require further editing. The Create Bill of Materials tool generates a somewhat disorganized header and does not sort parts by value. In this tutorial, you will learn simple techniques for editing bill of materials. This tutorial also covers sorting OrCAD bill of materials via a simple sort utility included on the disk supplied with the book.

The model for this tutorial is the TUTOR8 design that you created as part of the exercise in chapter 10.

Starting the Tutorial Exercise

Use the following sequence of DOS commands to create a new subdirectory, TUTOR9, and to copy the bill of materials file from the previous design:

CD\ORCAD<ENTER>

MD TUTOR9<ENTER>

CD TUTOR8<ENTER>

COPY TUTOR8.BOM C:\ORCAD\TUTOR9\TUTOR9.BOM<ENTER>

All subsequent work will be done on the new TUTOR9.BOM bill of materials file in the TUTOR9 subdirectory.

Figure 11-1 Unedited TUTOR9.BOM Bill of Materials

Preliminary Bill of Materials Editing

The unedited bill of materials is shown in Figure 11-1. The file is shown as it appears on the screen in WordStar, the text editor used by the author. The first two lines are part of the WordStar menu. The next six lines are the OrCAD generated bill of materials header.

The first editing step is to remove the header and write it to a separate file. The reason for removing the header is that it will interfere with the sort utility. You could just erase the header and retype it later but because most text editors provide a means of cutting and pasting text blocks by means of a few mouse clicks, saving the header to a file is more effective. The header will be cleaned up and later pasted back onto the sorted and edited bill of materials.

Remove the header and save it to a file. Use the file name TUTOR9.HDR. Then edit the header as shown in Figure 11-2 and save the file for later use. Clean up the header so that all important information is clearly labeled and neatly aligned in columns. Remove the OrCAD time stamp.

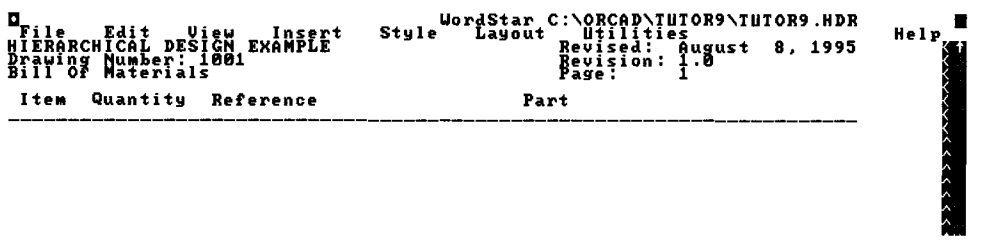


Figure 11-2 Edited TUTOR9.HDR Bill of Materials Header

The next step is to make preliminary edits to the bill of materials. Two areas that generally require special attention include:

- OrCAD schematics may contain PCB layout parts that do not correspond to actual physical parts. In this example, PL1 through PL5 are pads used to solder wire harness connections to the PCB. Because they are not physical parts, they should be deleted from the bill of materials.
- A common error is that different types of capacitors or resistors often appear together on the same line because they were assigned the same part value on the schematic. For example, if the design has 10K resistors, trimpots, and resistor networks, these parts may all appear together. This type of error is often first noticed when the bill of materials is edited. The suggested approach is to edit the schematic using more distinctive part values and then rerun the bill of materials. At a minimum, you must edit the bill of materials and place the different parts on separate lines.

Complete the preliminary edits by removing the lines containing PL1 through PL5. The file should appear as shown in Figure 11-3 with C1 appearing on the first line and no spaces between lines. The file is now ready for sorting.

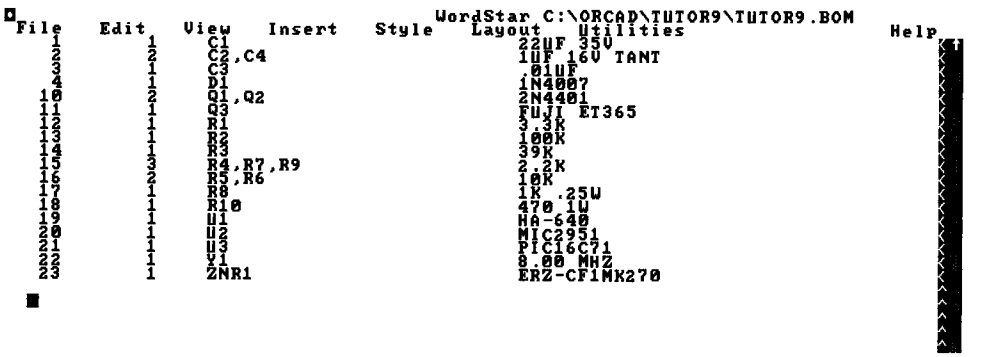


Figure 11-3 TUTOR9.BOM after Preliminary Edits

Sorting the Bill of Materials

Standard industry practice is to sort bill of materials by reference designator and part value. The Create Bill of Materials tool only sorts by reference designator. As you can see in Figure 11-3, capacitor and resistor part values appear in random order. You could use a text editor to manually rearrange the parts in order for a small design such as the tutorial example. However, manually sorting a large design with dozens of different parts becomes a very time consuming task that is prone to errors.

A handy sort utility, BOMSORT.EXE, is included on the disk supplied with this book. The BOMSORT utility automatically sorts parts by value. BOMSORT recognizes the common electrical value multiplier suffixes listed in Table 1-2 in chapter 1. BOMSORT makes the following assumptions about the bill of materials file that is to be sorted:

- **No header**. The file must contain only parts information, without any blank lines. The file shown in Figure 11-3 has a suitable input format. At a minimum each line must contain a reference designator and a parts description. The line length must not exceed 79 characters.
- **Maximum of 399 lines**. This should suffice for most practical designs. If a very large design contains more than 399 lines in the bill of materials, the file could be split into two sections and each section separately sorted.
- **Reference designator in columns 18 and 19**. BOMSORT assumes that the input file has already been sorted by reference designator, as is the case for files generated by the OrCAD Create Bill of Materials tool. BOMSORT accepts a reference designator with up to a two letter prefix, such as the standard prefixes listed in Table 1-1 in chapter 1.
- **Part description starting in column 44**. BOMSORT uses the first five characters (columns 44-48) of the part description to sort parts. Files generated by the OrCAD Create Bill of Materials tools always have the part description starting in column 44.

Load the BOMSORT utility onto your hard drive. The ORCADESP directory is the suggested location for OrCAD related utilities. If you use the ORCADESP directory, make sure that you add it to the PATH statement in your AUTOEXEC.BAT file and reboot the computer.

Assuming that you have loaded BOMSORT.EXE into your ORCADESP directory and added the directory to your path, use the utility to sort the TUTOR9.BOM file by typing:

CD\ORCAD\TUTOR9<ENTER>

BOMSORT<ENTER>

BOMSORT prompts for the name of the bill of materials input file and output file. The input file is not modified. Enter TUTOR9.BOM for the input filename and TUTOR9A.BOM for the output filename. BOMSORT exits back to DOS after completion.

Use your text editor to examine and print out the sorted bill of materials file TUTOR9A.BOM. Verify that your sorted file appears similar to that shown in Figure 11-4. Note that capacitors and resistors are now listed in order of parts value.

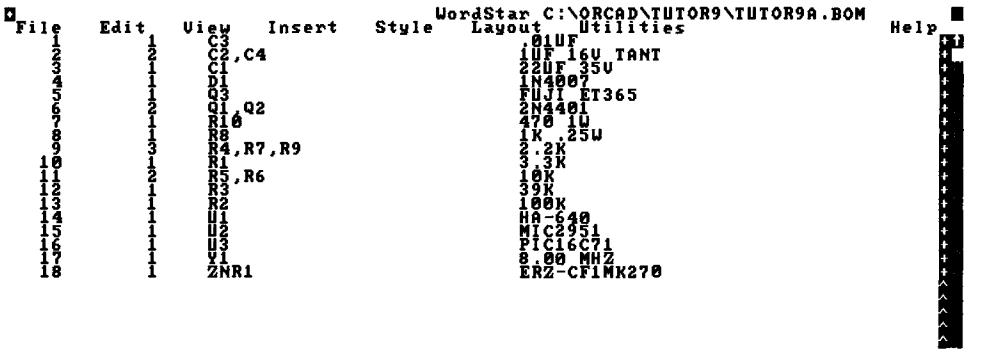


Figure 11-4 Sorted TUTOR9A.BOM Bill Of Materials

Completing the Bill of Materials

The final steps in completing a bill of materials include inserting the header back into the sorted file and adding any required information to the parts descriptions. Typical information added to the parts descriptions includes vendor names and vendor part numbers. Abbreviated parts descriptions used on schematics are usually rewritten in a more detailed and formal manner on the bill of materials. For example, schematics usually include notes to the effect that all resistors and capacitors are a certain wattage, voltage, and tolerance. Recommended practice is to write out such information in detail for each part on the bill of materials.

The completed bill of materials for the tutorial example is shown in Figure 11-5.

```
                                    WordStar C:\ORCAD\TUTOR9\TUTOR9A.BOM
 File   Edit   View   Insert   Style   Layout   Utilities                    Help
.LM1
.RM80
HIERARCHICAL DESIGN EXAMPLE                    Revised:  August  8, 1995
Drawing Number: 1001                           Revision: 1.0
Bill Of Materials                              Page:     1

 Item  Quantity  Reference                Part
-------------------------------------------------------------------------------
    1         1  C3                       .01UF 50V 10% X7R 805 SMD CAP
                                          PANASONIC ECU-V1H103KBG
    2         2  C2,C4                    1UF 16V 20% TANTALUM 3216 SMD CAP
                                          PANASONIC ECS-H1CY105R
    3         1  C1                       22UF 35V 20% ELECTROLYTIC SMD CAP
                                          PANASONIC ECE-V1VA220P
    4         1  D1                       1N4007 DIODE
    5         1  Q3                       FUJI ET365 TRANSISTOR
    6         2  Q1,Q2                    2N4401 TRANSISTOR
    7         1  R10                      470 1W 5% 2512 SMD RES
    8         1  R8                       1K .25W 5% 1210 SMD RES
    9         3  R4,R7,R9                 2.2K .1W 5% 0805 SMD RES
   10         1  R1                       3.3K .1W 5% 0805 SMD RES
   11         2  R5,R6                    10K CERMET TRIMPOT 1 TURN TOP ADJ
                                          MEPCOPAL CT6P SERIES
   12         1  R3                       39K .1W 5% 0805 SMD RES
   13         1  R2                       100K .1W 5% 0805 SMD RES
   14         1  U1                       HA-640 HALL EFFECT SENSOR
                                          CLAROSTAT
   15         1  U2                       MIC2951 SOIC
                                          MICREL
   16         1  U3                       PIC16C71 SOIC
                                          MICROCHIP
                                          WITH REV 1.0 FIRMWARE
   17         1  Y1                       8.00 MHZ SMD RESONATOR
                                          PANASONIC EFO-V8004E5
   18         1  ZNR1                     ERZ-CF1MK270 SMD SURGE ABSORBER
                                          PANASONIC
   19         1                           PCB REV 1.0
```

Figure 11-5 Completed TUTOR9A.BOM Bill of Materials

The completed bill of materials will also include information about parts that are not included on the schematic but that are required to build the product. Examples of such parts include the printed circuit board and hardware.

