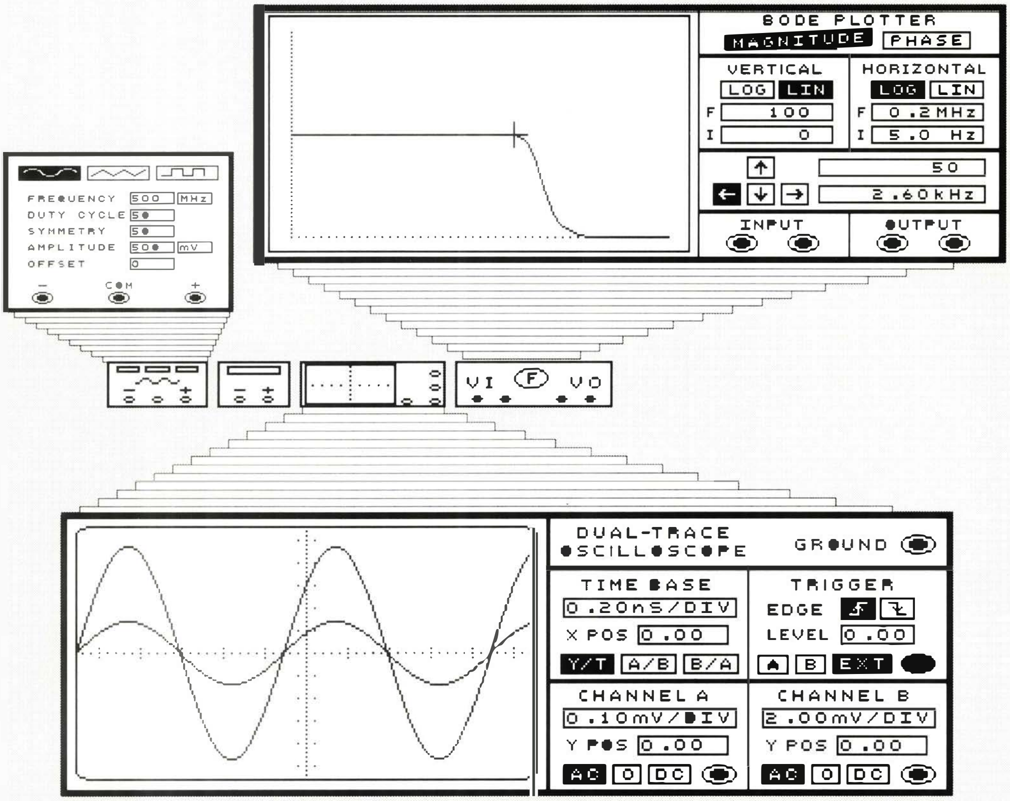
Electronics Workbench™

The electronics lab in a computer



Electronics Workbench®

The electronics lab in a computer

Professional Version

User Manual

MS-DOS

INTERACTIVE IMAGE TECHNOLOGIES LTD.
TORONTO, CANADA

DISCLAIMER AND ACKNOWLEDGEMENTS

95 94 93 92 91   10 9 8 7 6 5 4 3

Published January 1989. 4th Edition, published December, 1991

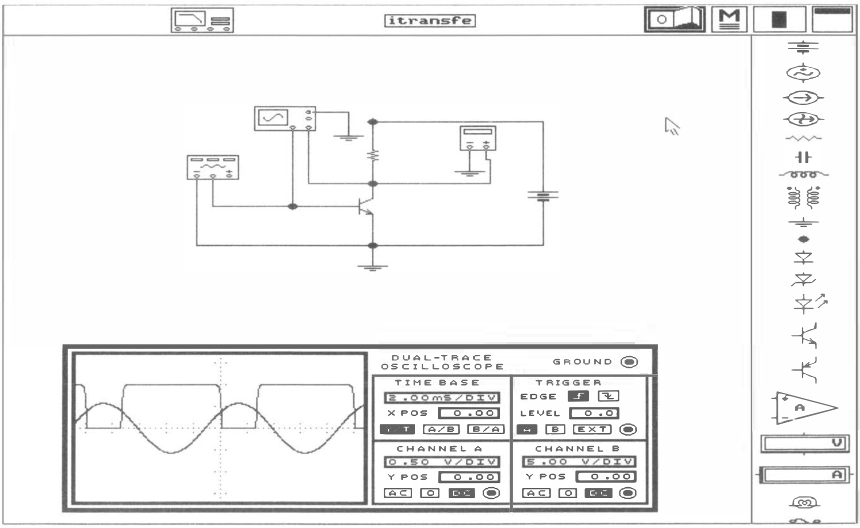
Printed in Canada.

Table of contents

Purpose

ELECTRONICS WORKBENCH lets you

- construct a schematic for an electronic circuit on a computer display,
- simulate the activity of that circuit,
- display its activity on test instruments contained within the program,
- and print a copy of the circuit, the instrument readings and parts list.



ELECTRONICS WORKBENCH has separate modules for analog and digital logic circuits. The size of circuit it can work with is limited only by available memory and the time a user will wait for results.

learning tool

ELECTRONICS WORKBENCH is a powerful learning tool. Many courses in high school, technical college and university deal with electricity, electronics and digital circuits. This program lets you experiment with digital and analog electronic circuits without using scarce and expensive laboratory facilities and materials.

It was designed with you in mind, to be so easy to use that it takes almost no time away from learning the subject. Within the first half hour of using the program, you can master the basic operations and make useful circuits.

If you have a personal computer for your own use, you can do exercises and assignments at your convenience. But if you are working in a computer facility, you still gain a lot of productivity.

learning goals

From Ohm's law onward, the program will let you experiment interactively with the fundamental theories and applications of electricity, electronic devices, and the principles of digital circuits. At the same time, ELECTRONICS WORKBENCH is powerful enough to let you do most assignments and exercises that may be found right up through a year or two of university courses in electrical engineering, computer science, physics, and the like.

If your purpose is to learn the theory of analog or digital electronic circuits, ELECTRONICS WORKBENCH provides an ideal environment for learning. It does not stop with theory alone, though. The test instruments in the program have the basic controls found on laboratory models and must be set up in the same way. You can use ELECTRONICS WORKBENCH to prepare for laboratory assignments and to rehearse for lab exams.

versatility

ELECTRONICS WORKBENCH is flexible enough to keep pace with most textbooks, laboratory exercise manuals, home study guides and courses, and many hobbyist books. It does not present lessons, but rather allows you to follow any course you wish. It contains comprehensive on-line help—including information on electronic parts and principles—that works with any other material.

It lets you print circuit schematics, the output from the test instruments, and parts lists in an attractive format. You can do exercises and experiments in a fraction of the time that breadboarding in the lab would take. Even if the assignment is to produce a breadboarded circuit, preparation with ELECTRONICS WORKBENCH will save time.

other uses

Teachers, lecturers and many others will find that using it to make demonstration materials is much faster also. Its ease of use and interactive design make ELECTRONICS WORKBENCH an excellent tool for lectures and presentations. Many devices are available to project the screen large enough for an audience to see. Even if all you need to do is prepare circuit diagrams of tested circuits for exercises, creating them with ELECTRONICS WORKBENCH saves time.

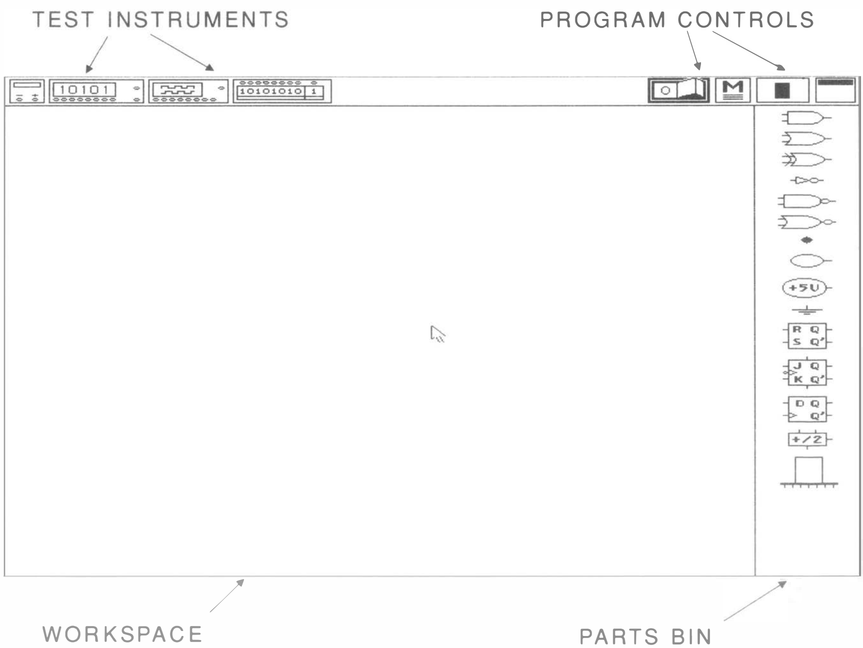
Appearance

This chapter presents a quick overview of the parts of the screen along with brief instructions to use them. If you are familiar with similar interactive systems and are willing to experiment, this may be all you need to get going with ELECTRONICS WORKBENCH.

All the actions referred to here are described in more detail in the following chapters.

the metaphor

The program models a workbench for electronics. The large central area on the screen acts as a breadboard for circuit assembly. On the right side of this workspace is a bin of parts. Above the workspace is a shelf of test instruments and program controls.



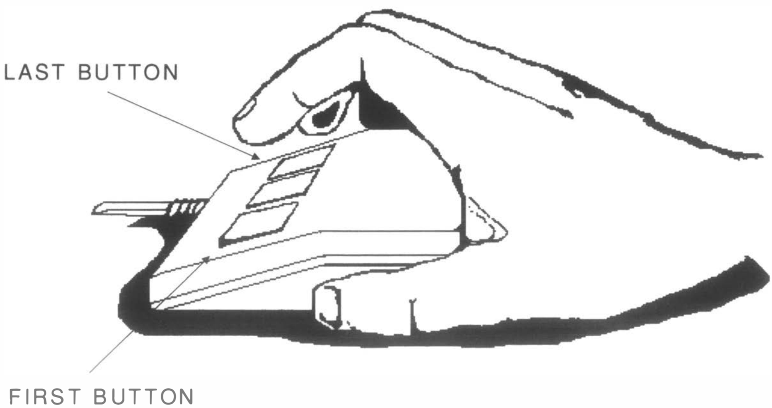
ease of use

A mouse is used to point to an object with the cursor. Clicking a mouse button causes an action to occur. All operations can be performed from pulldown menus or directly from graphic images on the screen, but function keys also perform most operations. Typing text from the keyboard is needed only to name new files, set values and labels for components and enter patterns into the word generator and truth table.

When the cursor appears as an hourglass, ELECTRONICS WORKBENCH is doing something that prevents user input. The program will accept input when the cursor returns to an arrow.

a note about the mouse

In the descriptions that follow, the left mouse button will be referred to as the first mouse button and the right will be called the last button. For right-handed users, this means the first button is controlled with the index finger and the last is controlled with the ring or middle finger.



This should be less confusing to most readers, since the mouse is used with one hand. It also allows left-handed persons to use a utility (if available, such as Logitech's CLICK program) to reverse the buttons and still follow the instructions.

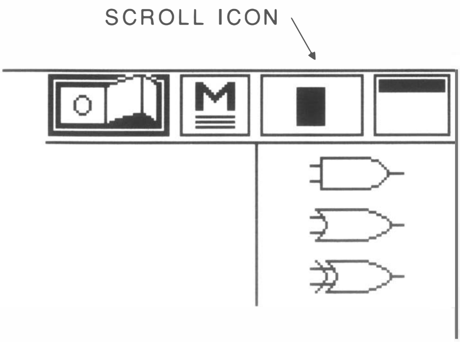
about help

Help is always available by pointing to an object and pressing F1. If no object is selected, a help window with a basic table of contents opens. You can then find the topic you wish.

the workspace

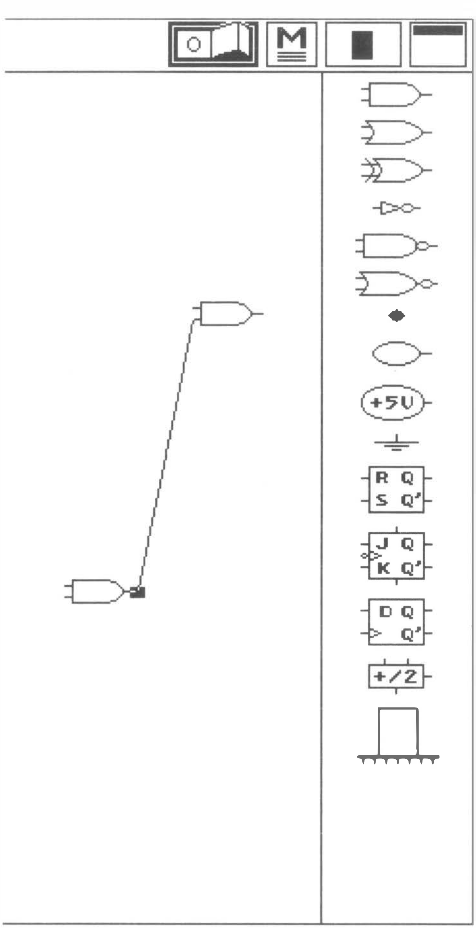
The workspace is used as a breadboard where you construct a circuit. After components are on the workspace, you can connect them with wires by pointing to their terminal connections and pulling a wire out to a terminal on another component.

The circuit may be considerably larger than the area of the screen. The workspace has an area of approximately three by three screens. A special scroll icon in the upper right allows you to move the circuit around to see different portions of it on the screen. The scroll icon shows the relative position of the visible portion in the workspace.



When you move a wire on the workspace beyond the edge of the screen, the workspace scrolls to keep up with it.

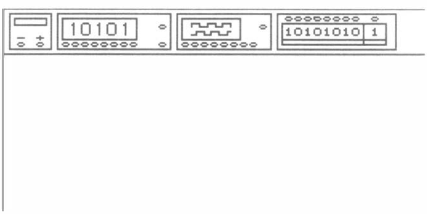
the parts bin



The parts bin contains the supply of components to build circuits. Pick components from the parts bin and place them on the workspace. To remove components from the workspace, drag them to the parts bin and release them, where they sort themselves into proper order. There is an unlimited number of each component.

When the tracker touches a component, the entire component highlights, showing it is ready to be moved or acted on. Drag components around by pressing and holding the first mouse button. The components have terminals for connecting wires. The terminals highlight when the tracker touches them. Press and hold the first mouse button after highlighting a terminal to pull out a wire. Connect the wire to a terminal on another component.

the instrument shelf



The left side of the "shelf" at the top of the screen holds test instruments. The instrument icons act just as components do, except that there is only one copy of each instrument available. Attach wires from the terminals of the instrument icons to test points in the circuit.

You must put connectors into the circuit wherever you wish to connect instruments to measure values.

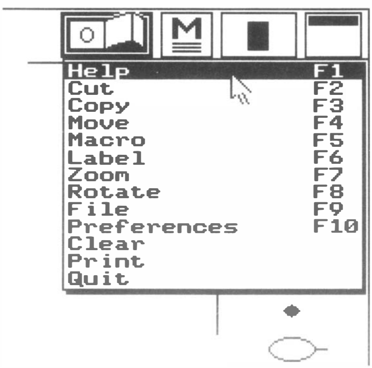
Instruments can be zoomed open by pointing to them and pressing F7 or double clicking. When they are zoomed open, you can move the faces of the instruments just as you do other objects (pressing and holding the first mouse button, then moving and releasing), adjust their controls and read their displays.

circuit name

The box near the middle of the top shelf contains the name of the file containing the circuit if it has been assigned. ELECTRONICS WORKBENCH does not require you to assign a circuit name before beginning work. You can type from the keyboard onto this text line after selecting it and clicking the first mouse button.

the control icons

Using the menu



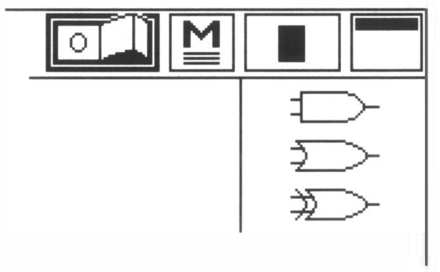
Point to the menu box with the mouse. Press and hold the first mouse button to pull down the menu. Still holding the first mouse button, move the selector bar to your choice and then release the first mouse button to make your choice. The menu will disappear without doing anything if you roll the tracker beyond the frame of the menu and release the mouse button.

Using the on/off switch

The on/off switch icon is used to start simulating the circuit. This is explained in detail below in the chapter on activating the circuit.

Using the scroll boxes

The two icons in the upper right are used to scroll the workspace and the parts bin. Move the tracker into the area, press and hold the first mouse button, and move the tracker to scroll the parts bin or the workspace.



Analog Tutorial

This tutorial will take you through the process of laying out a circuit and using the test instruments to see its activity. By the end, you will have used the basic features of ELECTRONICS WORKBENCH and will be ready to learn about its advanced features from the rest of this manual.

It does not matter which tutorial you begin with. Choose either the Analog Tutorial or the Digital Tutorial, whichever most interests you. This chapter contains the analog tutorial.

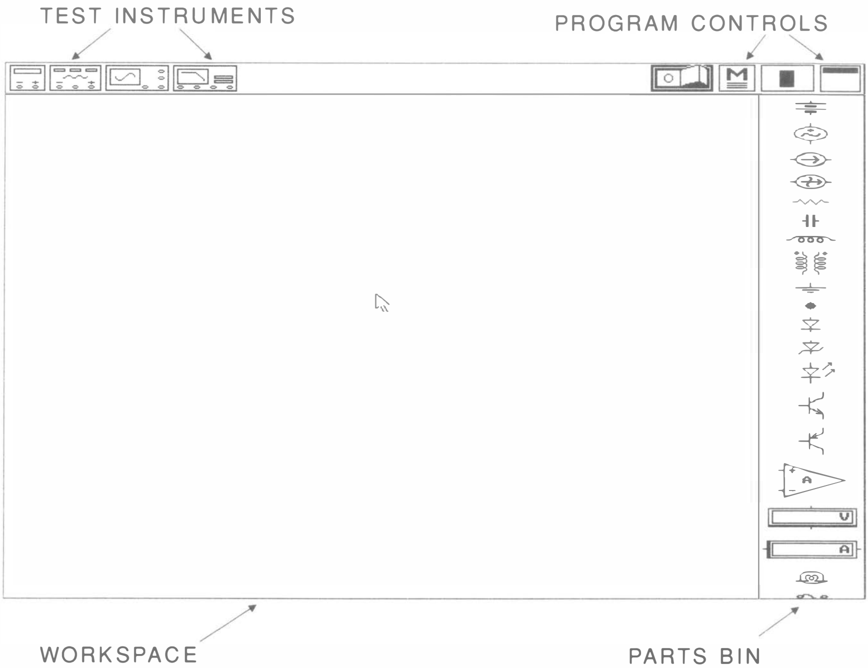
starting ELECTRONICS WORKBENCH

You must have MS-DOS running and your mouse driver installed. See the chapter on program setup and requirements for details about what ELECTRONICS WORKBENCH requires to operate and how to install it. ELECTRONICS WORKBENCH must be installed on your hard disk before you can run it.

For more information about the operation of DOS or of your mouse, consult the documentation that came with it.

To run the analog module, type “analog” at the DOS prompt and press Enter.

After a short pause, the title screen of ELECTRONICS WORKBENCH will appear and then be replaced by a screen similar to this:



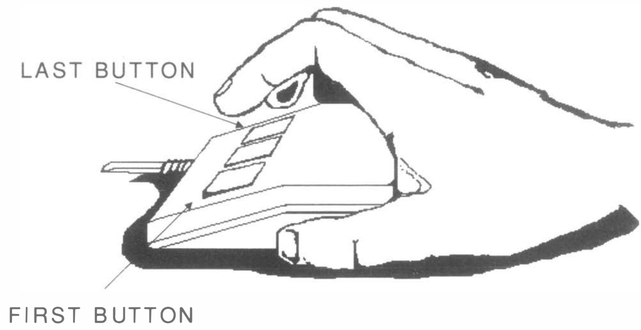
part 1: the basics of the mouse and the screen

The mouse

This section will introduce you to the use and feel of the mouse. If you are accustomed to using a mouse, you can skip forward.

If you are not familiar with using a mouse, take a moment to get comfortable with its operation.

We call the left button the "first mouse button" and the right one the "last mouse button." It is possible to use the software that comes with Logitech and Microsoft mice to reverse the buttons, if for instance a left-handed



person wishes. Check the documentation that comes with your mouse for details.

Mouse tracking

Use the mouse to move the arrow cursor around on the workspace. Point to the components in the parts bin. The components highlight when the tracker touches them. Move the mouse around until you get a feel for the movement to expect from the cursor.

Pick up components

This section shows you how to get components to build circuits. If you know how to press and drag objects, you may want to skip forward.

Point to the resistor in the parts bin. While it is highlighted, press and hold down the first mouse button. This picks up the component so you can drag it wherever you want. As long as you hold the first button, the component moves with the tracker.

Do not press the button before the component is highlighted.

Move your component to the center of the workspace. Release the mouse button and the component stays in place. You can move all components freely this way.

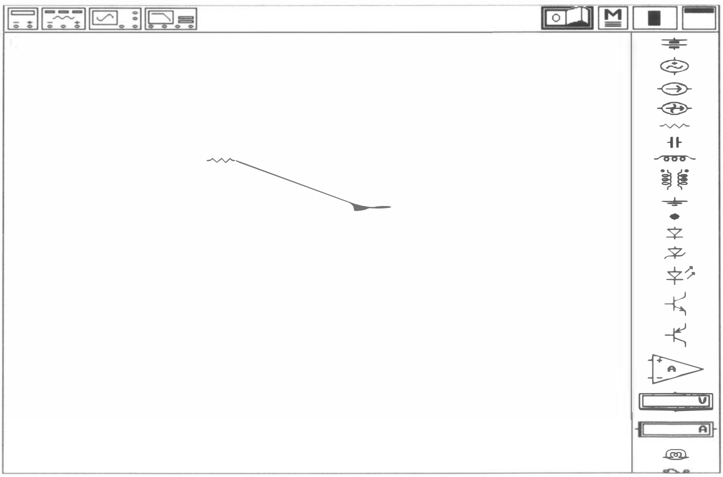
To remove a component from the workspace, pick it up and drag it to the parts bin. When you release components over the parts bin, they will sort themselves out neatly.

Practice moving components around until you are comfortable with the action.

Terminals and wires

This section introduces you to the terminals on the components and shows how to wire them together.

Components have terminals on them that connect wires. On most parts (except the connector) the terminals are short lines that stick out. On the resistor, for example, they are the two ends.



Put two resistors on the workspace.

Using the mouse, point to a terminal on one of the resistors — the terminal highlights with a small square. Press and hold the first mouse button and move the mouse to pull a wire out from the terminal. The wire follows the mouse tracker. This action is often called "rubber-banding" the line.

Release the mouse button and the wire disappears.

Pull a wire out and move the mouse until the end of the wire touches a terminal on the other component. The terminal should highlight. While the terminal is highlighted, release the mouse button and the wire will attach to the second terminal and find a route automatically.

You cannot connect a wire from one terminal to another on the same object.

Practice rubber-banding and connecting wires until you are confident about making connections.

The connector

This section explains the special features of the connector for joining wires.

The round dot in the parts bin is an important and special component. Use it to connect wires to each other and create test points in the circuit.

Point to the center of the connector to highlight the whole component and drag it.

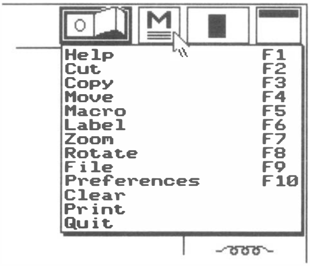
The connector can also be picked up "by the edge" by pointing to the region near it between two terminals. You may find it easier to handle this way.

The connector has four terminals, but the terminals do not stick out. When your tracker is pointing to a terminal, a square that is off-center highlights.

Practice moving the connector and attaching wires to it until you are comfortable with it.

The menu

This section explains how to use the menu to do things to objects in the program.



Before a menu choice can act on an object, you must select the object permanently with the last mouse button. Then choose an action from the menu. After the action is complete, deselect the object by pressing the last button without pointing to anything else.

If you are familiar with other menu systems, you might want to skip forward.

Before you use the menu in the next part of this tutorial, you should practice with it.

Move the tracker to highlight the icon on the top row with the letter “M” and press the first mouse button. While you hold the button down, a menu of operations is displayed.

Move the mouse to control the highlight bar in the menu. When you move the mouse off to the side, the tracker leaves the menu and appears as an arrow.

To make a choice from the menu, move the highlight bar to your selection and release the mouse button. To make no selection, just roll the tracker off the menu and release.

But, how does the program know what object to act on when you pick something from the menu?

Most operations on the menu will affect one or more objects in the program, but the objects must be selected first.

The simplest way to select something is to point to it and click the last mouse button. The object remains highlighted, showing that the next operation from the menu or function keys will act on it. You can select several objects this way, just by pointing and clicking the last button. To turn off the selections, click the last button without touching any object.

Put a component on the workspace. Point to it with the mouse and click the last mouse button. The component will remain highlighted after the tracker moves away.

Now point to the menu, press the first mouse button to pull it down, and move the highlight bar down to "Rotate." Release the first button. The component will rotate 90˚. Practice with other commands on the menu. You can cut, copy, move, etc.

Note that beside most menu operations, there is an F and a number. This F# reminds you of the function key that does the same thing. To use the function key shortcuts, just point to an object and press the function key.

part 2: build a simple circuit

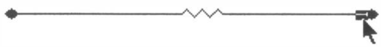
This section of the tutorial will lead you through the steps to build a simple analog circuit, a low-pass filter. Each operation will be introduced as you need it. The next part will step through simulating the circuit and using the test instruments.

Clear all other components off the workspace and put a resistor near the center top.

Pick up a connector from the parts bin and place it to the left of the resistor.

Point to the left terminal of the resistor, press and hold the first mouse button and pull out a wire.

Pull the end of the wire to the nearest terminal on the connector. When the terminal highlights, release the mouse button to connect the wire.



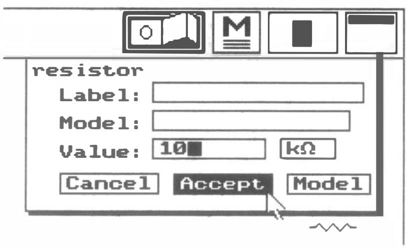
Get another connector from the parts bin and put it to the right of the resistor. Wire the resistor to the connector.

Now we have a transmission line with a load on it. But for it to work with the program, the resistance must have a value.

Point to the resistor and click the last mouse button to select it.

Now you will start to use the menu. If you are uncertain about what to do, go back and repeat the section of this tutorial about the menu.