Recommended practice is to make a bill of materials for each separable assembly. In the case of the tutorial example, the bill of materials represents the printed circuit board assembly. A separate bill of materials would be generated for the final assembly, which would include the printed circuit board assembly, housing, wire harness, mounting hardware, labels, and any other required miscellaneous items.

Importing Bill of Materials Files into Microsoft Excel

Microsoft Excel is an excellent tool for managing and printing bill of materials. Other spreadsheet programs, such as Quatro Pro, can be used for the same purpose. Using a spreadsheet offers easy organization of data in columns and the capability of inserting and totaling cost information.

Use the following steps to import an OrCAD generated bill of materials file into Excel (the assumption is made that the file has been sorted and edited as previously described in this tutorial):

- Edit the file so that all parts descriptions are on a single line. Note that the TUTOR9A.BOM file shown in Figure 11-5 uses multiple line parts descriptions, with the vendor name and part number on the second line. For better readability, leave a blank line between successive parts. You can eliminate the header now or during the file import process.
- Launch Excel and open the file. The Excel Text Import Wizard will appear because the file is not in Excel format. The Text Import Wizard consists of three steps, described in detail below.
- Text Import Wizard Step 1 - Select Data Type. This step selects the type of data to be imported. Select fixed width data, since the bill of materials file is organized into columns. You can also select the starting row. Select starting row 10 to eliminate the header information. You must also select a file origin. Select DOS or Windows.
- Text Import Wizard Step 2 - Set Field Widths. This step sets the column breaks. Set the column breaks to make separate columns for item number, quantity, reference designator, part value, and vendor name/part number.
- Text Import Wizard Step 3 - Set Data Format. This is the final step. You can set the data format for each column or skip particular columns. Skip the first column (item number) and set the remaining columns to text format.

Excel now imports the bill of materials data in neat columns. You can format the column width and insert a row at the top with column labels. The finished spreadsheet appears as shown in Figure 11-6. Note that Excel does not automatically save the file in Excel format. Make sure you select Excel format when you save your spreadsheet, otherwise some of the formatting information may be lost.

QTY	REFERENCE DESIGNATOR	DESCRIPTION	VENDOR PART NUMBER
1	C3	.01UF 50V 10% X7R 805 SMD CAP	PANASONIC ECU-V1H103KBG
2	C2,C4	1UF 16V 20% TANTALUM 3216 SMD CAP	PANASONIC ECS-H1CY105R
1	C1	22UF 35V 20% ELECTROLYTIC SMD CAP	PANASONIC ECE-V1VA220P
1	D1	1N4007 DIODE	
1	Q3	FUJI ET365 TRANSISTOR	
2	Q1,Q2	2N4401 TRANSISTOR	
1	R10	470 1W 5% 2512 SMD RES	
1	R8	1K .25W 5% 1210 SMD RES	
3	R4,R7,R9	2.2K .1W 5% 0805 SMD RES	
1	R1	3.3K .1W 5% 0805 SMD RES	
2	R5,R6	10K CERMET TRIMPOT 1 TURN TOP ADJ	MEPCOPAL CT6P SERIES
1	R3	39K .1W 5% 0805 SMD RES	
1	R2	100K .1W 5% 0805 SMD RES	
1	U1	HA-640 HALL EFFECT SENSOR	CLAROSTAT
1	U2	MIC2951 SOIC	MICREL
1	U3	PIC16C71 SOIC WITH REV 1.0 FIRMWARE	MICROCHIP
1	Y1	8.00 MHZ SMD RESONATOR	PANASONIC EFO-V8004E5
1	ZNR1	ERZ-CF1MK270 SMD SURGE ABSORBER	PANASONIC
1		PCB REV 1.0	

Figure 11-6 TUTOR9A.BOM Imported into Excel

Conclusion

Congratulations! You have completed the last of the tutorials. At this point you have mastered all of the details of OrCAD. You should now be able to productively use OrCAD on a professional basis for your daily work.

12

Command Reference

This chapter provides a quick reference to the OrCAD SDT Draft and Library Editor menu structure and commands. Draft and Library Editor are the only OrCAD tools with mouse activated "pop up" menus. Draft is used to capture and edit schematics. The Library Editor is used to create and edit parts libraries. Both tools are accessed from the main OrCAD SDT menu shown in Figure 12-1.

The other tools shown on the main SDT menu do not have associated commands or menus. The action of these tools is controlled via local configurations, some of which have extensive options. A quick summary at the end of the chapter provides an overview of all OrCAD tools and references page numbers in this book where more detailed information can be found. Design Management Tools are accessed from the main OrCAD ESP menu. Design Management Tools are explained in detail in chapter 7 starting on page 275.

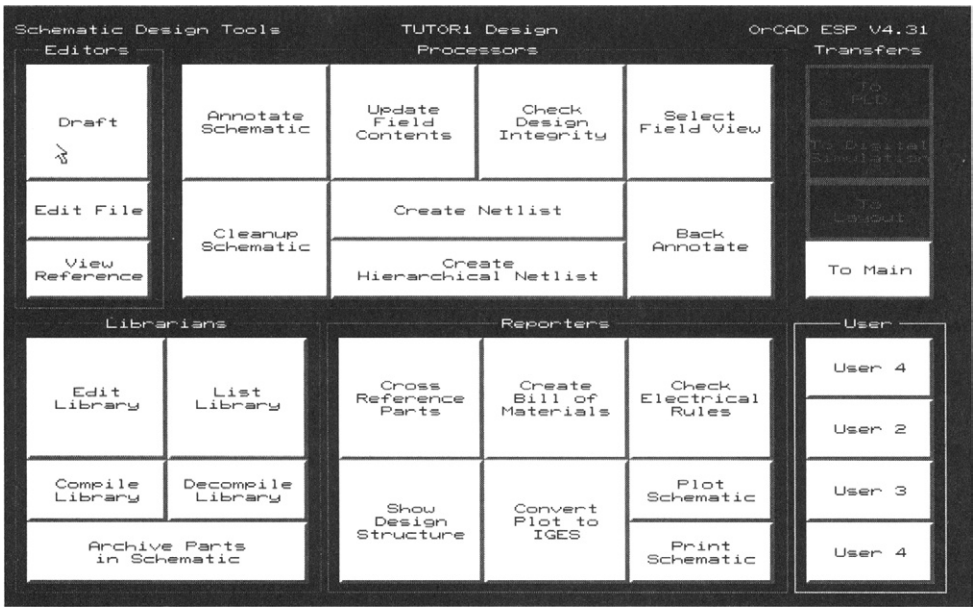


Figure 12-1 OrCAD SDT Main Menu

Draft Commands

Draft is the main OrCAD SDT program or tool. Draft is used for schematic capture and editing. All commands can be accessed via a main menu. The main menu is brought up by clicking on the left mouse button. Once the menu appears, the normal mouse arrow cursor remains stationary. Vertical mouse movements cause menu commands to be highlighted. A highlighted menu command can be selected by pressing the left mouse button again. Some commands have sub-menus, as shown in Figure 12-2, with more command options.

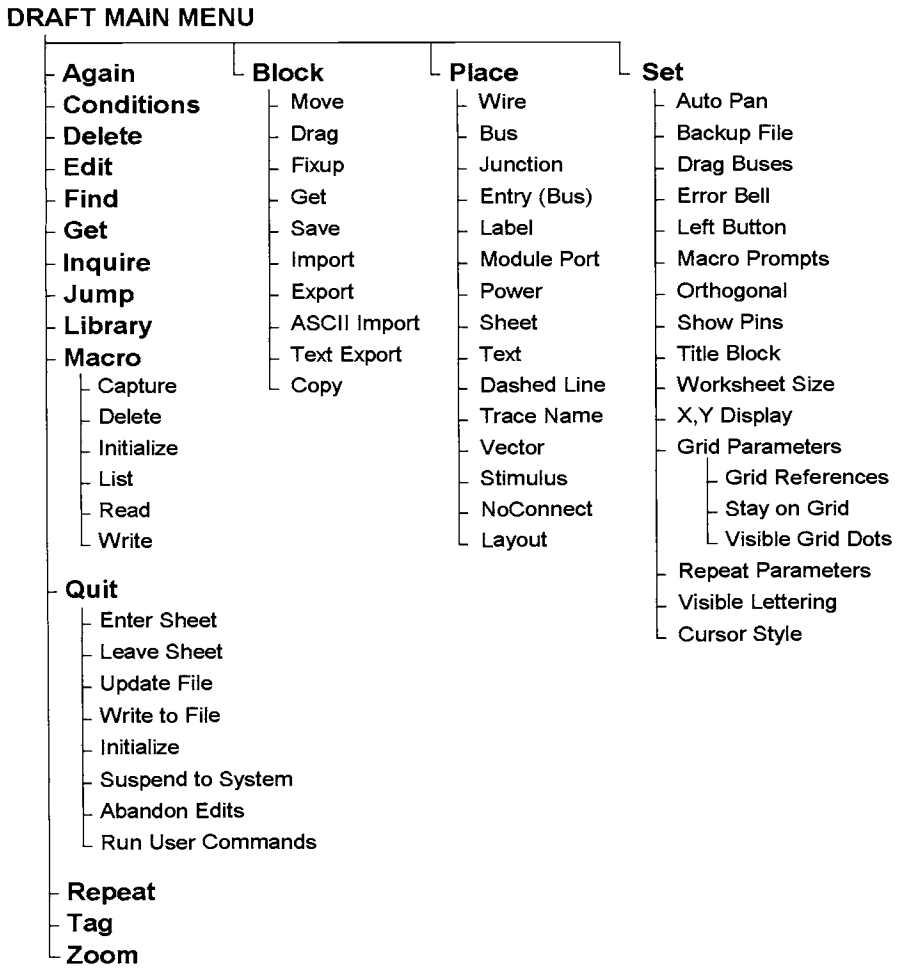


Figure 12-2 Draft Main Menu Structure

As far as mouse usage, use the left mouse button to select a highlighted entry. The left mouse button acts the same as the <ENTER> key. The right mouse button acts the same as the <ESC> key. Use the right mouse button to escape or close a menu. Commands can also be selected by typing in the first letter of the command or using macros.

Draft Main Menu

An overview of the main menu commands shown in Figure 12-2 follows. Commands that have extensive sub-menus are explained in subsequent sections. For configuration details, refer to chapter 2 starting on page 46. For more detailed information about using Draft, refer to the tutorial exercises starting in chapter 3 on page 65.

Again — Repeats the last command, but only repeats the first menu level. For example if the last command was Place Wire, using Again would only get you to the Place menu.

Block — Commands on the Block menu allow manipulating selected areas of the drawing. You can move, rubberband, copy, and paste areas of the drawing. Block commands are extremely useful for editing.

Conditions — Not applicable to most PC systems used today. Checks how much RAM memory is left available. With today's PC systems, available memory is rarely a cause for concern.

Delete — Allows deleting individual objects or entire areas of the drawings. Also has a lifesaving Undo option if you make a mistake.

Edit — Used to edit text, part descriptions, reference designators, and many other objects and fields. Unlike most other Draft commands, you must first position the mouse cursor to select an object and then launch Edit. Options will depend on the nature of the object selected.

Find — Locates the cursor at the object that contains a character string entered with the Find command. Occasionally useful for finding components based on part value or reference designator.

Get	Accesses the OrCAD libraries for placing a part on the schematic.
Inquire	Specialized command for accessing hidden text associated with certain objects. Not frequently used.
Jump	Moves the cursor to a preset tag location (see Tag command), to a grid reference, or along the X or Y axis. Forget about this command and use your mouse.
Library	Displays available libraries and the parts in the libraries. This command duplicates much of the functionality of the Get command, except that parts cannot be placed into the drawing. Library command options include: **Directory**. Displays a list of available libraries. Contents of the highlighted library can be listed to the screen or to a file. **Browse**. Displays part graphics on the screen. Allows scrolling through the parts in a selected library.
Macro	Used to capture macro commands and manage macro files. Not required to access the macro file automatically loaded at startup (defined in the SDT configuration).
Place	Used to place (draw) various objects onto the schematic, including wires, buses, junctions, module ports, sheets, and text.
Quit	Similar to the File menu found in Windows programs, Quit provides access to file management functions. Also used to navigate between sheets and levels in hierarchical schematics.
Repeat	Duplicates the last object placed into the schematic. By setting parameters with the Set command, labels or module port names can be automatically incremented and the objects placed on adjacent grid locations. This can result in significant time savings for 16 or 32 bit wide address or data buses.

Set Allows changing parameters for numerous Draft options including the worksheet size and grid.

Tag Sets up to eight tag locations for use with the Jump command. Not frequently used.

Zoom Used to change the zoom scale. Normal scale is 1, which provides the greatest level of detail. Scales 2-10 zoom out to show larger areas. Unless you configure OrCAD for very high resolution and have a large monitor, only the normal scale 1 is useful.

Draft Block Command Sub-menu

Commands on the Block menu allow manipulating selected areas of the drawing for editing. Most of the Block commands require using the mouse to define a box used to select objects. After starting the command, move the mouse cursor to the first corner point. Click the left button and then move the mouse cursor to the second corner point. A selection box outline will appear. When you click the left button again, all objects crossing the box are selected. A ghosted image of the selected objects appears and follows the cursor. Clicking the left button one more time locks the objects into place. For more details on using Block commands, refer to chapter 3 page 82 and chapter 7 page 251. The following Block command options are available:

Move Used to move objects crossing a selection box from one location to another on the same sheet. Any wire or bus segments crossing the selection box are moved in their entirety, including the endpoints. Connections at wire and bus endpoints are broken.

Drag Used to drag objects crossing a selection box from one location to another on the same sheet. Any wire or bus segments crossing the selection box are "rubberbanded." The wire or bus segments are stretched as required. Endpoints outside the selection box remain fixed and connections are not broken.

Fixup Specialized Block command used to "fix up" non-orthogonal wires and buses. Adds new segments to make the selected wires or buses orthogonal. This command has limited usefulness because most work in Draft is done with orthogonal mode set on. Fixup seems cumbersome to use. Many users find that

deleting and redrawing is more convenient when a mistake does require fixing up.

Get Retrieves objects previously saved to the clipboard (OrCAD uses the term buffer) via Block Save. Displays a box containing the objects. Multiple copies can be placed at desired locations on the sheet by clicking the left mouse button. Note that the clipboard is also used by the Block Drag and Move commands. Anything saved in the clipboard is lost when these commands are used. A further limitation is that the clipboard is flushed when navigating between sheets. In most cases, using the Block Copy command is more convenient.

Save Objects crossing a selection box are saved to the clipboard. These objects can then be retrieved using the Block Get command. Block Save and Block Get are analogous to Windows' Copy and Paste. A limitation is that the clipboard is flushed and any saved objects are lost when you use Block Drag or Move commands or when navigating between sheets. In most cases, using the Block Copy command is more convenient.

Import Retrieves objects previously saved to a file via Block Export. Prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Displays a box containing the objects retrieved from the file. Multiple copies can be placed at desired locations on the sheet by clicking the left mouse button.

Export Objects crossing a selection box are saved to a specified file. These objects can then be retrieved using the Block Import command. Block Export and Import are useful for moving objects between sheets or designs. After objects are selected, Block Export prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Suggested practice is to use the .BLK extension for exported block files.

ASCII Import Retrieves ASCII text from a file. Prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Places text immediately to the right of the cursor location. The ASCII text file can be created with any text editor or via the Block Text Export command and can contain multiple lines. When using a text editor, make sure that the imported text is free of any special formatting characters. The Block ASCII Import command is commonly used to import standardized notes.

Text Export Text objects crossing a selection box are saved to a specified file as ASCII text. The ASCII text can then be edited with a text editor or retrieved using the Block ASCII Import command. After text objects are selected, Block Text Export prompts for a path and file name. If no path is entered, the path defaults to the current design subdirectory. Suggested practice is to use the .TXT extension for exported ASCII text files. Note that only text objects are selected. Any characters associated with labels, part descriptions and reference designators, module ports, sheets, title block, or other objects are ignored.

Copy Objects crossing a selection box can be copied to a new location. Multiple copies can be placed at desired locations on the sheet by repeatedly clicking the left mouse button. The Block Copy command combines the action of Block Save and Block Get into one command.

Draft Macro Command Sub-menu

Macro commands improve productivity by reducing the number of keystrokes and mouse clicks required to carry out common, repetitive tasks. Macro commands are stored in macro files on disk. The Macro command is used to capture macros and manage macro files. A macro file and initial macro command can be defined in the SDT configuration. This macro file is then automatically loaded whenever Draft is launched. For more details on configuring a startup macro, refer to chapter 2 page 52. For more detailed information about using the Macro command and the CUSTOM.MAC macro file supplied on disk with this

book, refer to chapter 3 page 69 and chapter 8 page 308. The following options are available on the Macro menu:

Capture	Used to capture (define) a macro command from within Draft. Prompts for the macro name followed by the sequence of commands that make up the macro. Any valid sequence of keystrokes and mouse movements and clicks is recorded. Press <CTRL> <END> to stop the capture command. The macro command is written to the macro buffer.
Delete	Used to delete a macro command from the macro buffer. Prompts for the macro name key or key combination.
Initialize	Clears the macro buffer.
List	Lists the names of all macro commands available in the macro buffer. The List option does not display any information about the macros.
Read	Loads a new macro file into the buffer. Prompts for the macro filename.
Write	Saves the macro buffer contents to a file. Prompts for the filename. The macro buffer only stores macro definitions, not comments. If you use Macro Read to read a macro file (or OrCAD loads an initial macro at startup) and then use Macro Write, any comments are lost.

Draft Place Command Sub-menu

The Place menu can be used to draw all OrCAD objects. However, many users find that macro commands, such as those found in the CUSTOM.MAC macro used in the tutorial exercises, provide a more convenient means of quickly drawing the most widely used objects such as wires and junctions.

Place commands require using the mouse to locate the object insertion point or starting, corner, and end points. After starting the command, move the mouse cursor to the first point and click the left button. If the object requires additional points, such as corner and end points for a wire, a sub-menu appears. Placing some objects including text, labels, and module ports will result in prompts for additional information. Once these type of objects have been placed, the Edit command can be used to change their descriptive text or appearance. The Place

command and details about the most commonly used objects are covered in the first three tutorial exercises, starting in chapter 3 on page 71. The following objects appear on the Place menu:

Wire

Electrical connections between pins and objects. To give valid results, wires must be drawn in strict accordance with the following rules:

- Wires must begin and end on: a part pin, module port, power or ground object, junction, or bus entry. There is one exception listed below.
- Wires must be drawn as a continuous line from start point to end point. Never draw breaks where wires cross over one another. When it is not feasible to draw a continuous wire, label objects must be associated with each segment.
- Wires that cross over one another or intersect are not electrically joined unless a junction object is placed at the intersection point.
- Do not overlap wires onto pins. The wire must start at the grid location where the pin ends.
- Do not overlap segments of the same wire. If a wire is extended, start the new segment at the grid location where the last segment ends.

Bus

A means of representing multiple signal wires. Buses should be used for all data and address lines and any other signals that can be logically grouped together. Using buses improves readability and reduces clutter. Bus objects appear as thick lines. The following special rules apply to bus objects:

- Labels are used for naming buses. Every bus must have an associated label. The label name is of the form NAME[A..B] where NAME is any legal alphanumeric label name and A and B are integers that represent the wire numbers branching to and from the bus. Note that square brackets [ ] must be used and the numbers must be separated by two periods. Examples of valid names include: D[0..7] and ADDR1[16..31]. A must be less than B. Negative numbers are not

allowed. The label must be placed with the "hot point" (lower left hand corner) touching the bus.

- Signals branching from a bus have names based on the bus name. For example, data bus D[0..7] has eight data signals D0, D1, through D7. A given signal can branch from a bus any number of times.
- Signals can only be routed to or from a bus via a special entry object that appears as a 45 degree diagonal slash. Wires that cross a bus or end at a bus are not electrically connected to the bus unless an entry object is used. Junctions cannot be used to connect pins or wires to a bus.
- Sections of the same bus can be joined together in a "Y" or can "cross" without junctions (junctions have no effect on bus objects). Different buses that cross one another do not join, just as with wires. A bus can be routed between sheets by module ports. The name of the module port must correspond exactly with the bus name.
- Buses are usually drawn with 45 degree beveled corners and a 45 degree split at the point where the bus splits into two separate sections. Two styles of bus entry appearing as / or \ are available. Pick an entry style that flows with the direction of the bus.

Junction Electrical interconnection point for crossing wires.

Entry (Bus) Used to connect wires to buses. Appear as a short 45 degree diagonal line. Two styles appearing as / or \ are available.

Label A means of identifying signals. Can also be used to electrically join wire sections and route signals between different areas on a given sheet.

Module Port Used to route signals between sheets in hierarchical designs. Must correspond to sheet nets.

Power	Used to connect wires and pins to power planes. Several styles and orientations at 90 degree increments are available. Power objects are named the same as the corresponding power plane.
Sheet	Used in hierarchical designs. Can be thought of as representing the circuit blocks in a block diagram.
Text	Used to annotate schematics with miscellaneous information such as notes. No electrical properties.
Dashed Line	Like text, dashed lines can be used to annotate schematics and do not have any electrical properties. Common uses are to outline functional circuit blocks within a sheet or to show hand wired connections used to implement engineering changes on a printed circuit board.
Trace Name	Advanced feature used in conjunction with digital timing simulation software. Identifies a node to be traced (timing analyzed).
Vector	Advanced feature used in conjunction with digital timing simulation software. Associates a test vector with a particular node.
Stimulus	Advanced feature used in conjunction with digital timing simulation software. Associates a stimulus signal with a particular node.
NoConnect	Special object placed on unconnected pins or wires to identify them as deliberately unconnected and prevent an error from being flagged by the Electrical Rules Check routine. Appears as a small X.
Layout	Advanced feature used in conjunction with OrCAD PCB design software to associate a layout directive (such as trace width or via type) with a given node.