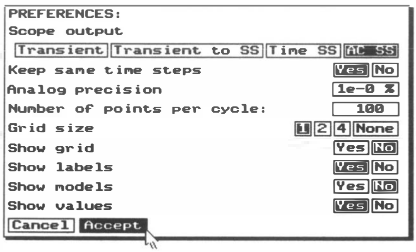
Point to the menu, press the first button to pull it down and move the highlight bar to "Label." Release the button. A window should open that allows you to give a value to the component or give it a text label.



Type "10" in the box labelled "Value;" then use the mouse to point to the small box with the ohm symbol in it. Press and hold the first mouse button and move the mouse to spin the selector to Kilohms. Release the mouse button.

Now move the tracker to "Accept" and click the first mouse button.

The resistor now has a value of 10 K ohms. But it would be nice to see the value somewhere.



Pull down the menu and select "Preferences." A large window with several buttons for choices will appear. For now, just click on "Yes" beside "Show values" and "Show labels" and then click on "Accept." When the window closes, the value of the resistor will be displayed beside it.

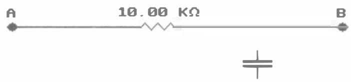
Now use the same steps to give a label to each connector. Point to the connector on the left and click the last mouse button to select it. Pull down the menu and choose "Label." When the label box is open, point to the top field and click the mouse button to put a text cursor in. Type in the letter "A" and then point to "Accept" and click.

Repeat this process to label the connector on the right "B."

When you finish, click the last mouse button while not pointing to any object in order to deselect the components you just labelled.

Now let's get on with the circuit. A low-pass filter is really a circuit that lets the high frequencies find an easy path to ground. A capacitor with the right value will pass high frequencies and block low.

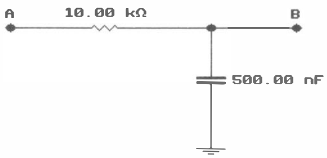
Pick a capacitor from the parts bin and put it on the workspace — below the resistor with its connectors and a little to the right.



Now rotate the capacitor — select it by clicking on it with the last mouse button, pull down the menu, and choose "Rotate." The capacitor is now vertical. Click the last mouse button again without touching the capacitor to deselect it.

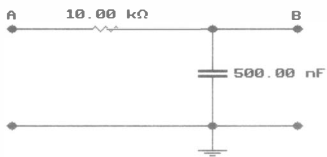
Pick a ground symbol from the parts bin and put it on the workspace below the capacitor. Wire the capacitor to ground.

Pick a connector from the parts bin. Place it on the wire from the resistor, just above the capacitor, and release it. It should snap into place in the wire. It is properly connected, just as if you had made connections from the other terminals to it. Now connect the bottom of the connector to the capacitor.



Using "Label" from the menu as you did above, give the capacitor a value of 500 nF.

To complete the layout of the circuit properly, use connectors to run another line, parallel to the first one, below the capacitor. Where the wires cross above the ground symbol, insert another connector so the whole side is grounded.



part 3: use the test instruments

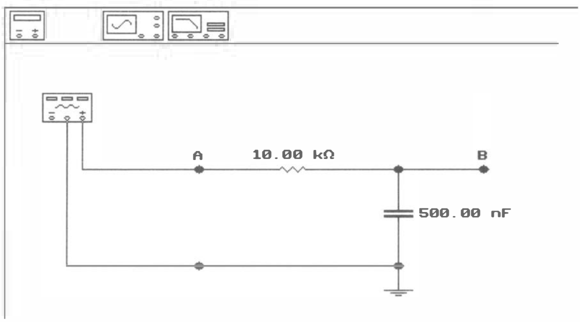
The icons for the test instruments are on the left of the shelf at the top of the screen. They highlight when the cursor touches them; you can pick them up and put them on the workspace just as you do components. There is only one copy of each instrument.

The function generator

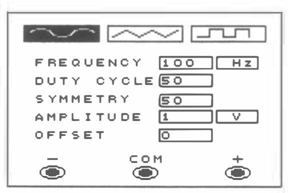
This unit of the tutorial introduces the operation of the function generator.

We will use the function generator to apply a signal to the circuit we have just built.

Pick the function generator icon from the equipment shelf and put it on the workspace near the connectors on the left. Attach wires from the "+" and "–" terminals of the function generator to the connectors on the left, "A" and the one below it.



Now point to the function generator icon and press F7 to zoom the face open. The instrument faces always go to the lower left corner when you first open them, but you can point to the face, press the first mouse button, and drag it anywhere you want.



On the function generator face are buttons and spin selectors.

The button with the sine wave on it should be lighted. If it is not, point to it with the mouse and click the first button.

Point to the spin selector beside “FREQUENCY” and press the first mouse button. Roll the mouse until the number 100 appears in the box. The next box should say “Hz.” Spin it to that value if necessary.

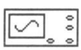
Point to the spin selector beside “AMPLITUDE” and set it to “1.” Make sure the units box beside it is set to “V.”

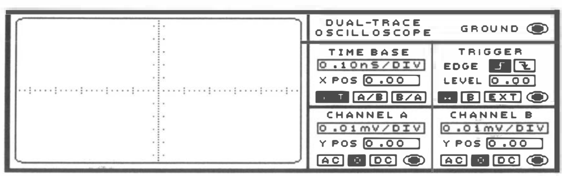
The other selectors, “DUTY CYCLE” and “SYMMETRY,” should be at 50% and “OFFSET” should be 0.

Now we have a signal source set up for the circuit. Next we have to measure the output.

The oscilloscope

This unit introduces you to the operation of the oscilloscope.

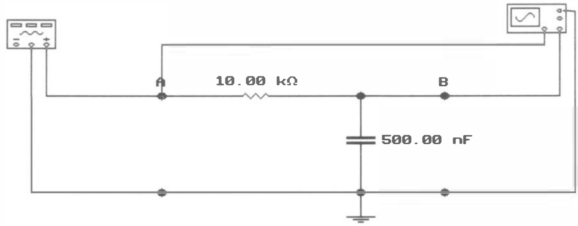




Bring down the oscilloscope icon and put it toward the right, above the circuit.

Using F7, zoom the face of the oscilloscope open. Move the scope face and the components of the circuit around so you can see the connections while the face is open.

The terminals on the oscilloscope icon correspond to the ones on the face. Run a wire from the terminal on



the icon for channel A to connector A in the circuit. Run another wire from channel B to connector B. Connect the ground terminal on the scope icon to the ground side of the circuit.

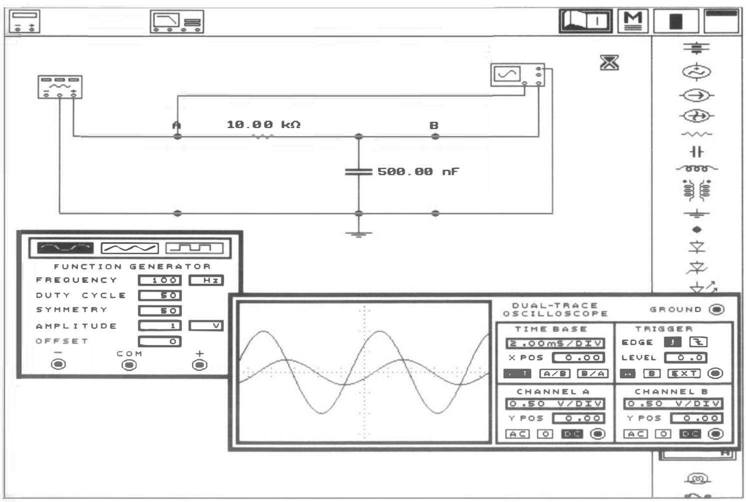
On the oscilloscope face, set the spin selector for the TIME BASE to 2.00 mS/DIV. Set both channel A and B to 0.50 V/DIV.

For both channels A and B, click on AC.

Simulating the circuit

This short unit summarizes the important facts about simulating the activity of the circuit.

Point to the switch on the top row and click the first mouse button. This starts the simulation of the circuit, like turning on the power. If everything is correct, the



oscilloscope should show two sine waves, one lower than the other and slightly out of phase.

When the simulation is complete, you can move the wires attached to the instruments around on the circuit and read values wherever you can make a connection. This is important, because sometimes the calculations to do the simulation can take a long time. (It may take minutes before you see any activity on the scope. It may take many more minutes before the simulation reaches a steady state condition.)

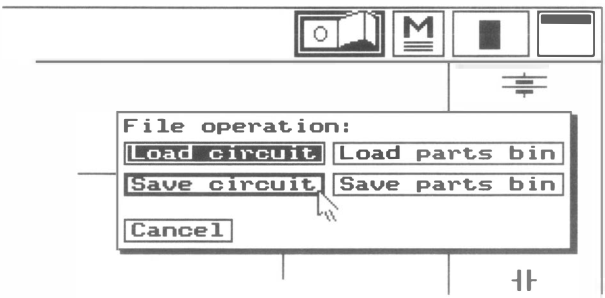
If you add anything to the circuit — even a connector — you must click on the switch again. If you change the values of any part of the circuit or change the input from the function generator, you must click on the switch.

part 4: saving work

This part of the tutorial explains how to save circuits.

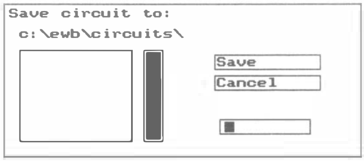
To save your circuit, pull down the menu and select "File." A dialog box will open.

The dialog box contains four buttons. Click on "Save circuit."



Another dialog box opens.

This box contains a list of files if any have been saved.



The current path to the directory where they will be saved is shown.

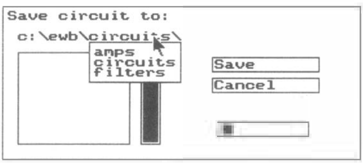
In the lower right corner is a text entry box. Type a name (up to eight characters) for the circuit.

The characters you type appear in the text field on the dialog box.

If you make a mistake or change your mind, just use the backspace or cursor keys to make corrections.

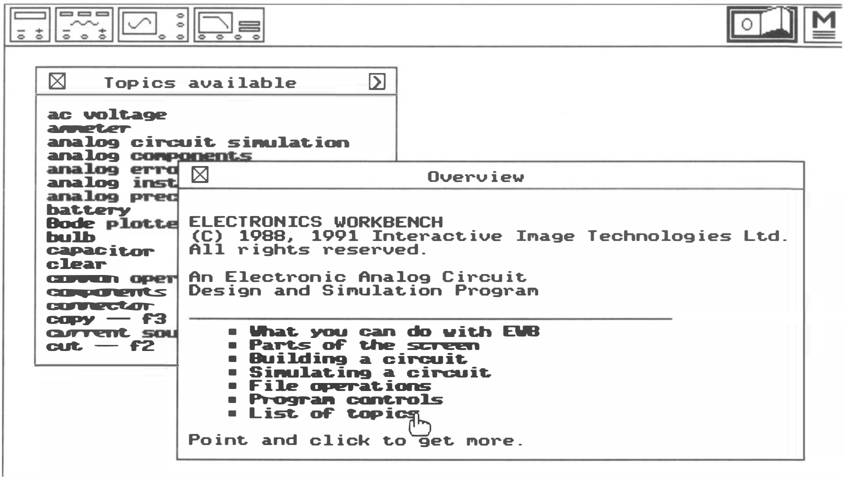
When you finish typing the name, click the first mouse button on “Save.” The filename then also appears in the box near the center of the top of the screen.

You can now retrieve this circuit in the future. Just choose “Load circuit” from the “File” dialog box.



The line above the list of files shows the drive and path to the current directory. If you want to change any part of it, point with the mouse and click the first button. This causes a sub-menu of all the available choices at that level to pull down. Roll the selector bar to your choice and release to change to the indicated drive or directory. The backslash at the end of the line may contain subdirectories. You can change drives this way as well. Any circuit files in the selected directory will appear in the file list box. This feature is especially useful on hard disk drives and networks.

While you are using ELECTRONICS WORKBENCH, help is always available. To learn what an object is, just point to it with the mouse and press F1. A window containing



help about the object will open. You can get more information about any word that is blue in the help window if you click on it. To get an overview of ELECTRONICS WORKBENCH and an index of all help topics, just press F1 while nothing is selected, or pick "Help" from the menu.

part 5: quitting

To quit ELECTRONICS WORKBENCH, choose "Quit" from the menu.

Now you have used all the basic operations you need to make use of ELECTRONICS WORKBENCH. Much more power is available to you. Read through the chapter on menu operations.

If you need to know more about how to use the instruments and what the controls are for, look in the chapters of this manual for the analog or digital modules.

Digital Tutorial

This tutorial will take you through the process of laying out a circuit and using the test instruments to see its activity. By the end, you will have used the basic features of ELECTRONICS WORKBENCH and will be ready to learn about its advanced features from the rest of this manual.

It does not matter which tutorial you begin with. Choose either the Analog Tutorial or the Digital Tutorial, whichever most interests you. This chapter contains the digital tutorial.