Draft Quit Command Sub-menu

The Quit command is similar to the File menu found in Windows programs, Quit provides access to file management functions. Quit is also used to navigate between sheets and levels in hierarchical schematics. Note that OrCAD does not automatically save the design when exiting the program or leaving a particular

sheet. You must use the Update File option to save your work. The following options appear on the Quit menu:

Enter Sheet Used to navigate to a lower level on a schematic hierarchy. Click on the Enter Sheet command, then position the cursor on the sheet object to be entered and click the left mouse button again. The selected sheet will appear on the screen.

Leave Sheet Used to navigate to a higher level on the schematic hierarchy. Leaves the current sheet and returns to the next higher level where the sheet appears as a sheet object.

Update File Writes the schematic sheet data to disk with all edits made during the session. Always use this option to save your work **before** exiting via Abandon Edits. Good practice is to also use Update File whenever a significant task has been accomplished (to save recent work in case of system crash or power failure) or before a major edit (OrCAD has very limited undo capabilities).

Write to File Same action as Update File except this option is used to write data to a specified file.

Initialize Abandons any edits since the last file update and then prompts for the name of a new schematic design.

Suspend to System Suspends program operation and loads the DOS command interpreter to allow DOS operations. An additional > system prompt appears as a reminder that OrCAD is still in the background. Type EXIT<ENTER> to exit DOS and return to the Draft program.

Abandon Edits Ends the editing session and exits back to the main SDT screen. Does not save any edits. You must use Update File before Abandon Edits in order to save your work. If any edits were made during the session, Abandon Edit asks for a YES/NO confirmation before exiting.

Run User Commands Similar to Suspend To System, except exits to DOS and runs the DRAFTUSR command in the design

directory or on the DOS path. This can be a DOS batch file named DRAFTUSR.BAT. Automatically exits DOS and returns to Draft after the DRAFTUSR command is completed.

Draft Set Command Sub-menu

The Set command is used to set certain Draft display and editing options. Note that most of the menu options are followed by a YES/NO indication that shows the status of the option. Options which do not include a status indication have an additional sub-menu. For more detailed information about the Set command, refer to chapter 5 page 159. The following options are available on the Set menu:

Auto Pan Enables automatic panning (movement of the display window relative to the sheet) when the mouse is moved to the edge of the display window. An archaic option from the days of slow 286 and 386 class PCs. Turning Auto Pan off slightly speeds up the display, but makes moving around the sheet very difficult. Always set Auto Pan to YES.

Backup File Enables automatic creation of a backup file whenever the Quit command is used to write to or update a schematic file that already exists. The original file is saved with the filename extension .BAK. Always set Backup File to YES. If a serious error, system crash or other unforeseen event occurs, the backup file allows recovery of data.

Drag Buses Enables rubberbanding of buses when using the Block Drag command. Another archaic option related to graphics performance. Always set Drag Buses to YES.

Error Bell Enables the error beep tone. OrCAD has an annoying tendency to flag even relatively innocuous actions as errors. For example, trying to select an object for editing and not having the mouse cursor within the object border. When an error occurs, everything is locked up for about one second and you must then click the right mouse button to escape. The error beep tone helps you recognize the occurrence of an error.

Left Button Enables automatic generation of an <ENTER> keystroke when the left mouse button is released. If this option is disabled, many commands will require double mouse clicks.

Macro Prompts Displays the sequence of commands contained in the macro when a particular macro is executed. This is another archaic option related to graphics performance. Always set Macro Prompts to YES. If Macro Prompts is set to NO, screen redraw during macro execution is disabled and any macros using pan or zoom will fail.

Orthogonal When Orthogonal mode is set to YES, wires and buses can only be drawn in the vertical or horizontal direction. This can result in a significant time savings because OrCAD automatically places a corner if the mouse cursor is moved in an "L" motion. Generally accepted practice is to draw all wires in orthogonal mode. To help differentiate buses from wires, corners on buses are generally drawn with a 45 degree bevel. Temporarily set Orthogonal mode to NO while drawing buses.

Show Pins This option actually means "show pin numbers." Normally set to YES. Setting Show Pins to NO eliminates display (and hard copy) of pin numbers on library parts. Pin names are not affected. The only circumstance where display of pin numbers is not required would be an abstract schematic used for instructional or reference purposes and not intended to be built up as a functioning circuit.

Title Block Enables display and hard copy of the title block area. Generally accepted practice requires that all engineering drawings include a title block, so this option is normally set to YES.

Worksheet Size Sets the size of the worksheet area. Valid entries are A through E. The actual worksheet dimensions and associated parameters are defined in the template table during the configuration process (refer back to chapter 2).

X,Y Display Enables display of X,Y coordinates. The OrCAD coordinate system is somewhat of an oddity. Unlike normal Cartesian coordinates, OrCAD coordinates define the zero (origin) point at the upper left corner. X and Y coordinates increase going to the lower right corner. X,Y display is normally set to YES.

Grid Parameters This command has a sub menu that is discussed in detail in the next section.

Repeat Parameters This command has a sub-menu with four options used to set parameters for the Repeat command on the main menu and to allow automatic incrementing of label and module port numeric suffixes, that is DATA0, DATA1, DATA2 and so forth. The options include:

X Repeat Step. Sets the X axis step size in grid units.

Y Repeat Step. Sets the Y axis step size in grid units.

Label Delta. Sets the label or module port number suffix step, that is step size 1 results in labels such as A0, A1, A2, and so forth.

Auto Increment (Yes/No). Enables automatic increment of label and module port number suffixes.

Visible Lettering This command only applies to zoom scale 2. Unless you are using a very large monitor and high resolution graphics board, zoom scale 2 is almost useless. Visible Lettering has a sub-menu that allows selecting what types of text, that is pin numbers, labels, and so forth, are still displayed at zoom scale 2.

Cursor Style Selects the cursor style: either an "arrow" pointer or a crosshair.

Draft Grid Parameters Sub-menu

The following additional options are available on the Grid Parameters second level sub-menu:

Grid References Enables display of alphanumeric grid references on the top and left borders of the display. Does not affect hard copy. Useful for maintaining your orientation on the sheet.

Stay on Grid Forces the cursor to stay on grid when placing objects. Caution! Never place any objects off grid. Always leave this option set to YES. Placing electrical objects off grid may cause serious errors. Connections between objects located off grid can appear intact, yet not be recognized by postprocessing routines such as Check Electrical Rules and Netlist. In some cases text (but never labels) might be located off grid to provide better alignment. However, if text or any other object is located off grid, locating the cursor on the object for editing or deleting becomes very difficult. In most cases Delete Block or Block Move/Drag are the only commands that can find and manipulate off grid objects.

Visible Grid Dots Enables display of grid dots. Visible Grid Dots should always be set to YES. Note that this option has no effect on whether or not objects are placed on grid. At the normal zoom scale 1, the spacing between grid dots is 1/10 of X and Y units. For example, configuring SDT for inch units results in 10 grid dots per inch at zoom scale 1.

Library Editor Commands

The Edit Library tool (library editor) is used to create and edit library parts. Edit Library has many similarities to Draft, including the use of the mouse to select menu commands. The Edit Library main menu shown in Figure 12-3 also has the same organization and shares many of the same commands as in Draft.

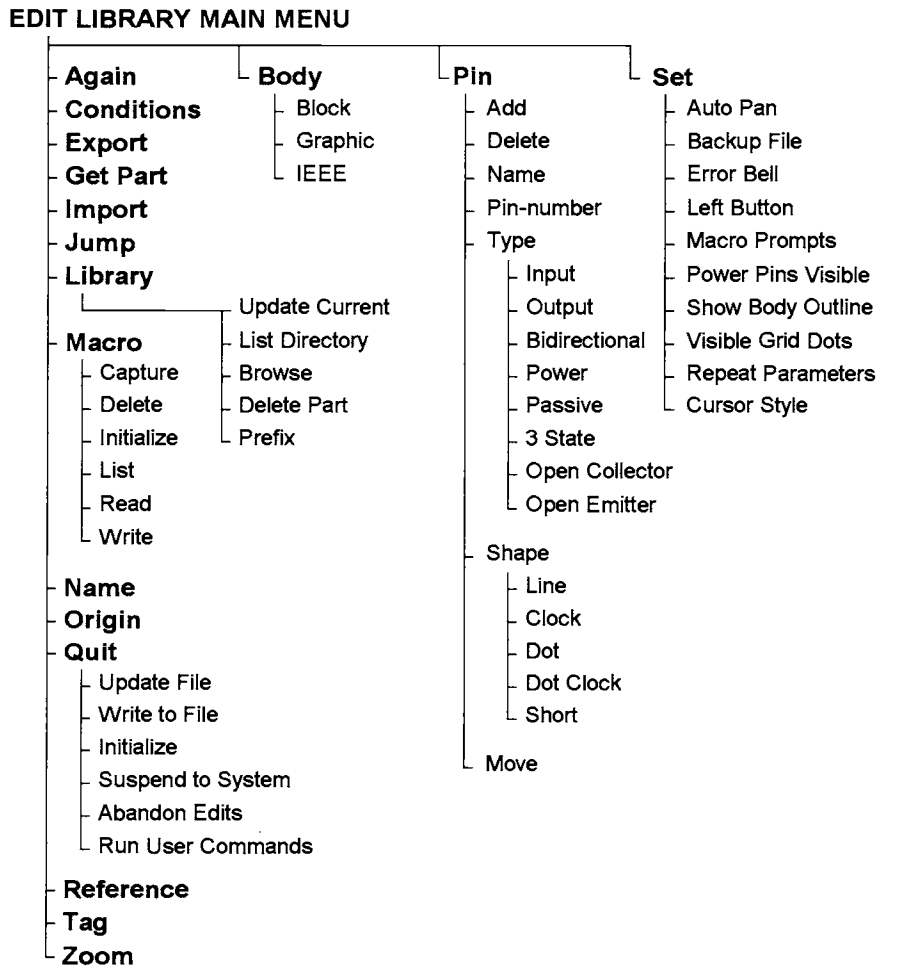


Figure 12-3 Library Editor Menu Structure

Library Editor Main Menu

An overview of the main menu commands shown in Figure 12-3 follows. Commands that have extensive sub-menus are explained in subsequent sections. For configuration details, refer to chapter 2 starting on page 50. For more detailed information about using the library editor, refer to chapter 6 starting on page 205.

Again — Repeats the last command, but only repeats the first menu level. For example if the last command was Body Graphic, using Again would only get you to the Body menu.

Body — Body commands are used to draw and edit the body of a part. This command has separate sub-menus for block, graphic, and IEEE type parts.

Conditions — Shows information about the part being created. This information is of little practical use.

Export — Writes data for the current part to a selected file. Useful for transferring part data to another library. Note that the part must have a name before using Export. Data is formatted as a library source file.