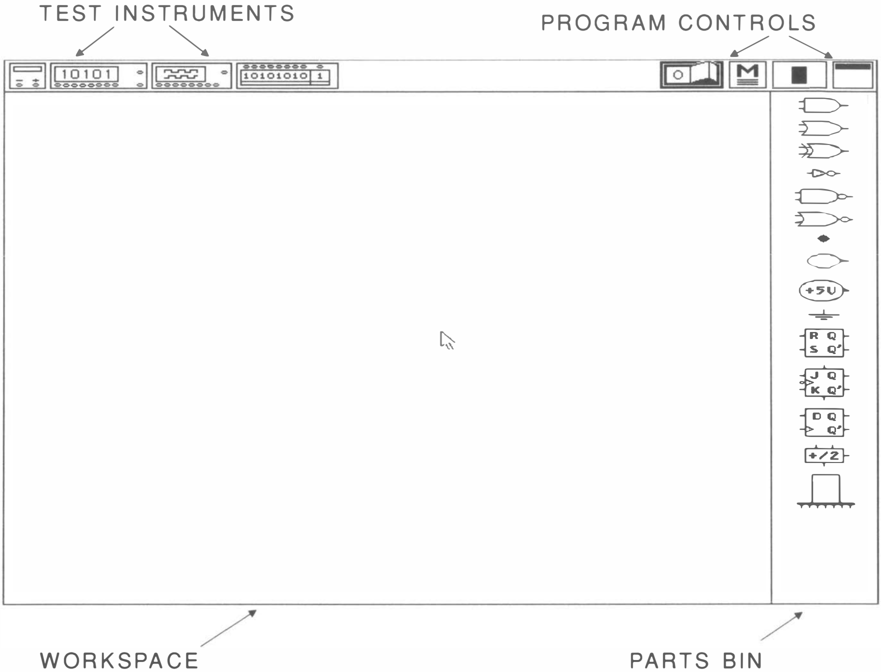
starting ELECTRONICS WORKBENCH

You must have MS-DOS running and your mouse driver installed. See the chapter on program setup and requirements for details about what ELECTRONICS WORKBENCH requires to operate and how to install it. ELECTRONICS WORKBENCH must be installed on your hard disk before you can run it.

For more information about the operation of DOS or of your mouse, consult the documentation that came with it.

To run the digital module, type “digital” at the DOS prompt and press Enter.

After a short pause, the title screen of ELECTRONICS WORKBENCH will appear and then be replaced by a screen similar to this:



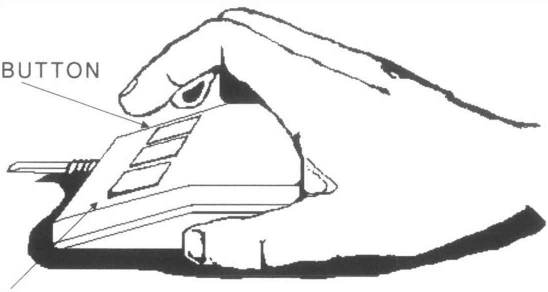
part 1: the basics of the mouse and the screen

The mouse

This section will introduce you to the use and feel of the mouse. If you are accustomed to using a mouse, you can skip forward.

If you are not familiar with using a mouse, take a moment to get comfortable with its operation.

We call the left button the “first mouse button” and the right one the “last mouse button.” It is possible to use the software that comes with Logitech and Microsoft mice to reverse the buttons, if for instance a left-handed



person wishes. Check the documentation that comes with your mouse for details.

Mouse tracking

Use the mouse to move the arrow cursor around on the workspace. Point to the components in the parts bin. The components highlight when the tracker touches them. Move the mouse around until you get a feel for the movement to expect from the cursor.

Pick up components

This section shows you how to get components to build circuits. If you know how to press and drag objects, you may want to skip forward.

Point to the AND gate in the parts bin. While it is highlighted, press and hold down the first mouse button. This picks up the component so you can drag it wherever you want. As long as you hold the first button, the component moves with the tracker.

Do not press the button before the component is highlighted.

Move your component to the center of the workspace. Release the mouse button and the component stays in place. You can move all components freely this way.

To remove a component from the workspace, pick it up and drag it to the parts bin. When you release components over the bin, they will sort themselves out neatly.

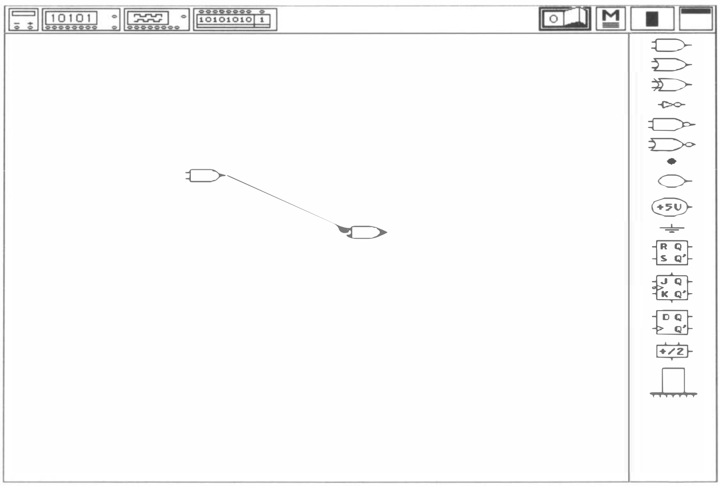
Practice moving components around until you are comfortable with the action.

Terminals and wires

This section introduces you to the terminals on the components and shows how to wire them together.

Components have terminals on them that connect wires. On most components (except the connector) the terminals are short lines that stick out. On the inverter (NOT gate), for example, they are the two ends.

Put two AND gates on the workspace.



Using the mouse, point to a terminal on one of the AND gates — the terminal highlights with a small square. Press and hold the first mouse button and move the mouse to pull a wire out from the terminal. The wire follows the mouse tracker. This action is often called "rubber-banding" the line.

Release the mouse button and the wire disappears.

Pull a wire out and move the mouse until the end of the wire touches a terminal on the other component. The terminal should highlight. While the terminal is highlighted, release the mouse button and the wire will attach to the second terminal and find a route automatically.

You cannot connect a wire from one terminal to another on the same object.

Practice rubber-banding and connecting wires until you are confident about making connections.

The connector

This section explains the special features of the connector for joining wires.

The round dot in the parts bin is an important and special component. Use it to connect wires to each other and create test points in the circuit.

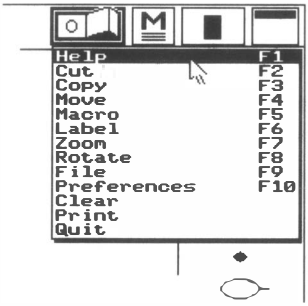
Point to the center of the connector to highlight the whole component and drag it.

The connector can also be picked up "by the edge" by pointing to the region near it between two terminals. You may find it easier to handle this way.

The connector has four terminals, but the terminals do not stick out. When your tracker is pointing to a terminal, a square that is off-center highlights.

Practice moving the connector and attaching wires to it until you are comfortable with it.

This section explains how to use the menu to do things to objects in the program.



Before a menu choice can act on an object, you must select the object permanently with the last mouse button. Then choose an action from the menu. After the action is complete, deselect the component by pressing the last button without pointing to anything else.

If you are familiar with other menu systems, you might want to skip forward.

Before you use the menu in the next part of this tutorial, you should practice with it.

Move the tracker to highlight the icon on the top row with the letter “M” and press the first mouse button. While you hold the button down, a menu of operations is displayed.

Move the mouse to control the highlight bar in the menu. When you move the mouse off to the side, the tracker leaves the menu and appears as an arrow.

To make a choice from the menu, move the highlight bar to your selection and release the mouse button. To make no selection, just roll the tracker off the menu and release.

But, how does the program know what object to act on when you pick something from the menu?

Most operations on the menu will affect one or more objects in the program, but the objects must be selected first.

The simplest way to select something is to point to it and click the last mouse button. The object remains highlighted, showing that the next operation from the menu or function keys will act on it. You can select several objects this way, just by pointing and clicking the last button. To turn off the selections, click the last button without touching any object.

Put a component on the workspace. Point to it with the mouse and click the last mouse button. The component will remain highlighted after the tracker moves away.

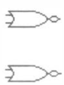
Now point to the menu, press the first mouse button to pull it down, and move the highlight bar down to "Rotate." Release the first button. The component will rotate 90°. Practice with other commands on the menu. You can cut, copy, move, etc.

Note that beside most menu operations, there is an F and a number. This F# reminds you of the function key that does the same thing. To use the function key shortcuts, just point to an object and press the function key.

part 2: build a simple circuit

This section of the tutorial will lead you through the steps to build a simple digital circuit, a flip-flop. Each operation will be introduced as you need it. The next part will step through simulating the circuit and using the test instruments.

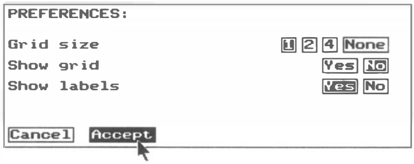
Clear the workspace if any components are on it.



- Pick a NOR gate from the parts bin and put it on the workspace. Place a second NOR gate below it.

Pick a connector from the parts bin and place it to the right of the first NOR gate.

Place another connector to the right of the second NOR gate, below the first connector.

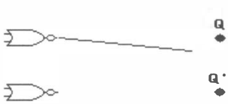


It would be nice to label parts of the circuit. To be able to see the labels, you must pull down the menu and choose “Preferences.” When the dialog box opens, click on “Yes” beside “Show labels” and then click on “Accept.”

Point to the upper connector so it is highlighted, click the last mouse button to select it, and pull down the menu and choose “Label.” Type the letter “Q” into the label field of the dialog box and then click on “Accept.”

Label the lower connector “Q′” (for NOT Q).

Connect the output of the top NOR gate to the Q connector with a wire —



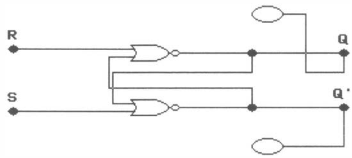
- Point to the end of the NOR gate so it highlights as a small box.
- Press the first mouse button and roll to pull a wire out.
- Still holding the first button, move the tracker until the terminal on the connector highlights.
- Now release the button and the wire will find a route between them.

Connect the lower NOR gate to Q′ the same way.

Put a connector between the output of the NOR gate and Q by picking it from the parts bin and releasing it on the wire.

Insert another connector between the lower gate and Q′.

Draw a wire from the upper connector to the input of the lower NOR gate and from the lower connector to the input of the upper NOR gate.



Add connectors and connect wires to the remaining inputs of the NOR gates as shown. Label the connectors on the left “R” and “S.”

Place two LED probes as shown and attach them to Q and Q′.

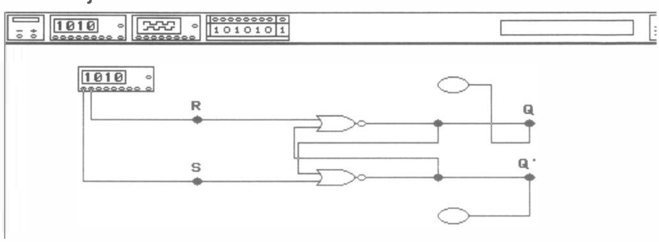
part 3: use the test instruments

The icons for the test instruments are on the left of the shelf at the top of the screen. They highlight when the cursor touches them; you can pick them up and put them on the workspace just as you do components. There is only one copy of each instrument.

The word generator

This unit introduces the use of the word generator.

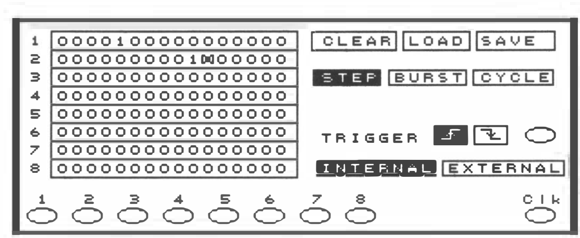
We will use the word generator to supply input to the circuit just built.



Pick the word generator icon from the equipment shelf and put it on the workspace near the connectors at the left of the circuit. Attach wires from the first two terminals of the word generator to connectors R and S.

Now point to the word generator icon and press F7 to zoom the face open. The instrument faces always go to the lower left corner when you first open them, but you can point to the face, press the first mouse button, and drag it anywhere you want.

On the word generator face are several buttons and a large field where you enter ones or zeros to form digital words.



Point to the top row of zeros and click the first mouse button. A box cursor will appear in the input field where you pointed. Press the right arrow key to move the cursor a few spaces in from the left. Type a “1.”

Press the down arrow key to move the cursor down a line. Move it a few more spaces to the right and type another “1.”

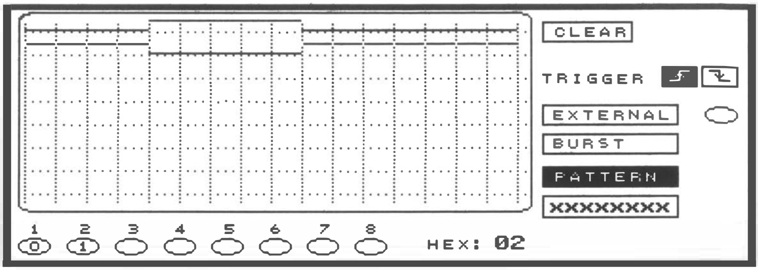
(You should have the first two terminals on the word generator icon connected to the input, R and S, of the circuit.)

Now we have the input to the circuit ready. Next we must set up the logic analyzer to see the output.

The logic analyzer

This unit of the tutorial demonstrates the logic analyzer.

Pick the logic analyzer icon from the equipment shelf and put it on the workspace near the connectors at the right of the circuit. Attach wires from the first two terminals of the logic analyzer to connectors Q and Q′ on the right.



Now point to the logic analyzer icon on the workspace and press F7 to zoom the face open. Move the faces of both instruments around so you can see the displays of both faces while they are open.

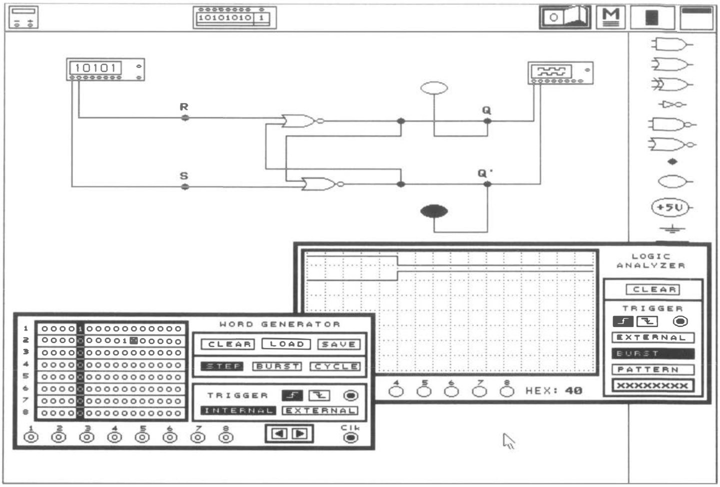
On the logic analyzer face are several buttons and a large field where the levels of the circuit are displayed.

Simulating the circuit

This short unit introduces digital simulation.

To simulate the operation of the circuit, point to BURST on the word generator and click the first mouse button. This sends all sixteen words from the word generator to the circuit. The output of the circuit will be displayed on the logic analyzer.

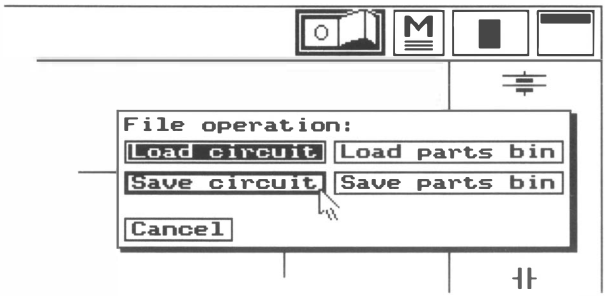
Much of the simulation you will want to do with digital circuits can be controlled from the face of the word generator directly. See the chapter on activating the circuit for more details on the difference between analog and digital simulation procedures.



If you click on the switch on the top row while the word generator is in the circuit, the current selection on it will run.

part 4: saving work

This part of the tutorial explains how to save circuits.



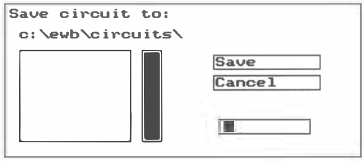
To save your circuit, pull down the menu and select "File." A dialog box will open.

The dialog box contains four buttons. Click on "Save circuit."

Another dialog box opens.

This box contains a list of files if any have been saved.

The current path to the directory where they will be saved is shown.



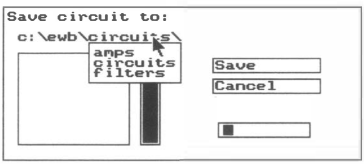
In the lower right corner is a text entry box. Type a name (up to eight characters) for the circuit.

The characters you type appear in the text field on the dialog box.

If you make a mistake or change your mind, just use the backspace or cursor keys to make corrections.

When you finish typing the name, click the first mouse button on "Save." The filename then also appears in the box near the center of the top of the screen.

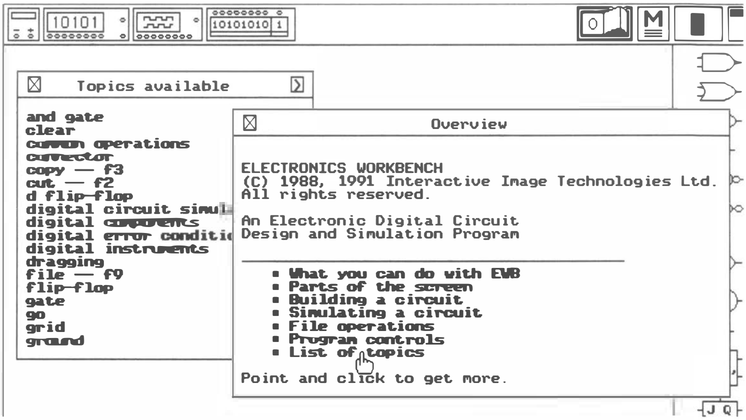
You can now retrieve this circuit in the future. Just choose "Load circuit" from the "File" dialog box.



The line above the list of files shows the drive and path to the current directory. If you want to change any part of it, point with the mouse and click the first button. This causes a sub-menu of all the available choices at that level to pull down. Roll the selector bar to your choice and release to change to the indicated drive or directory. The backslash at the end of the line may contain subdirectories. You can change drives this way as well. Any circuit files in the selected directory will appear in the file list box. This feature is especially useful on hard disk drives and networks.

Help

While you are using ELECTRONICS WORKBENCH, help is always available. To learn what an object is, just point



to it with the mouse and press F1. A window containing help about the object will open. You can get more information about any word that is blue in the help window if you click on it. To get an overview of ELECTRONICS WORKBENCH and an index of all help topics, just press F1 while nothing is selected, or pick "Help" from the menu.

part 5: quitting

To quit ELECTRONICS WORKBENCH, choose "Quit" from the menu.

Now you have used all the basic operations you need to make use of ELECTRONICS WORKBENCH. Much more power is available to you. Read through the chapter on menu operations.

If you need to know more about how to use the instruments and what the controls are for, look in the chapters of this manual for the analog or digital modules.

Common operations

Lay out a circuit by picking components from the parts bin and placing them on the workspace.

The procedure is simple:

- To do anything with an object in ELECTRONICS WORKBENCH, point to it with the mouse.
- When an object is touched by the cursor, it lights up, showing that it is ready for action.
- A lighted object will do the simplest thing it can when the first mouse button is clicked. For most objects, this means it will move with the tracker.
- Function keys do other things to a highlighted object. The details of other operations are given below in the chapter on Menu Operations.

Moving the tracker onto a component causes it to highlight. Pressing and holding the first mouse button makes the component move with the tracker until the button is released, leaving the component in place on the workspace. Components are connected with wires by pointing to a terminal, pressing the first mouse button and stretching the wire to another terminal to highlight it, and then releasing the button.

In summary:

To pick an object up, point to it with the tracker and then press and hold the first mouse button. Do not press the mouse button before the object highlights. To release the object, release the button.

selecting objects

Throughout these instructions, we refer to actions on selected objects. To keep the explanations simple, we use “select” to refer to any of the following actions:

One at a time

When the cursor moves into an active area or near an active object, the area or object highlights to indicate that it is selected and ready for action. Clicking the first mouse button causes the appropriate action to happen, such as picking up a component, stretching a wire, or moving a spin selector. Other keys may be pressed to act on an object that is selected by pointing. These other actions are explained below in the chapter on Menu Operations (p. 6-1).

Permanently

Use the last mouse button to select a component so it stays selected while you do something else. You deselect it by clicking the button while not touching any component. You can select several components at one time for mass cutting, moving, or copying. You must do this to objects to use Menu selections.

In a block

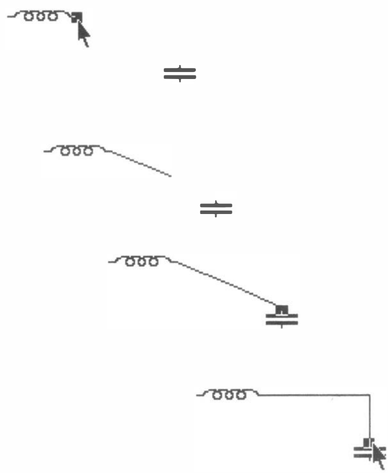
You can also select components in groups — holding the last mouse button while moving the tracker causes a rectangle to expand, selecting components in its range. When the button is released, the components within the rectangle will be selected.

dragging objects around

Point to a component or instrument with the mouse, press and hold the first button to move the object around. Release the button to leave the object in its new place. Objects released on top of other objects will return to their original places.

wiring components together

Connect components by selecting a terminal and stretching a wire to its destination.



To connect components or instruments with wires, point to a terminal on the object. When the terminal is highlighted, press and hold the first mouse button while moving the tracker. A "rubber band" line appears, with one end attached to the terminal and the other attached to the tracker. Move the tracker to highlight a terminal on another component and release the button.

The wire finds a route between the terminals automatically. The program draws a wire consisting of horizontal and vertical line segments between the two terminals, making sure the wire does not go on top of other components and does not overlap other wires. Wires cross at right angles without connecting. To make a connection between two wires, use the connector from the parts bin.

You can insert components into existing wires by aligning their terminals with the wire and releasing the mouse button. There must be room for the component to fit or it will bounce away.

A simple junction is included in the parts bin to allow direct connection between wires or attachment of test instruments. Like all other circuit elements, these connectors must be in a circuit before it is simulated with the on/off switch in order to have values. Put connections in your circuit for test points while you build it.

If a component already wired in place in the circuit must be rotated or moved, the wires automatically connect to the new position. Moving a component, even accidentally, does not undo prior work.

Disconnect wires by selecting a terminal, holding the first mouse button, and releasing away from any terminal.

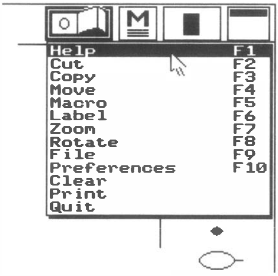
Setting wire colors

To make your circuit easier to read, you can set the color of the wires. Point to a terminal on a component and click the last mouse button while holding the ALT key. A small menu of colors will appear at the cursor. Move the cursor to the color you wish and release the mouse button. All wires drawn thereafter will be in the selected color. You can change the color of wires already drawn this way too. If a colored wire is attached to Channel A or B on the oscilloscope, the trace for that channel will be the same color as the wire.

In summary:

Put components on the workspace and wire them together to form a circuit. You can select components for operations chosen from the menu or by pressing function keys.

Menu operations



Use the Menu by selecting the "M" icon on the top shelf and holding the first mouse button. While the button is held, the menu stays down and the tracker controls the highlight bar while it is on the menu box. Move the highlight bar to the desired selection and release the button to make the action happen.

To make the menu disappear without doing anything, move the tracker outside the menu frame and release the button.

Most actions require you to select an object on the workspace beforehand. If nothing happens when you release the button, this may be the problem.

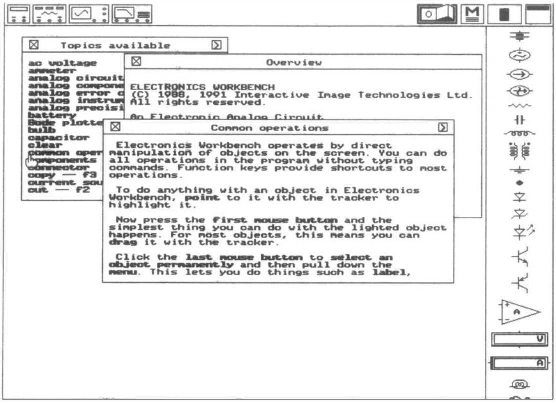
Most of the operations on the menu also have function key shortcuts. You can point to an object and simply press the function key.

For more general background on using the program, see the description of selecting objects and using the menu in the chapter on common operations.

Help — F1

Choose "Help" from the menu or press F1 to activate the help system. If you select an object before calling help, the help window will refer to the selected object. If you call help with no object selected, an overview of topics will appear. Click on any word that appears blue for more information. You can call an index of all help topics from the home screen.

Within the help system, cross-references to other topics



are marked in blue if you are using the default palette. You can select these words to open another window on the new topic. More than one help window can be open at once. You can move the windows around on the workspace and the workspace is usable while the windows are open.

Cut — F2

You can cut permanently selected components from the workspace with this menu selection or F2. This is useful for removing several components at once from the circuit. You can select the components one at a time by clicking the last mouse button on them, or you can draw a rectangle over a group by pressing the last mouse button and dragging to select a block. You can drag individual components back to the parts bin and release them; they need not be disconnected beforehand.

Copy — F3

To copy one or more components on the workspace to another place, select them permanently and use the menu or point with the mouse and press F3.

A quick way to get lots of the same component on the workspace is to select a component permanently and press F3 repeatedly while moving the mouse.

Copies of components have the same labels and values as the original, so you can make many copies with the same value easily by assigning a value before copying.

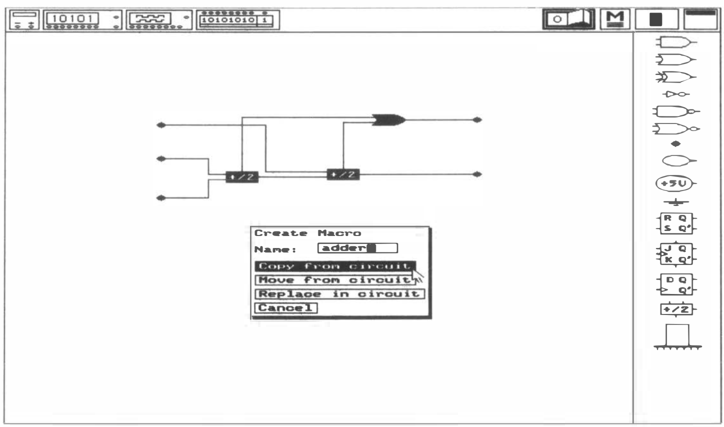
You can select and copy groups of objects together with their connections to other parts of the screen.

Move — F4

You can move a group of selected components with the menu selection or F4. If there is no room to place them on the workspace when released, they will bounce back to their source.

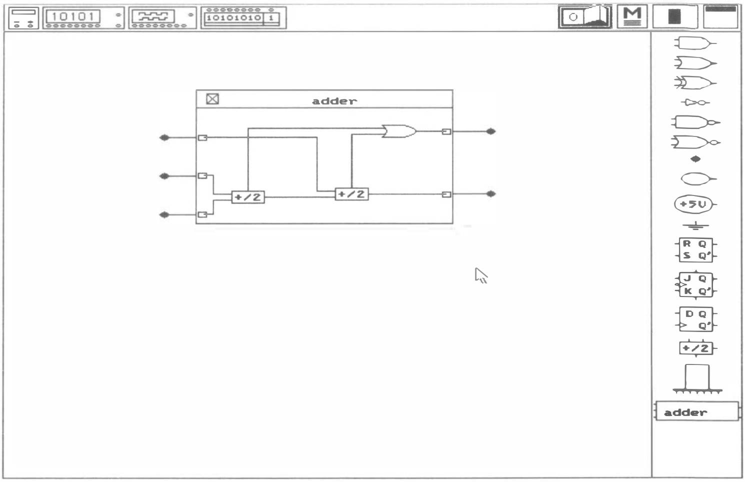
Macro — F5

You can combine circuits into a macro block, creating in effect your own integrated circuit.