Get Part — Accesses the library and retrieves a part for editing. Double click on this command to bring up a scroll list of available parts.

Import — Reads part data from a library source file that was written using Export. Useful for transferring part data from another library.

Jump — Moves the cursor to a preset tag location (see Tag command), to a grid reference, or along the X or Y axis. Forget about this command and use your mouse.

Library — Accesses special library functions used to update the current library, list parts, browse through the library, delete a part, and define prefixes.

Macro — Used to capture macro commands and manage macro files. Not required to access the macro file automatically loaded at startup (defined in the SDT configuration)

Name	Used to add, delete, or edit names and assign prefixes to a part. Note that a part may have multiple names and prefixes.
Origin	Resets the X,Y origin at the current cursor position. Limited usefulness.
Pin	Commands on the Pin menu are used to add, delete, and edit the part pins.
Quit	Similar to the Quit command in Draft. Provides access to file management functions. Also used to exit from the library editor.
Reference	Used to specify and edit the reference designator prefix of a part.
Set	Allows changing parameters for numerous library editor options including display of body outlines, visible power pins, and grid dots.
Tag	Sets up to eight tag locations for use with the Jump command. Not frequently used.
Zoom	Used to change the zoom scale. Normal scale is 1, which is adequate for most part editing. In some cases, zooming in on a detailed part may facilitate editing.

Library Editor Body Command Sub-menus

The type of part determines the available options when the Body command is selected as shown in Figure 12-4.

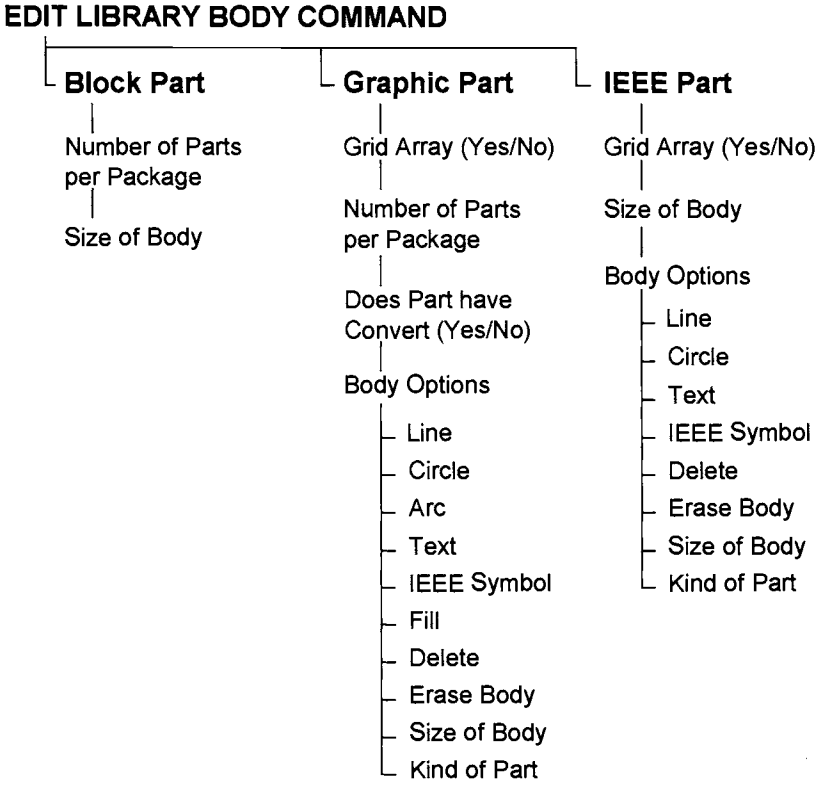


Figure 12-4 Library Editor Body Command Options

Graphic parts have the widest range of options. Use graphic parts to create discrete components such as capacitors, diodes, resistors, and transistors. You can also use graphic parts for digital logic (such as AND and OR gates) and analog ICs (such as comparators and op-amps). Block parts have a rectangular outline and are limited to two options: number of parts per package and size. Use block parts for complex ICs. For more details on the various types of part bodies, refer to chapter 6 page 206. The following Body command options are available:

Number of Parts per Package Up to 16 parts per package can be specified for multiple gate parts. Zero parts per package is used as a special option to

	suppress appearance of pin numbers on the schematic for discrete components such as resistors.
Grid Array	If the grid array option is selected, OrCAD allows alphanumeric pin numbers.
Does Part have Convert	If the convert option is selected, OrCAD allows definition of an alternate body style that can be selected via the Edit command in Draft. Commonly used for DeMorgan equivalents of digital logic gates.
Size of Body	The mouse cursor is used to place two corner points defining the size of the part body. Block parts are represented by a rectangular outline. The Size Of Body option allows changing the body outline size. The mouse cursor drags the lower right hand corner. Click to place the corner in a new location. Unpredictable results can occur if the outline size of an existing part is reduced.
Line	Draws a line. Functions similar to the Place Wire command in Draft.
Circle	Draws a circle within the part outline. Start by clicking the mouse cursor on the center point. The circle then expands as the cursor is moved out. Click the mouse again to define an edge point.
Arc	Draws an arc ranging from zero to 90 degrees within the part outline. Start by clicking on the center as with the Circle command. Initially a circle appears and expands as the cursor is moved out. Click the mouse again to define the first point on the arc and repeat for the second point. Note that the endpoints of the arc must remain within the same quadrant (defined by imaginary horizontal and vertical lines passing through the center point of the arc).

Text	Draws text within the part outline. Functions similar to the Place Text command in Draft. Primarily used for symbolic labels, such as the + polarity label on an electrolytic capacitor. Do not used the Text command for pin numbers, pin names, part name, or reference designator prefix. These entities are separately defined.
IEEE Symbol	Draws IEEE/ANSI symbols within the part outline. A menu appears with a listing of available symbols. IEEE symbols are no longer widely used and are not reviewed in detail in this book.
Fill	Draws a solid fill pattern within an enclosed area. Position the cursor within the enclosed area and then select the Fill command. The triangular filled in area of a diode symbol is created using this command. Note that drawing or editing any graphics on a part will cause all fill patterns to be deleted. Use Fill only as a final step after all other graphics are completed.
Delete	Deletes the graphic object. Functions similar to the Delete command in Draft. Note that unlike Draft, the library editor does not provide an undo option for Delete.
Erase Body	Deletes all objects within the part body. Note that an undo option is not provided.
Kind of Part	Changes part type (block, graphic, or IEEE). Also used to access additional body parameters for each part type. Note that changing from one type of part to another (for example graphic to block) generally causes any previous changes and part data to be lost.

Library Editor Library Command Sub-menu

The Library command accesses special functions used to update the current library, list parts, browse through the library, delete a part, and define prefixes. Note that in order to save a newly created or edited part in the library file, you must first use the Library Update Current command and then the Quit Update File command. The following Library command options are available:

Update Current — This command option must be used when a part has been created or edited in order to save the part information into the library. Only the library data in RAM memory is updated with this option. You must also use the Update File option of the Quit command to save the library to disk. **Both steps must be completed or data will be lost.**

List Directory — Lists all parts in the library. A sub-menu allows listing the directory to the screen, the printer, or a file.

Browse — Can be used for scrolling through the list of available parts. Browse cannot retrieve parts for editing. Note that the Get command also allows scrolling through the list of parts.

Delete Part — Used to delete a part from the library. The part name (suffix only for parts with defined prefixes) can be directly entered. If the <ENTER> key is pressed without specifying a part name, the entire parts list is displayed. You can then scroll through the list and select the part to be deleted. If a part has multiple names, you must delete every part name in order to delete the part itself.

Prefix — Used to edit the list of available prefixes. The existing list is displayed and a sub-menu provides Add, Delete, Edit, and Quit options. The list can contain up to 16 prefixes. The Quit option saves changes and returns you to the Library menu.

Library Editor Macro Command Sub-menu

Macro commands improve productivity by reducing the number of keystrokes and mouse clicks required to carry out common, repetitive tasks. Macro commands are stored in macro files on disk. The Macro command is used to capture macros and manage macro files. A macro file and initial macro command can be defined in the SDT configuration. This macro file is then automatically loaded whenever Edit Library is launched. For more details on configuring a startup macro, refer to chapter 2 page 52. For more detailed information about using the Macro command, refer to chapter 8 page 308. The following options are available on the Macro menu:

Capture	Used to capture (define) a macro command from within Edit Library. Prompts for the macro name followed by the sequence of commands that make up the macro. Any valid sequence of keystrokes and mouse movements and clicks is recorded. Press <CTRL> <END> to stop the capture command. The macro command is written to the macro buffer.
Delete	Used to delete a macro command from the macro buffer. Prompts for the macro name key or key combination.
Initialize	Clears the macro buffer.
List	Lists the names of all macro commands available in the macro buffer. The List option does not display any information about the macros.
Read	Loads a new macro file into the buffer. Prompts for the macro filename.
Write	Saves the macro buffer contents to a file. Prompts for the filename. The macro buffer only stores macro definitions, not comments. If you use Macro Read to read a macro file (or OrCAD loads an initial macro at startup) and then use Macro Write, any comments are lost.

Library Editor Pin Command Sub-menu

The pin command is used to add, delete, and edit pins. The command functions in essentially the same manner for graphic, block, or IEEE body parts. When a pin is added, the library editor automatically steps through the process and prompts for the information required for each step. Unlike part body parameters, individual pin attributes can easily be edited using the available command options. Except for the Move option, the cursor should be located at the desired position (where a pin is to be added, deleted, or edited) before using the Pin command. The following options are available with the Pin command.

Add — Adds a pin at the cursor location. The library editor then prompts for the name, number, type, and shape attributes (these are explained in detail below).

Delete — Deletes the pin at the cursor location. Note that an undo function is not available.

Name — Edits the name of an existing pin. All pins must have a defined name. Pin names are not visible on graphic parts.

Pin-number — Edits the number of an existing pin. Pin numbers are only visible on graphic parts if the part is defined as having zero parts per package.

Type — Edits the electrical type definition of a pin. The Check Electrical Rules tool uses pin type information to check for electrical interconnection rule violations as defined by a decision matrix (refer to chapter 5 page 174). A menu of pin type options appears as shown in Figure 6-24 and the current type is listed at the top of the display. Available pin types include:

Input. Signals are applied to a part via input pins. For example, pin 1 of a 74HCT14 hex inverter is an input pin.

Output. The part generates a signal at output pins. For example, pin 2 of a 74HCT14 hex inverter is an output pin.

Bidirectional. The pin can be either an input or output, depending on the internal state of the part. For example, pin 2 of a 74HCT245 bus transceiver is a bidirectional pin.

Power. Supplies power to the part and is automatically connected to ground or one of the supply rails. Power pins are invisible in Draft. The name of the power pin, such as GND, VCC, VSS, or VDD, determines which power or ground plane the pin is connected to.

Passive. Typically used with passive devices such as resistors or discrete semiconductors without external power connections. You can also use passive pins for visible power connections on ICs.

3 State. Special logic output signal with three states: active low, active high, or off (high impedance). For example, pin 2 of a 74HCT373 latch is a three state pin.