Select the desired components on the workspace and choose "Macro" from the menu. A dialog box will prompt you for more action. You can leave the original components on the workspace by choosing to copy the components, or remove them by choosing to move the components. You should also give the macro a name, which appears on the component.

The macro is then automatically placed into the parts bin in a standard package. You can edit the contents of the macro by zooming it open on the workspace.



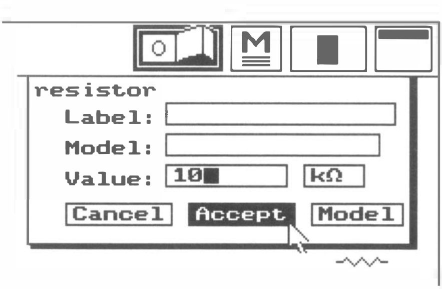
Terminals appear on the macro package where wires leave the box. You can move the terminals when the macro is zoomed open or you can create terminals by stretching wires to the sides of the macro box from components inside. The icon reflects the structure of the full-sized view.

If you want to use macros in a future session, you must save the parts bin containing them and reload it when you use the circuit again.

Label — F6

Labels and values can be given to components in the circuit or in the bin. Analog components must be given values for the circuit simulation to work.

Default values are not provided in the analog parts bin for passive components. The active analog components have a generally useful default model. You can change various parameters of the active components to create any model desired. If you are not familiar with the parameters, you can find a brief explanation of them along with their use in the component model in the chapter that presents the technical exposition.



To give values to components, select them and choose "Label" from the menu or press F6. A dialog box opens that contains text fields. One is for the label you wish to give and another is for the value of an analog component. To enter a name or value, put the text cursor in the field by pointing with the mouse and clicking. The text cursor remains active in the selected field even while the mouse tracker is outside. To remove the text cursor from the field, click the first mouse button outside.

Analog components can be given model names, all having the same value. If you name a resistor R1, for example, all resistors labeled with the model name will have the same value. If you change the value of one of them, all will change.

You can label the entire circuit by choosing "Label" when no other object is selected. The Label box for the circuit contains six lines of twenty-five characters, which may be printed out with the schematic.

Values may be given to the components in the parts bin, so that many components with the same value can be used. Changing the value of the component in the bin does not change values already in the circuit.

You can select a block of components and then label each in turn by choosing "Label," assigning values or labels to each component, and cancelling the operation when desired or finished.

The Label function also lets you control the characteristics of the multimeter. You can set the impedence to make it match actual equipment.

Zoom — F7

"Zoom" operates to show enlarged views of the instrument faces or the contents of macros. You can also zoom these objects open or closed by pointing and clicking the mouse button twice in quick succession (double clicking) or by pressing F7.

Rotate — F8

You can rotate most of the components to achieve nearly any desired layout on the workspace. Each time you select a component and rotate it, it turns 90 degrees clockwise.

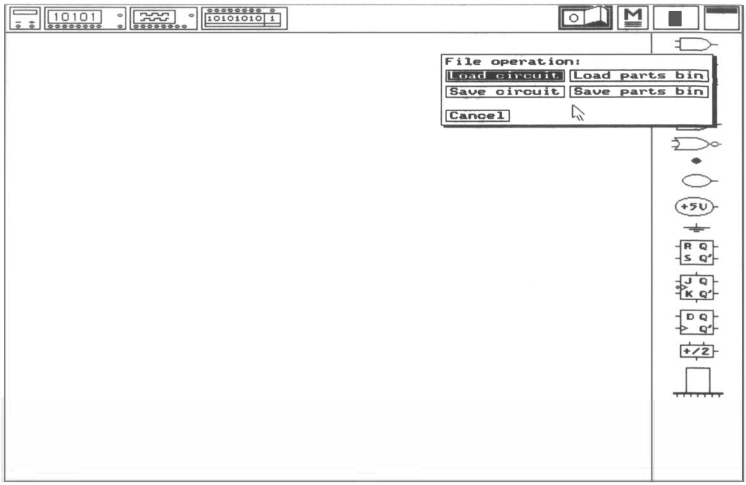
The ground symbol does not rotate.

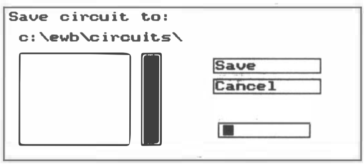
The transistors rotate 90 degrees the first time "Rotate" is selected, then reverse their symmetry for the remaining two positions. This enables you to follow standard drafting conventions when laying out a circuit.

The ammeter, voltmeter and half adder do not rotate as a whole, but their terminals do.

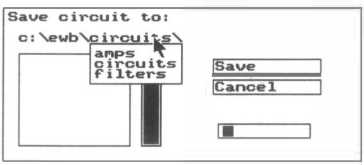
File — F9

To save or load a circuit or a set of components, select "File" or press F9. A dialog box appears where you choose to load or save circuits or parts bins.





After you choose from the dialog box, a file selector box will appear. It contains a list of filenames and buttons to act or cancel. The lower box is a text field. Enter a new name in it before saving.



Across the top of the file selector box, the current path and directory are listed. You can point to any part of the pathname and press the first mouse button to get a selector box of all other possibilities. Roll the mouse selector down the menu list and release on a directory name to change to it.

Circuits

A file selector box with a list of existing circuits will appear. To save a new circuit, type a name (up to eight characters) into the lower empty box. To load a circuit file, select a name from the scrolling list or type the name in the lower box.

Components

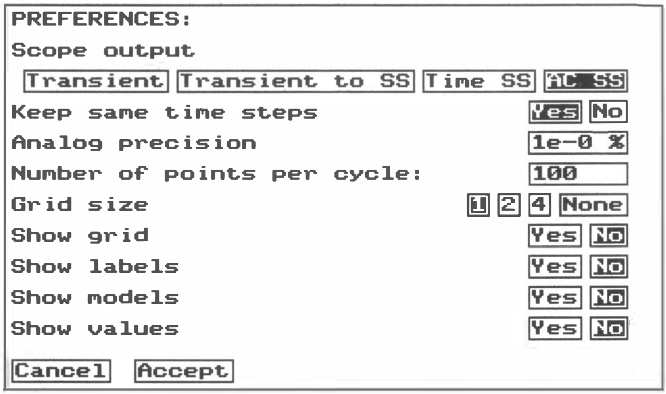
A file selector box with a list of existing parts bins will appear. Saving special parts files is useful to collect special macro circuit components or to control teaching assignments.

You can label components in the parts bin, so that many components of the same value can be put into the circuit easily. The value in the parts bin can be changed without affecting components already on the workspace. You can cut components from the parts bin by pointing to them with the tracker and pressing F2.

WARNING

If you cut components from the parts bin and wish to save it, be sure to change the name of the special set. If you save a special set with the same name as the default set, you can lose components and not get them back. Always keep a backup copy of the original parts files. The default set of analog components is in the file EWB.LA. The default set of digital components is in the file EWB.LD.

Preferences — F10



Scope output

Appearing for analog simulation only, this controls the display of the waveform on the oscilloscope.

When “AC SS” is selected (the default, standing for AC steady-state analysis), a one-point solution is calculated. If this solution exceeds the limits set by certain components in the circuit, or if “Time SS” is selected, a time domain analysis will be performed, using the single-point solution as a starting point. The analysis of the circuit continues until a steady state is achieved. The scope redraws each time the screen is full. Some circuits may take many cycles to reach stability.

When “Transient” is selected, the time-domain analysis, taking all circuit values set to zero as the starting point, is displayed for a single cycle and the trace stops. When “Transient to SS” is selected, the initial response is calculated in this manner and displayed, and then the analysis continues as in “Time SS” until steady state is achieved.

Keep same time steps

When ELECTRONICS WORKBENCH calculates a point in the circuit, it automatically reduces the time step used to solve the equations representing the circuit until it reaches a solution. It then uses the same time step for subsequent cycles, thus converging to a solution more rapidly. However, with some circuits this time step may be too small for subsequent cycles and you will get an error message to check your circuit configuration. In such cases, choose "No" to tell the program to start each cycle with the original default time step. This will be slower, but will usually give correct results.

Analog precision

You can control the degree of accuracy used to compute the values of the analog simulation. Requiring less accuracy can speed up circuit simulation. The smaller the percentage of precision, the more accuracy will be computed. The program defaults to the least precision and greatest speed.

Number of points per cycle

Changing this spin selector to a larger number will provide more detail in the scope trace. This will naturally make the analysis slower.

Grid size

A gridsnap feature causes components to align themselves automatically to preset lines in the workspace. This helps you get wires straight without small kinks in them. Three sizes of grid are available: of one unit, two units or four units. (A unit is the smallest possible distance between terminals on the components or instruments.) Only at the smallest grid size will all terminals on all objects align to grid points. By default, the grid is active at the smallest size when the program begins. You can turn it off if you wish.

Show grid

When the gridsnap is active, you can display the grid if you wish. If you do not choose to show the grid, it will still operate as long as a size is selected.

Show labels

You can select this to display labels that you have given to components. When both “Show labels” and “Show values” are selected, the label takes precedence over the value. If a model is assigned to a component, the label takes precedence over the model, but the model takes precedence over the value. If only labels are displayed, the label shows if assigned.

Show models

Select this to display the model name of components assigned models. Labels assigned to components will take precedence over model names if “Show labels” is also selected. If “Show values” is also selected, the model name will take precedence.

Show values

You can select this to display values that you have given to components. As explained above, if more than one of labels, models and values are displayed, labels take precedence over models and models take precedence over values.

Clear

Select this to clear the workspace, the circuit name, and the readings on instruments. A dialog box will ask confirmation of the action.

Print

Select "Print" from the menu to get the circuit printed on a dot-matrix or laser printer. A menu of options will appear to print the circuit, the faces of the test instruments, and a component list.

You can create a label for the schematic by labelling the circuit. If you choose "Label" from the Menu with no components selected, a window with a text field of six lines by twenty-five characters opens. Type into each line of this box.

To change the printer configuration, choose "Print" from the menu. When the print selection box is open, you can then choose to change the printer configuration. When the next menu appears, you can scroll through the list of printers available by pointing to the printer name and holding the first mouse button. There are over 150 dot matrix and laser printers supported.

You can select the output destination from the printer configuration box. If you choose to print to a file, the format of the file will match the selected printer.

Since graphics modes are often very much the same on different models of printer from the same manufacturer, try a model that is similar to your printer if yours does not appear on the list. Also try some widely emulated standard such as the IBM Graphics Printer or the Epson FX series.

The printer support was developed by

Autumn Hill Software Inc.
1145 Ithaca Drive
Boulder, Colorado 80303
USA
Telephone (303) 494-8865

If you need to develop support for a printer not on the present list, you can contact Autumn Hill Software for the Baby Driver kit. This inexpensive software allows you to create drivers for any printer you know the commands for.

Quit

Selecting “Quit” exits the program and returns to the operating system.

Instruments

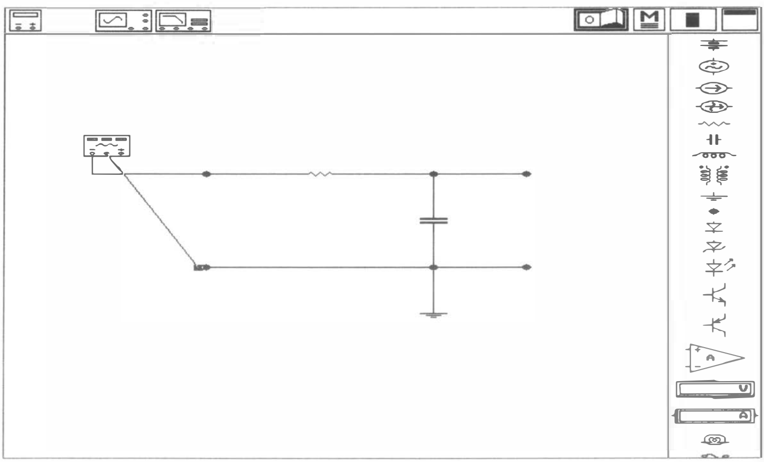
An important feature of ELECTRONICS WORKBENCH is its simulation of electronic test instruments. To see the activity of a circuit, you must attach test instruments to it to measure it.

The simulated instruments rest on the top shelf, above the workspace. There they appear as small icons. Drag the icon down to the workspace to use it. Attach wires from the terminals on the icon to points in the circuit you wish to measure.

The simulated instruments contain controls and displays that are typical of real ones. The controls of the simulated instruments must be adjusted properly to read the display. To adjust the controls and read the display, zoom open the face of the instrument (point to the icon with the tracker and press F7).

connecting test instruments to the circuit

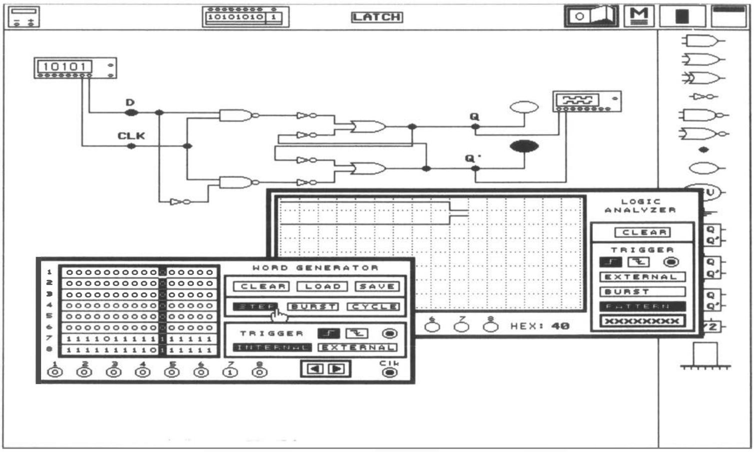
Handle the test instruments the same way as you do components, by dragging their icons from the instrument shelf at the top of the screen to the workspace



window. Connect them to the desired test points in the circuit by stretching wires. The test points must be active terminals that were in the circuit before simulation began, usually connectors inserted into wires. See the chapter on activating the circuit for more information.

setting and viewing the test instruments

After an instrument icon is placed in the workspace window, you can zoom in on the instrument, by pointing

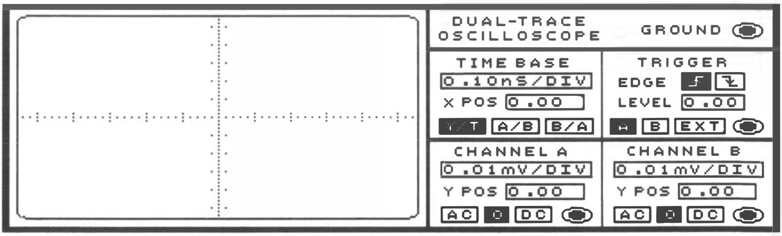


and using F7 or clicking the first mouse button twice. This opens a window containing the instrument face with its controls. All the instruments may be zoomed open this way at once. The instrument windows can overlap and can be easily positioned and stacked. When open, you can move them on the screen with the usual point and drag method. Open windows may be closed by double clicking on any inactive area in the instrument face.

adjusting the instrument controls

For the most part the controls of the instruments mimic the functions of real ones.

The controls on the instrument faces consist of buttons and spin selectors. Press buttons by clicking the first mouse button on them. Change numerical settings by pointing to the number box and holding the first mouse button while moving to the mouse to get the desired value.



You must use the keyboard to enter word patterns in the word generator, Boolean expressions and truth table outputs in the symbolic converter (truth table), and at several other places such as label boxes. Point to a text field and click the first mouse button to put a text cursor into it. Then you can type text in the field.

Buttons

The instruments contain active areas that act as buttons to control functions. They light up when the tracker touches them. Press them by pointing with the cursor and clicking.

Spin selectors

Some of the controls of the instruments are spin selectors that simulate rotary switches and dials. The tracker changes from an arrow to a hand when it touches one of these areas. You can set the value by pointing to the area, pressing the first mouse button and moving the tracker to change from one value to the next.

Many of the spin selectors containing numbers have a second spin selector associated with them that contains units, typically ranging from micro-units to mega-units as appropriate.

Activating the circuit

analog circuits

The on/off switch on the command bar is used to simulate the activity of the circuit. This is the only way to get readings on the instruments for an analog circuit. Clicking on the switch starts the simulation of the circuit (a sequence of mathematical operations to compute values at all the nodes of the circuit) and the cursor changes shape to an hourglass while this process continues.

After simulation is complete, you can use the test instruments to display the values and waveforms of the signal at any node in the circuit. As long as the simulation is valid, that is, as long as the circuit has not changed, the wires from the oscilloscope and the multimeter may be moved freely about on the circuit to display the values. When anything in the circuit is changed (components removed or inserted, values of components or settings on the function generator changed) the analysis is no longer valid.

The old values remain at each node in the circuit even after changes are made in the circuit, so if you accidentally clip a wire and restore it, the values can still be read. (This feature is provided because the mathematics of simulating a circuit can take a long time, especially with a complex circuit.)

Remember to select the switch whenever anything in the circuit is changed or the input to the circuit from the function generator is changed. If you wish to stop the simulation before it is complete, simply click the first mouse button.

In summary:

- Analog simulation always requires the on/off switch to simulate the circuit.
- The values remain in place at the nodes of the circuit and remain true for that configuration.
- To see the effect of changes in the circuit, you must press the on/off switch again.

The controls on the oscilloscope and its connections in the circuit may be changed to display the steady-state or first transient cycle waveform at a different scale or place without pressing the on/off switch again.

Error conditions

If the circuit is improperly grounded (including any test equipment on the workspace) you will get an error message when you start a simulation.

If the circuit is improperly constructed, it will not function and no mathematical simulation is possible. When this happens, a confirmation box will give an error message.

You must determine the cause of the failure by inspection.

digital circuits

The simulation of a digital circuit is somewhat different from that of analog circuits.

The on/off switch icon activates any output device in the digital circuit, whether it is a TTL voltage source or the word generator, and the circuit with any indicators it includes changes to its next stable state.

The word generator may be operated from its zoomed open face, however. Selecting "STEP," "BURST" or

"CYCLE" causes the contents of the word generator to be sent to the circuit and the output displayed immediately on the logic analyzer or indicators. Selecting the on/off switch also acts as clicking on the highlighted button of the word generator, even when the instrument is not zoomed open. "CYCLE" causes the word generator to operate continuously. Pressing the first mouse button will interrupt the operation.

Error conditions

Two error conditions may occur in a digital circuit that halt simulation.

When the order that the input to a circuit is received affects the state of the output, the circuit is said to be in a "race condition." This condition arises in R-S flip-flops and simplified J-K flip-flops, and could appear in any improperly constructed circuit. A dialog box signalling the condition appears and asks the user to cancel or continue the simulation.

Any other error will appear as a logical contradiction in the circuit. A dialog box signals this condition.

In summary:

- Digital simulation can be activated by the word generator itself.
- Using the word generator as input you can press STEP, BURST or CYCLE to start the simulation.
- If a circuit gets its input from voltage sources or the clock in the word generator, click on the on/off switch to activate the simulation.

Analog Module

The following components and instruments are included in the analog module:

	ground connector
passive components	resistor capacitor inductor transformer light-emitting diode bulb fuse relay
active components	npn transistor pnp transistor diode zener diode operational amplifier
dc sources	battery dc current source
ac sources	sinusoidal voltage source sinusoidal current source function generator
test instruments	ammeter (in parts bin) voltmeter (in parts bin) multimeter oscilloscope Bode plotter

The parts bin holds an unlimited number of each object; only one copy, however, of each test instrument on the top row is available.

Using ELECTRONICS WORKBENCH, you construct a circuit with a combination of components, instruments, wires, and connectors.

Connections

All components and instruments have wire terminals. Point to the terminal with a tracker, then press and hold the first mouse button to drag a wire out. Move the tracker to another terminal (which will highlight when touched) and release the mouse button to connect the wire. A single wire joins two terminals together at a single point. Connectors may also be inserted into the wire to allow more terminals to be connected to the same point.

Connectors are necessary to join two wires together. In addition to connecting wires within the circuit, connectors let you set up test points. Before you simulate your circuit, put connectors into wires wherever you will want to measure values. After simulation, you can attach instruments and measure values at any point without running the simulation again.

Ground

The ground symbol provides an unambiguous point of reference for the circuit solution, and is assigned a known voltage of zero value. This component may be placed freely throughout the circuit. All terminals to which this symbol is connected, however, represent a common point, and are treated as joined together through ground.

Although it is possible to obtain a solution without grounding a circuit, the mathematical solution may not remain consistent in different runs. Moreover, it is good electrical practice to include a ground in a circuit. To maintain consistent results, the analysis will return an error message if it attempts to analyze a circuit which is improperly grounded.

Values

You must assign values to all components in the circuit. The passive components all have a default value of 0 in the parts bin. The active components have default models that can be modified as explained below.

You can set the values of resistors, capacitors, inductors and voltage and current sources to any number you wish. Point to the component and press F6 or select it and choose “Label” from the menu; then type in the value.

The frequency of the alternating voltage and current sources is controlled by the setting of the function generator. After the frequency is set, it is saved with the circuit, so you do not need to save the function generator on the circuit.

The internal resistance of the ammeter and voltmeter in the parts bin comes from the setting of the multimeter. To adjust the value, select the multimeter and choose “Label” (F6). A dialog box lets you adjust the parameter for the currently selected function. See the description of the multimeter for more details.

Connectors and ground are not adjustable, though you can label these components with names if you wish.

Labels

All components, including connectors and ground, can be labelled by either pointing to the component and pressing F6, or by selecting the component permanently, choosing “Label” from the menu, and entering the name you wish.

If you have more than one component permanently selected and choose “Label” from the menu, the label box will act on each selected component in turn. While the box is open, only the component being acted on will remain highlighted. When you have finished the group, the label box closes and all components are again highlighted.

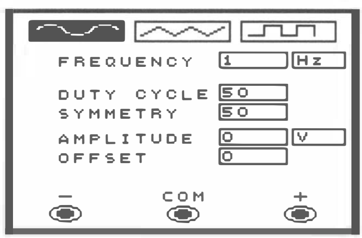
Active component models

The models for the diodes, the two bipolar transistors, the operational amplifier and the transformer can be customized by either pointing to the component and using F6, or by selecting the component permanently and choosing “Label” from the menu. This will provide you with the option to modify parameters. Select “Model” and a list of model parameters will appear. For an explanation of the function and the range of appropriate values for these parameters refer to the section on analog component models in the technical exposition chapter on page 11-1.

You can save and reload different models. In order to use your customized models in future simulations, save your changes using the “Save changes” option for the model, and then save the parts bin.

instruments

Function generator



The function generator is a voltage source that supplies analog signals in the form of sine, square and triangular waves. It provides a convenient and realistic way to supply power to a circuit under test. You can easily change the waveform among sine, square and triangular forms and control the frequency and strength of the signal. The range of the generator is great enough to produce conventional current as well as audio and radio frequency signals.

When the face of the function generator is zoomed open, the controls must be adjusted to produce the desired waveform.

Waveform

The form of the signal is selected by clicking on the sine wave, the triangular wave, or the square wave button.

Frequency

The frequency value can be spin selected from 1 to 999 and the multiplier can be selected from Hz to MHz. The frequency set here also applies to the alternating power sources in the parts bin.

Duty cycle

“Duty cycle” can spin from 1 to 99 percent and affects the shape of the square and triangular waves.

Symmetry

“Symmetry” controls the amount of signal generated above and below the d.c. level of the signal. Values range from 1 to 99 %, where 50 % provides rms symmetry about the d.c. level (i.e. positive and negative peaks are the same distance from the offset). Square wave output requires the duty cycle and symmetry to be equal for the peaks to be symmetrical.

Amplitude

“Amplitude” controls the value of the wave from its d.c. level to its peak value. This is the same as the difference between the COM and + or – terminals. If the output leads are connected to COM and to + or –, the peak to peak measurement of the wave equals twice the amplitude value. If the output comes from + and –, the peak-to-peak value will be four times t he amplitude value.

Offset

"Offset" controls the d.c. level about which the alternating signal varies. The multiplier for "Amplitude" applies to the offset as well.

+

The positive terminal provides a signal with the selected amplitude in the positive direction from neutral COM.

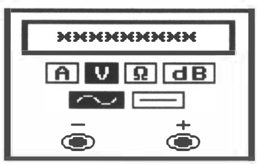
COM

The common terminal provides a reference level signal.

–

The negative terminal provides a signal with the selected amplitude in the negative direction from neutral COM.

Multimeter



Use the multimeter to measure electrical signals and components to determine voltage, current, resistance or decibel loss between two points in the circuit. You can switch modes to measure current, voltage, resistance or decibels.

Ideal meters would have no effect on the circuit being measured. An ideal voltmeter would have infinite resistance, so no current flowed through it while attached to the circuit. An ideal ammeter would present no resistance to the circuit. Real meters do not achieve this ideal, so their readings will not match theoretical, calculated values for a circuit.

ELECTRONICS WORKBENCH uses very small and very large numbers that approximate zero and infinity to calculate near ideal values for the circuit.

If you wish, you can make the behavior of the meter more realistic, by changing the value used to model its action. To do so, select the meter and choose “Label” (F6) to open a box that allows you to enter the values for the meter’s functions.

Take care when reducing the ammeter resistance. If the readout is zero, then your value is too low.

When the face of the multimeter is zoomed open, the controls must be used to obtain proper readings.

AMP

Displays the current through the circuit at the test point. The instrument must be inserted into the circuit to measure current flow. Note that it is not possible to switch from measuring voltage across points of the circuit to measuring current in the circuit without connecting the multimeter appropriately, clicking on the “AMP” button, and clicking on the on/off switch to activate the new configuration.

VOLT

Clicking on this button allows you to measure voltage between any two points in the circuit. After the circuit has been simulated with the on/off switch, the probes may be moved around to test values at any node in the circuit.

OHM

This setting measures resistance between the probes. Note that a component cannot be in a closed circuit to get an accurate measurement. Total resistance between points in a resistive network may be measured. If you want to use the meter to measure the value of a resistor on the workspace, one side of the component must be grounded.

dB

Clicking on this button will cause the multimeter to read in decibels. If you need to change the standard base for calculating dB, point to the multimeter icon and press F6 to Label it. A box will open allowing you to adjust the internal parameters of the instrument.

AC (sine wave symbol)

This button causes the meter to display the root mean square value of an alternating signal with the DC component removed.

DC (straight line symbol)

This button causes the meter to display the direct current value of a signal. When an AC signal is measured with DC selected, the average signal strength is displayed (as opposed to the RMS value).

–

The negative terminal.