Open Collector. Special logic output signal that can only sink current (such as the collector of a NPN transistor with emitter grounded). Requires an external pull up resistor. Multiple open collector outputs can be "wired OR" together. For example, pin 2 of a 7406 hex inverter is an open collector pin.

Open Emitter. Special logic output signal that can only source current (such as the emitter of a NPN transistor with collector tied to VCC). Primarily found in high speed ECL logic, such as the Motorola MECL 10K/10H logic series. Requires an external resistor network for proper termination.

Shape

Edits the shape of a pin. Note that the shape of a pin has no effect on electrical properties. Pin shapes are shown in Figure 6-25. The current shape is listed at the top of the display. Available pin shapes include:

Line. A normal pin with a length of three grid units. Except for the short pin, all other pins are three grid units long.

Clock. A pin with the triangular clock symbol appearing inside the part outline. Used for clock inputs on logic devices.

Dot. A pin with the circular negation symbol appearing outside the part outline. Used for inverted inputs (active low) on logic devices. Also handy for drawing switches.

Dot Clock. A pin with both the circular inversion symbol and triangular clock symbol. Use for inverted clock inputs on logic devices.

Short. A normal pin with a length of one grid unit. Primarily used with passive and discrete devices drawn as graphic parts.

Move — Used to move the location of an existing pin. Unlike the other Pin commands, start the command first. The Move command then prompts for the pin to be moved. Click on the selected pin. Then move the cursor and click on the desired location.

Repeat — Automatically repeats the last pin definition, using the X,Y location step sizes and name and number "delta" values as determined by the Set command Repeat Parameters option. Useful for adding sequential address and data pins on large ICs.

Users are often bewildered by the range of pin type options. Some tips related to part pins are given below. In addition, general rules for part layout are given in chapter 6 on page 236.

- If you are creating a part and are uncertain about which pin type is correct, use passive pins. Passive pins are general purpose and never cause electrical rules violations when connected to other types of pins or objects.
- Invisible power pins can be a real nuisance on modern designs using multiple power levels. An easy solution is to import standard logic parts into your custom library and edit the power pins to assign them a passive type attribute. This makes the power pins visible and later allows directly connecting them to a particular supply voltage.
- OrCAD uses two conventions to represent inverted signals. In Draft, labels and module ports use the backslash (\) convention for denoting inverted (or active low) signals. In Edit Library, pin names for inverted signals use the overbar ($\overline{\ }$) convention. The overbar convention is also used by most IC manufacturers to represent inverted inputs on device data sheets. To enter a

pin name with an overbar, type a backslash after every character (for example C\L\K\).

- Pins on block parts such as ICs always have both a defined pin name and pin number. The pin name describes the pin function and the pin number is determined by the physical location of the pin on the device package.
- Pins on graphic parts have a defined pin name, that is always invisible. Pins on graphics parts can have a visible pin number. For graphic parts that display pin numbers, suggested practice is to also use the same number for the pin name. This will avoid possible problems and confusion when generating a netlist

Library Editor Quit Command Sub-menu

The Quit command is similar to the File menu found in Windows programs, Quit provides access to file management functions. Note that in order to save a newly created or edited part in the library file, you must first use the Library Update Current command and then the Quit Update File command. The following options appear on the Quit menu:

Update File	Writes the library data to disk. Remember that you must use the Library Update Current command first. Always use both commands to save your work to disk **before** exiting via Abandon Edits. Good practice is to save your work to disk after every part creation or edit to save recent work in case of system crash or power failure or in case things don't work out on the next edit (the library editor has no undo capability).
Write to File	Same action as Update File except this option is used to write data to a specified file.
Initialize	Abandons any edits since the last file update and then prompts for the name of a new library.
Suspend to System	Suspends program operation and loads the DOS command interpreter to allow DOS operations. An additional > system prompt appears as a reminder that OrCAD is still in the background. Type EXIT<ENTER> to exit DOS and return to the library editor program.
Abandon Edits	Ends the editing session and exits back to the main SDT screen. Does not save any edits. You must use

Update File before Abandon Edits in order to save your work. If any edits were made during the session, Abandon Edit asks for a YES/NO confirmation before exiting.

Run User Commands Similar to Suspend To System, except exits to DOS and runs the LIBUSR command in the design directory or on the DOS path. This can be a DOS batch file named LIBUSR.BAT. Automatically exits DOS and returns to the library editor after the LIBUSR command is completed.

Library Editor Set Command Sub-menu

The Set command is used to set certain library editor display and editing options. Note that most of the menu options are followed by a YES/NO indication that shows the status of the option. Options which do not include a status indication have an additional sub-menu. The Set command menu for Edit Library has many of the same options as in Draft with several new options specific to editing library parts. A detailed description of the various Set options follows:

Auto Pan Enables automatic panning (movement of the display window relative to the sheet) when the mouse is moved to the edge of the display window. An archaic option from the days of slow 286 and 386 class PCs. Turning Auto Pan off slightly speeds up the display, but makes moving around the sheet very difficult. Always set Auto Pan to YES.

Backup File Enables automatic creation of a backup file whenever the Quit command is used to write to or update a library file that already exists. The original file is saved with the filename extension .BAK. Always set Backup File to YES. If a serious error, system crash or other unforeseen event occurs, the backup file allows recovery of data.

Error Bell Enables the error beep tone. OrCAD has an annoying tendency to flag even relatively innocuous actions as errors. For example, trying to select an object for editing and not having the mouse cursor within the object border. When an error occurs, everything is locked up for about one second and you must then click the right mouse button to escape. The error beep

tone helps you recognize the occurrence of an error.

Left Button Enables automatic generation of an <ENTER> keystroke when the left mouse button is released. If this option is disabled, many commands will require double mouse clicks.

Macro Prompts Displays the sequence of commands contained in the macro when a particular macro is executed. This is another archaic option related to graphics performance. Always set Macro Prompts to YES. If Macro Prompts is set to NO, screen redraw during macro execution is disabled and any macros using pan or zoom will fail.

Power Pins Visible Displays power pins that are normally invisible. Recommended practice is to always set this option to YES so that power pins are not inadvertently deleted or duplicated. In some cases text associated with power pins may overlap other pin names and turning power pin display off is advantageous.

Show Body Outline Displays the body outline as a dotted line. Graphic parts have an overall body outline (the term border is more appropriate). Editing outside the body outline can cause unpredictable results. Always display body outlines when editing parts.

Visible Grid Dots Enables display of grid dots. Visible Grid Dots should always be set to YES. At the normal zoom scale 1, the spacing between grid dots is 1/10 of X and Y units. For example, configuring SDT for inch units results in 10 grid dots per inch at zoom scale 1.

Repeat Parameters This command has a sub-menu with four options used to set parameters for the Pin command Repeat option. Allows automatic placement and incrementing of pin names and numbers. The options include:

X Repeat Step. Sets the X axis step size in grid units.

Y Repeat Step. Sets the Y axis step size in grid units.

Name Delta. Sets the pin name numeric suffix step, that is step size 1 results in pin names such as A0, A1, A2, and so forth.

Pin Number Delta. Sets the step size for automatic sequential pin numbering.

Cursor Style — Selects the cursor style: either an "arrow" pointer or a crosshair.

Overview of OrCAD SDT Tools

All OrCAD SDT tools have associated local configurations. With the exception of Draft and Edit Library, which have a graphical user interface with "pop up" menus, the action of the remaining tools is entirely controlled by these local configurations. Detailed information about tool usage and local configuration is presented in the preceding chapters. The index below summarizes the function of the various OrCAD tools and references page numbers in this book where more detailed information can be found. In addition to the local configurations, global configurations exist for OrCAD ESP (the graphic environment used to select tools) and SDT (the schematic drafting tool set).

ESP — Graphic user environment used to launch OrCAD SDT and individual SDT tools. Global configuration options consist mainly of video driver setup and screen colors for the initial ESP screen and the main SDT menu shown in Figure 12-1. Refer to chapter 2 page 44.

SDT — Schematic Drafting Tools. Global configuration affects all OrCAD SDT tools. Configuration options include video, printer, and plotter drivers, parts library selection, worksheet setup, schematic color, plotting options, key fields, and electrical rules matrix. The video driver setup applies only to Draft and Edit Library. Refer to chapter 2 page 46. Additional information on key fields

	configurations is given in chapter 7 on page 286.
Annotate Schematic	Automatically assigns reference designators and reports last used reference designators. Refer to chapter 5 page 170.
Archive Parts in Schematic	Creates a special library to permanently archive all parts used in a schematic design. The archived library can be added to the configured library list. The design then becomes independent of the main libraries. Refer to chapter 6 page 240.
Back Annotate	Automatically updates schematic reference designators that have changed during the PCB design phase due to gate swapping or spatial sequencing requirements. Uses a "was-is" file to update the changed reference designators. Refer to chapter 7 page 299.
Check Design Integrity	Automatically runs three individual tools in sequence: Cleanup Schematic, Cross Reference Parts, and Check Electrical Rules. Refer to chapter 7 page 265.
Check Electrical Rules	Checks the entire design for possible violations of basic electrical connectivity rules. Uses a decision matrix to analyze all possible combinations of connections. The decision matrix is defined as part of the OrCAD SDT configuration. Refer to chapter 5 page 174. Also refer to the Check Design Integrity tool.
Cleanup Schematic	Corrects minor drafting errors such as overlapping wires, parts off grid, and improper labels. Also used to remove error objects after running the Check Electrical Rules tool. Refer to chapter 4 page 134. Also refer to the Check Design Integrity tool.
Compile Library	Once used to compile parts libraries prepared off-line using a text editor. Now obsolete. Use Edit Library tool for direct graphic

	editing instead. The Compile Library tool is briefly reviewed in chapter 6 on page 244.
Convert Plot to IGES	Translates generic OrCAD vector plot data to IGES data for transfer to other CAD systems. Requires a two step process. Largely obsolete. In most cases, use of DXF and/or HPGL data that can be directly generated by the Plot Schematic tool gives better results. The Convert Plot To IGES tool is briefly reviewed in chapter 5 on page 182.
Create Bill of Materials	Automatically creates an ASCII file containing bill of materials information from parts data on the schematic. Refer to chapter 4 page 138.
Create Hierarchical Netlist	Automatically creates a hierarchical netlist of circuit elements and interconnections. Primarily used for transfer of data to specialized EDA tools discussion of which are beyond the scope of this book. Not commonly used for PCB design or circuit simulation. General information on netlists is given in chapter 5 starting on page 187.
Create Netlist	Automatically creates a flat netlist of circuit elements and interconnections. Primarily used for transfer of data to PCB design and circuit simulation tools. Refer to chapter 5 page 191.
Cross Reference Parts	Used to locate certain errors in reference designator and module port assignments and to report unused gates in multiple gate IC packages. Refer to chapter 4 page 143. Also refer to the Check Design Integrity tool.
Decompile Library	Once used to decompile parts libraries prior to off-line editing using a text editor. Now obsolete. Use Edit Library tool for direct graphic editing instead. The Decompile Library tool is briefly reviewed in chapter 6 on page 244.

Draft	Main OrCAD schematic drafting tool. Used to create and edit schematics. Minimal local configuration because all major options are set as part of the SDT global configuration. For configuration details, refer to chapter 2 starting on page 46. Tutorials on using Draft start in chapter 3 on page 65.
Edit File	Launches the Stony Brook M2EDIT text editor supplied with OrCAD. Can be used to examine error files. The editor's overall usefulness is limited by lack of printing capability and mouse support. No local configuration options exist. Refer to chapter 7 page 260. Installing a User Button to launch the user's favorite DOS based text editor is recommended as an alternative.
Edit Library	Graphic parts library editing tool. Used to create and edit library parts. Refer to chapter 6 starting on page 205.
List Library	Generates a listing of parts in a library. This tool duplicates the functionality of the Library command available within Edit Library. Refer to chapter 6 page 239.
Plot Schematic	Sends vector plot data for a schematic design to a pen plotter or file. Also used to create DXF files for transfer to AutoCAD and other CAD systems. Refer to chapter 5 page 181.
Print Schematic	Sends raster plot data for a schematic design to a printer. Refer to chapter 3 page 109. Additional information is presented in chapter 5 on page 181.
Select Field View	Modifies visibility attribute of parts fields on a global basis for an entire schematic design. Refer to chapter 7 page 297.
Show Design Structure	Generates a report that shows the hierarchical structure of a design. Refer to chapter 4 page 146.

View Reference	Provides on-line access to OrCAD reference materials in the ORCADESP\SDT directory. Displays a selection list of available references and then launches the Stony Brook M2EDIT text editor as a viewing tool. Refer to chapter 7 page 262
Update Field Contents	Automatically updates parts fields in a schematic design based on data contained in a "stuff" file. Refer to chapter 7 page 292.
User Buttons	OrCAD provides four user buttons that can launch DOS commands or external programs, such as a text editor. Refer to chapter 8 starting on page 305.

Conclusion

In addition to the information provided in this chapter, the reader is encouraged to refer to the index. Efforts have been made to provide an extensive index to allow easy access to information required to answer just about any question.

Appendix A

Plotter Information

Interfacing to Hewlett-Packard Pen Plotters

While many companies have switched to newer inkjet plotters for large format hard copy, Hewlett-Packard (HP) pen plotters are still widely used. Most HP pen plotters use an RS-232 serial interface. Users often encounter problems with these serial interfaces. While the following discussion focuses on HP plotters, the general principles apply to all plotters.

The area that seems to cause the most confusion and problems relates to the communications handshaking employed in the RS-232 link to the plotter. The computer system is typically capable of sending data at a much faster rate than can be accepted by the plotter. Most plotter models also have a limited buffer memory, sometimes no more than 1KB, for plot data storage. Proper setup of the communications handshake assures that the plotter data buffer does not overflow. Two schemes are used: hardware handshaking and software handshaking. The situation is further complicated by the fact that some systems use both schemes depending on the application that is sending data to the plotter. A typical example is a PC workstation running both AutoCAD and OrCAD. AutoCAD uses software handshaking and OrCAD requires hardware handshaking.

Hardware Handshaking

All RS-232 communications links require at least three signal wires. The minimum connections include TXD (transmitted data), RXD (received data), and SIGNAL GROUND. Hardware handshaking requires additional signals that control data flow. Refer to Figure A-1 for a PC XT serial port to HP plotter cable. The hardware handshake signals are CTS (clear to send), DSR (data set ready), and DTR (data terminal ready). The computer senses the CTS and DSR lines that are connected to the DSR signal from the plotter. When the plot data buffer is empty or below a certain lower threshold, the plotter asserts the DTR signal. The computer will then start to send data. When the plot data buffer fills beyond an

upper threshold, the DSR signal is no longer asserted, and data transmission stops until the buffer again drops below the lower threshold.

The cables shown in Figures A-1 and A-2 implement hardware handshaking. Note that these cables are for full bidirectional data communications. In some cases data is only transferred from the computer to the plotter and the return data path is not used.

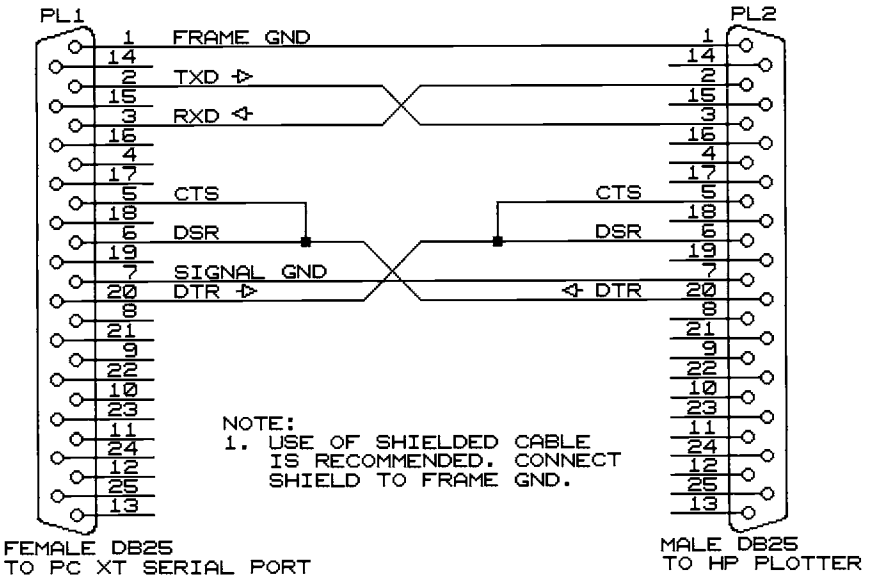


Figure A-1 PC XT Plotter Cable for Hardware Handshaking

Plotting on-line directly from OrCAD requires hardware handshaking. If you connect an HP plotter to your PC workstation and plot directly from OrCAD, you must use one of the cables shown in Figures A-1 and A-2. The choice of cable depends on whether your PC uses 9 pin or 25 pin connectors. The 25 pin connector originated on the PC XT and the 9 pin originated on the PC AT. Some multi-purpose IO cards include one serial port with a 25 pin connector and a second port with a 9 pin connector. In the PC environment, the term "COM" port refers to an RS-232 serial port.

Cables such as those shown in Figures A-1 and A-2 are available from HP. Common RS-232 cables sold in computer stores for connecting an external modem are unlikely to have the correct signal connections. If you are handy with a soldering iron, you can make your own cable. You should be able to find the connectors and cable at Radio Shack. Shielded cable is preferred, but not required for most installations unless the cable is very long. The FRAME

GROUND signal to pin 1 in Figure A-1 is not required, but may help in noisy environments. No FRAME GROUND signal exists on PC AT style 9 pin RS-232 connectors.

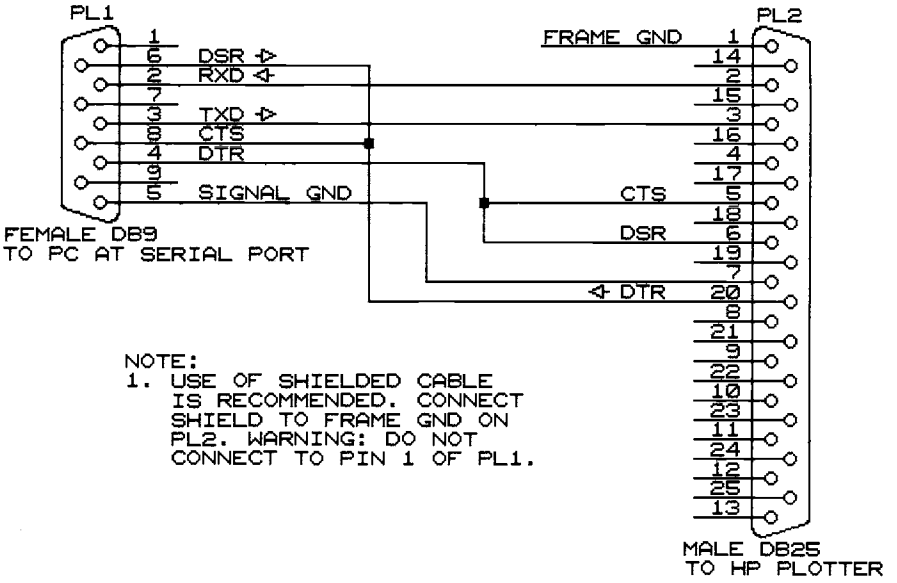


Figure A-2 PC AT Plotter Cable for Hardware Handshaking

All HP plotters that the author has encountered are compatible with hardware handshaking and these plotters do not require any special setup.

If the plotter is connected to a different workstation than the one used for running OrCAD, plot files can be transferred via floppy disk or a network. Users have reported some compatibility problems with OrCAD plotting directly to network devices on Novell networks. In a network environment or when transferring files via floppy disk, configure OrCAD to plot to disk. Then exit to DOS and use DOS commands to setup the COM port and copy the file to the appropriate device.

For example, assume that the plot file is named MYFILE.PLT. Assume that the plotter is attached to COM2 and requires data transmission at 9600 baud with no parity, 8 data bits, and 1 stop bit (the most common default). The required DOS commands are:

MODE COM2: 9600,N,8,1,P<ENTER>

COPY MYPLOT.PLT COM2: /B<ENTER>

The Mode command sets up the COM2 serial port. The Copy command copies the file. The /B option assures all data is copied even if it is not ASCII text. You only need to issue the Mode command once. Subsequent files can be copied by repeating the Copy command with the appropriate plot filenames.

In a network environment, the local PC running is typically configured to "map" a certain COM port to the remote plotter. You do not require a Mode command prior to copying files because communications setup is handled by the network.

Software Handshaking

Software handshaking is also known as XON/OFF handshaking. The RS-232 communications link used with software handshaking requires only three signal wires: TXD (transmitted data), RXD (received data), and SIGNAL GROUND. The term "three-wire RS-232 interface" is sometimes used for this type of link. Data flow control signals are jumpered at the connector ends so that communications is always enabled. Refer to Figure A-3 for a PC XT serial port to HP plotter cable. Note that CTS (clear to send) is jumpered to RTS (request to send) and DSR (data set ready) is jumpered to DTR (data terminal ready).

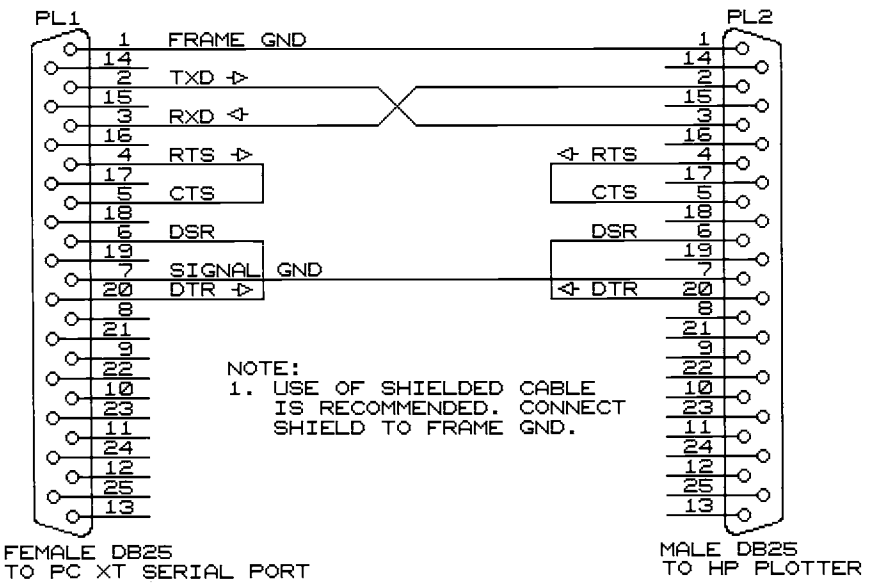


Figure A-3 PC XT Plotter Cable for Software Handshaking

Special communications drivers at both ends (computer and plotter) are required for software handshake. When the plot data buffer is empty or below a certain lower threshold, the plotter will send an XON (transmission on) character to the

computer. The computer will then start to send data. When the plot data buffer fills beyond an upper threshold, the plotter sends an XOFF (transmission off) character, and data transmission stops until the buffer again drops below the lower threshold. The default ASCII characters are DC1 (decimal equivalent 17) for XON and DC3 (decimal equivalent 19) for XOFF.

The cables shown in Figures A-3 and A-4 implement software handshaking. Full bidirectional data communications is always required for software handshake. As mentioned before, common RS-232 cables sold in computer stores are unlikely to have the correct signal connections and you may have to make your own cable.

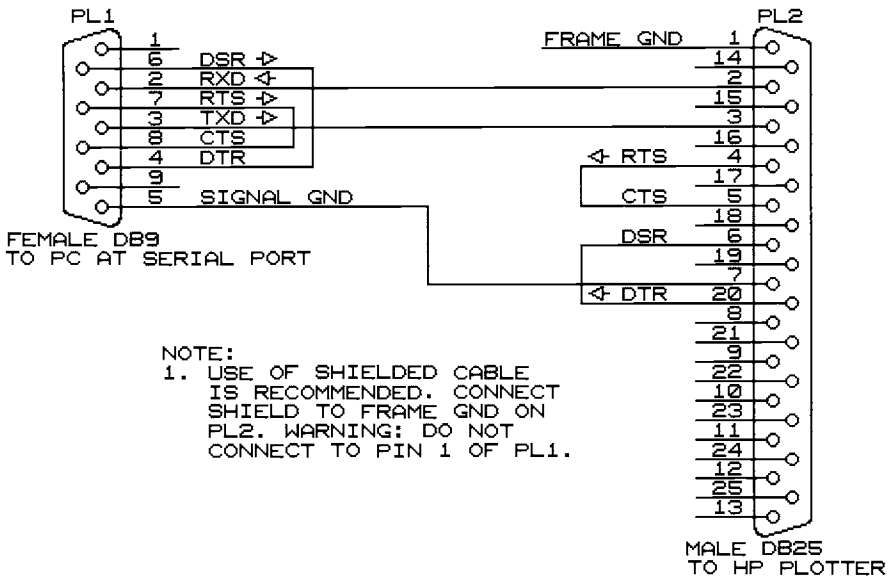


Figure A-4 PC AT Plotter Cable for Software Handshaking

Plotting in Environments Running OrCAD and AutoCAD

OrCAD and AutoCAD are the market leaders in their respective segments. Environments where both OrCAD and AutoCAD run on the same PC workstation and interface to the same plotter are not uncommon.

AutoCAD uses software handshaking. The AutoCAD installation instructions call for the cables shown in Figures A-3 and A-4. Consequently, you will typically find this type of plotter hookup on PC workstations running AutoCAD. You **cannot** plot directly from OrCAD using this hookup. Two possible workarounds exist:

- OrCAD requires hardware handshaking when plotting directly on-line. You may be able to use one of the hardware handshaking cables shown in Figures A-1 or A-2 with AutoCAD. Based on the author's experiments, this appears to work. The extra control lines become redundant when plotting direct from AutoCAD which requires only three-wire RS-232 communications.
- Plot from OrCAD to a file and then use a communications program such as PROCOMM PLUS which supports software handshaking to transfer the file to the plotter. This process is explained in more detail in the next section. In this case, you would continue to use the AutoCAD recommended three-wire RS-232 communications cable for interfacing to the plotter.

Using PROCOMM PLUS to Transfer Plot Files

If your environment dictates three-wire RS-232 communications to the plotter, you can use PROCOMM PLUS to transfer OrCAD plot files. PROCOMM PLUS is available from Datastorm Technologies Inc., Columbia, MO, phone: 314-443-3282. Both Window and DOS versions are available, but for this file transfer application, the DOS version is recommended.

The following steps provide an overview of the file transfer process. The example used is for an HP plotter, but the same general principles will apply to other plotters:

1. Configure OrCAD to plot HPGL data to disk. Use the Plot Schematic tool to generate plot files. Then exit to DOS.
2. Use an ASCII text editor to strip out the following OrCAD generated plotter initialization string at the beginning of each plot file:

   ESC.@1000;1:

   The ESC character may appear as a left arrow or ^[on some text editors because it is a nonprinting control character.
3. Determine the plotter port setup parameters. The AutoCAD defaults for HP plotters are: 9600 baud with even parity, 7 data bits, and 1 stop bit.
4. Launch PROCOMM PLUS. Make the port settings the same as given above. Setup terminal options for: half duplex, software flow control to ON, and hardware flow control to OFF. Set ASCII transfer options for: 15 millisecond character pacing and 1 second line pacing (if this seems to plot slowly, you can try smaller pacing values).
5. Type in the following setup string. This is sent to the plotter to establish software handshaking. The particular string given below was successfully tested on several HP plotters. It is based on the initialization string AutoCAD

sends to HP plotters with a higher XOFF threshold to prevent buffer overflows (refer to your plotter documentation for details):

ESC.(;ESC.I500;;17:ESC.N;19:IN;SC;PU;

Note that the ESC character is generated by simultaneously pressing the <CTRL> key and the [(left bracket) key. This sometimes appears in documentation as ^[where the ^ (caret) character is used to represent the <CTRL> key. However, you must use the <CTRL> key and not the ^ key. Also note that the parameter after the second ESC is the letter I followed by the number 500. There are no spaces in these commands. This initialization string is contained in the file HPINIT.TXT on the disk supplied with the book. You can upload this file once at the beginning of a plotting session to initialize the plotter. Use ASCII file transfer to upload the file.

6. Use the PROCOMM PLUS ASCII file transfer to upload the plot files to the plotter.

Plotting Tips

Large HP plotters locate the plot origin (zero X and Y coordinate) in the center of the paper. This causes considerable problems with the interaction of paper size, plot size, and offsets. You must configure SDT with negative plot offsets on the template table (refer to chapter 2 page 55). The plot offsets should typically be set at -1/2 the value of horizontal and vertical dimensions. For example: a C-size drawing with 20.2 inch horizontal dimension requires a -10.1 inch plot X offset. You may have to fine tune the offsets to precisely center the plot on the paper. You also have to allow for border areas. Often the easiest approach is to load a sheet of paper substantially larger than the intended plot, set approximate offsets, and then trim the resulting plot.

OrCAD schematics utilize fine text and usually include graphic parts with small details. Getting good pen plotting results requires using quality media and pens. Forget about using fiber tip pens and cheap bond paper. Use good grade velum or mylar media and disposable liquid ink pens. HP and Staedtler Mars liquid ink pens seem to offer the best resistance to skipping and clogging. Pen sizes ranging from .25mm to .35mm seem to give the best overall appearance.

Interfacing to Hewlett-Packard DesignJet Plotters

HP recently introduced the DesignJet family of large format inkjet plotters. While OrCAD does not offer drivers specifically for these plotters, family members such as the DesignJet 650C maintain HPGL compatibility. Tests

revealed that the standard OrCAD HPGL driver appears to work just fine with the DesignJet 650C configured for 7586, HP-GL2 compatibility. When plotting direct from OrCAD in a network environment, occasional problems were encountered with DOS errors. Changing to off-line plotting and using the DOS copy command to transfer plot files eliminated the problem.

Appendix B

Inside OrCAD Disk Information

Disk Contents

This book includes a disk that contains custom OrCAD macro and library files, useful utilities, and sample files for the tutorial exercises:

BOMSORT.EXE	Bill of materials sort utility (chapter 11)
CUSTOM.LIB	Custom library (installed in chapter 2)
CUSTOM.MAC	Custom macro (installed in chapter 2)
HPINIT.TXT	HP plotter initialization string (appendix A)
LIST91M.EXE	Shareware list utility self extract (chapter 4)
NETSTRIP.EXE	Netlist strip utility (chapter 10)
TUTOR1 <DIR>	Sample files for Tutorial 1 (chapter 3)
TUTOR2 <DIR>	Sample files for Tutorial 2 (chapter 4)
TUTOR3 <DIR>	Sample files for Tutorial 3 (chapter 5)
TUTOR4 <DIR>	Sample files for Tutorial 4 (chapter 6)
TUTOR5 <DIR>	Sample files for Tutorial 5 session 1 (chapter 7)
TUTOR5A <DIR>	Sample files for Tutorial 5 session 2 (chapter 7)
TUTOR7 <DIR>	Sample files for Tutorial 7 (chapter 9)
TUTOR8 <DIR>	Sample files for Tutorial 8 (chapter 10)
TUTOR9 <DIR>	Sample files for Tutorial 9 (chapter 11)
README.TXT	Text file with any updated information not available when the book was printed

The sample files for the tutorial exercises files are intended to help complete the exercises and to serve as examples of finished designs. In some cases, fine details on schematic figures within this book may be hard to make out because of the reduced size. You can use OrCAD to print out the sample schematic files for improved readability.

Requirements and Compatibility

The files are supplied on a MS-DOS formatted 1.4MB 3.5 inch floppy disk. The three utility programs should run on any MS-DOS/PC-DOS compatible computer capable of running OrCAD SDT 386+ schematic drafting software. At least 5MB of free hard disk space is recommended to allow for the files created during the tutorial exercises.

Shareware List Utility

The self extracting archive file LIST91M.EXE contains the LIST.COM shareware file list utility and associated documentation. The copyright for this material belongs to Vernon Buerg. Any use, including use as part of the tutorial exercises in this book, requires payment of a registration fee and is subject to the software license in the program documentation.

You can install LIST.COM and the associated documentation files on your root directory or any subdirectory on your PC's path. For example, to install the files from floppy disk on your A: drive to the root directory on your hard drive, type:

A:<ENTER>

COPY LIST91M.EXE C:\<ENTER>

C:<ENTER>

LIST91M<ENTER>

The list utility can then be run by typing **LIST<ENTER>** at the C: prompt. This will bring up a file directory and you can highlight the file to list by using the cursor keys and then pressing **<ENTER>.** You can also directly list a specific file by including the filename with the command, that is by typing **LIST filename.ext<ENTER>.** Refer to the documentation supplied with the program for more details.

Index