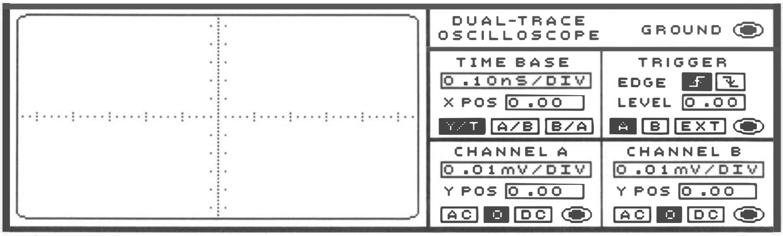
+

The positive terminal.

Oscilloscope

The oscilloscope displays the amplitude and frequency variations of electronic signals. It provides a graph in time of the strength of the signal.

When the face of the oscilloscope is zoomed open, the controls must be used to obtain a proper display of the waveform.



The oscilloscope has two input channels, A and B, allowing two different signals to be displayed simultaneously. Their controls are on the lower right of the scope face.

TIME BASE

The "TIME BASE" controls the scale of the x-axis on the display. You must adjust it relative to the signal frequency to get a readable display. The TIME BASE box is a spin selector with values ranging from 0.1 nanoseconds to 0.5 seconds per horizontal division. Thus if you want to see one cycle of a 1000 Hertz signal, the TIME BASE should be 0.10 milliseconds. One cycle at 10 KHz requires a TIME BASE of 0.01 milliseconds.

X POS

Change the value at "X POS" to move the trigger point along the x-axis. Note that the intersection of the vertical and horizontal scales on the scope does not represent zero on the x-axis. The trigger point is at the left edge of the display when X POS is zero.

Y/T | A/B | B/A

The axes of the oscilloscope display can be switched to show amplitude against time (Y/T) or one input channel against the other (A/B or B/A). The latter settings allow you to display phase shifts, known as Lissajous figures.

ground

The ground symbol indicates where ground should be connected on the scope icon. The oscilloscope must be attached to a ground reference point in the circuit for accurate displays. When not grounded, the scope is influenced by 60 Hz noise.

TRIGGER

Triggering controls when the waveform begins to display. This may be adjusted to start the trace on a positive or negative slope of the input signal on channel A, channel B, or an external signal. An external trigger signal may be attached to the terminal on the scope icon. The level of the signal at which the display triggers can be set with the spin selector.

If you fail to get a display on the scope when you think you should, try connecting the trigger to ground and setting “EXT” on. This will cause any signal received by the scope to display.

/DIV & Y POS

The value of the divisions on the vertical scale and the origin on the y-axis may be adjusted independently for each channel with these spin selectors.

To get a readable display you must adjust the y-scale appropriately. An input signal of 1 volt will fill the screen of the oscilloscope vertically if the y-axis is set to 0.1 volts/division.

If you want to separate channels A and B by some vertical distance to compare their waveforms, spin select the y-position (Y POS) for each channel to move its display up or down the screen.

AC | 0 | DC

Each channel may be switched to show the AC component of the signal or the sum of DC and the AC components. Selecting 0 shows a flat line at the origin set by Y POS.

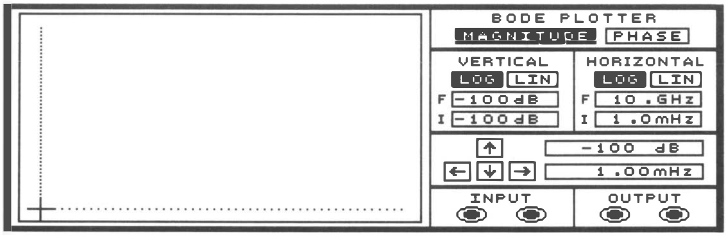
The AC setting performs the same function as a coupling capacitor in series with the scope test probe. Do not place a coupling capacitor in series with a scope lead, as the scope does not provide a path for current and the analysis will consider the capacitor improperly grounded.

If the analysis of a circuit is still valid, the probes of the oscilloscope may be moved to another node and the signal displayed without simulating the circuit again. Moving the probes automatically redraws the trace.

If you need to adjust the scale of the display at all, you can change the settings and then just click on AC or DC to redraw the scope face with the new signal. This lets you fine-tune the display without repeating what could be a long simulation. If you find that the curve is uneven when enlarged, adjust the settings until you have the proportions you like and simulate the circuit again to get more detail in the range chosen.

Bode plotter

The Bode plotter shows the ratio of output signal to input signal against frequency. This gives you a graph of the frequency response of a circuit, such as a bandpass filter.



The plotter controls its own sweep frequency generator that varies the input signal to the circuit, so the frequency of the source to the circuit is not relevant to this display. However, note that a signal source must be included in the circuit.

You can set the display to show either the MAGNITUDE of the signal or its PHASE.

By default, the F[inal] and I[nitial] values of the vertical and horizontal scales are at maximum. After a simulation is finished, you can change these values to see the plot at a different scale. If you expand the scale a great deal, you may want to click on the switch to run the simulation again to get sufficient detail in the plot.

VERTICAL

The scale can be either linear (LIN) or logarithmic (LOG). Set the F[inal] and I[nitial] values of the scale with the spin selectors.

HORIZONTAL

The horizontal axis can be either linear (LIN) or logarithmic (LOG) and you must use the spin selectors to set the initial (I) and final (F) frequency of the scale.

Readouts

Moveable crosshairs on the instrument provide direct readout of the frequency and amplitude of any point on the display. The crosshairs are in the lower left corner of the Bode plotter screen to begin with.

There are two ways to move the crosshairs:

1) point to the arrow symbols on the control panel of the Bode plotter and press the first mouse button to move the crosshairs in the indicated direction; or

2) point to the crosshairs and press and hold the first mouse button to drag them directly to the desired place.

The amplitude and frequency of the intersection of the crosshairs on the display will be shown in the boxes beside the arrow symbols.

You can toggle the base scale between linear and logarithmic without simulating the circuit again. But with some responses, switching the base may result in missing some important frequency points, giving a ragged, inaccurate graph. Click on the on/off switch to simulate the circuit again to get accurate results.

You can also change the frequency markers after simulation. If you are not certain of where to set the initial and final frequencies to begin, allow a large range and then fine tune the display by adjusting the settings. If you want to see more detail after adjusting, just simulate the circuit again.

Digital Module

The following components and instruments are provided in the digital module:

	connector
fixed levels	ground TTL voltage
gates	and gate or gate xor gate not gate nand gate nor gate
devices	R-S flip-flop J-K flip-flop D flip-flop half adder
displays	probe (LED) seven segment display
test instruments	multimeter word generator logic analyzer truth table

components

The parts bin holds an unlimited number of each component.

All components, including connectors, can be labeled with F6.

Use the connector to join wires or set up test points within the circuit. You can attach the instruments to the test points to observe the action of the circuit.

The TTL voltage can be used to hold a point high (+5v) and ground to hold a point low.

The gates perform their logical functions as ideal devices.

The R-S flip-flop is a simple, unclocked, level-triggered device.

The J-K flip-flop is a more sophisticated and complex device, being clocked, negative-edge-triggered and having direct set and clear lines.

The D flip-flop is clocked and positive-edge-triggered.

The half adder computes the binary sum of two inputs and provides the sum and carry out on its terminals.

The probe serves to indicate high (on) or low (off) levels at any point. It lights or goes off as the circuit is running.

The seven segment display will actively show its state while the circuit is running. The seven terminals (left to right respectively) control segments *a* to *g*.

instruments

Multimeter

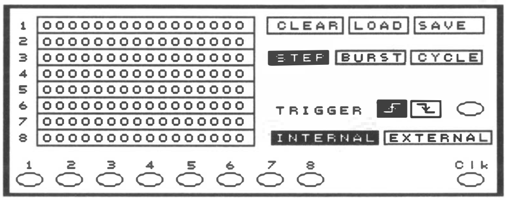
The multimeter acts only as a voltmeter in digital simulation. It can be used to determine the level of the circuit at any point in it, with high levels reading +5 volts and low levels reading 0 volts.

Word generator

Use the word generator to send patterns of bits into circuits you have built with the program to test them. The generator lets you save and reload these patterns.

To adjust the controls of the word generator, zoom the icon open to show the instrument face.

The left half of the face of the word generator contains input fields for digital words. The generator can hold sixteen eight-bit words. Each vertical column of the display represents a word; there are sixteen columns. The horizontal rows are numbered with the bit position. When the word generator is activated, a column is sent in parallel out the correspondingly numbered terminals at the bottom of the device.



To enter values into the word fields, point to the desired position with the tracker and click the first mouse button to select the area for text entry. A text cursor will appear where you clicked. Now you can type in ones or zeros to form the word pattern.

As the words are transmitted by the generator, the value of each bit appears in the circles representing the output terminals at the bottom of the instrument face.

CLEAR | LOAD | SAVE

The buttons across the upper right allow the word patterns to be cleared, saved in a file for future use, or loaded from a file. Selecting “SAVE” or “LOAD” causes a file selector box to appear. Follow the procedures explained in the section on file operations in the menu operations chapter.

STEP | BURST | CYCLE

The words in the storage field can be injected into a circuit by clicking on the "STEP," "BURST" or "CYCLE" buttons. A vertical bar highlights the current word in the display. "STEP" transmits one word at a time and moves the bar one position to the right. "BURST" sends all sixteen words in sequence, starting at the current position in the display. "CYCLE" continues to repeat the stored patterns until the first mouse button is pressed again.

TRIGGER

The "TRIGGER" terminal is an input to drive the word generator and each input cycle causes a word to be transmitted.

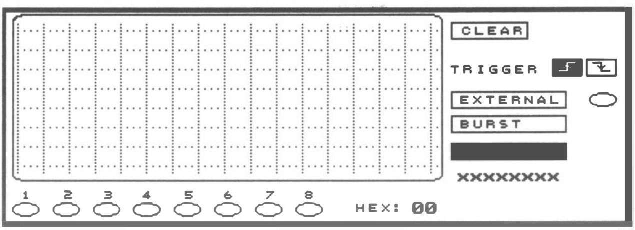
Clk

A clock pulse is produced by the word generator and is available at the clock (Clk) terminal, one complete positive and negative cycle for each advance in the display.

Logic analyzer

Use the logic analyzer to display the levels of TTL signals within the circuit. It can display up to eight signals at once.

To adjust the controls of the logic analyzer, zoom the icon open to show the instrument face.



The left half of the face of the logic analyzer displays the state (high or low) of the signals received on each input channel. The numbered terminals across the bottom of the instrument correspond to the horizontal rows of the display.

The logic analyzer displays a signal as a square wave against time, and as binary and hexadecimal values for all bits of the current input word across the bottom of the instrument face.

The display can be made to trigger on the positive or negative edge of a signal, which may be an external signal, a burst of any data (the default), or a particular pattern of data. The pattern can be entered by pointing to the box below pattern (which begins as a word of *x*'s) and entering ones and zeros to form the desired pattern. An *x* signifies that either value is accepted in that location.

"CLEAR" resets the display of the logic analyzer.

Truth table

The truth table/symbolic converter is able to perform several transformations of a circuit representation.

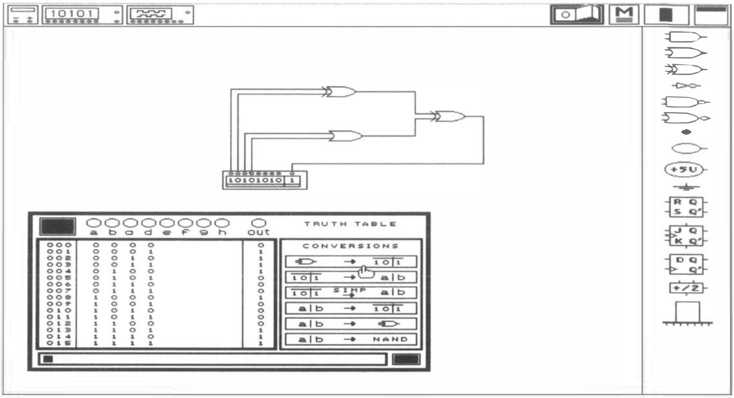
The truth table is able to convert logic circuits among three different representations — logic gates, truth table and Boolean expression. This device has no real-world counterpart. You can attach it to a circuit to derive the truth table or Boolean function the circuit embodies, or you can use it to produce the circuit of logic gates from a truth table or Boolean expression.

To adjust the controls of the truth table, zoom the icon open to show the instrument face.

Deriving a truth table

[circuit to truth table]

Place the icon for the truth table on the workspace and attach it to points in the circuit using up to eight “inputs.”



Attach the single output of the circuit under test to the final terminal of the truth table icon.

To display the truth table for the circuit, click on the “circuit → truth table” button.

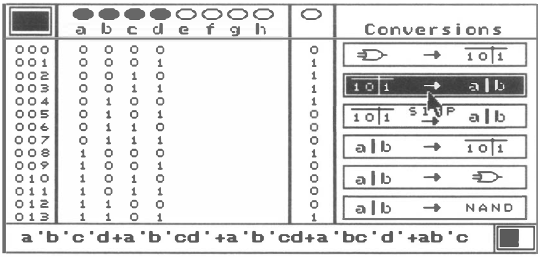
Transforming a truth table

The other function of this device is to generate a circuit.

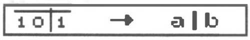
You can input to the truth table/symbolic converter in two areas.

Construct a truth table by clicking on the input channels (*a* – *h*) across the top to turn them on. The display area below the terminals fills up with the necessary combinations of ones and zeros to fulfill the input conditions. The values in the output column to the right are all zero initially. You can edit the output column to specify the desired output from the input conditions. An *x* in the output column indicates either value is acceptable. Point to the column with the tracker and click the first mouse button to select it for text input.

You can enter a Boolean expression into the text box at the bottom. You can use either sum-of-products nota-



tion or product-of-sums. To handle long expressions, scroll this box with the icon to its right.



[truth table to Boolean expression]

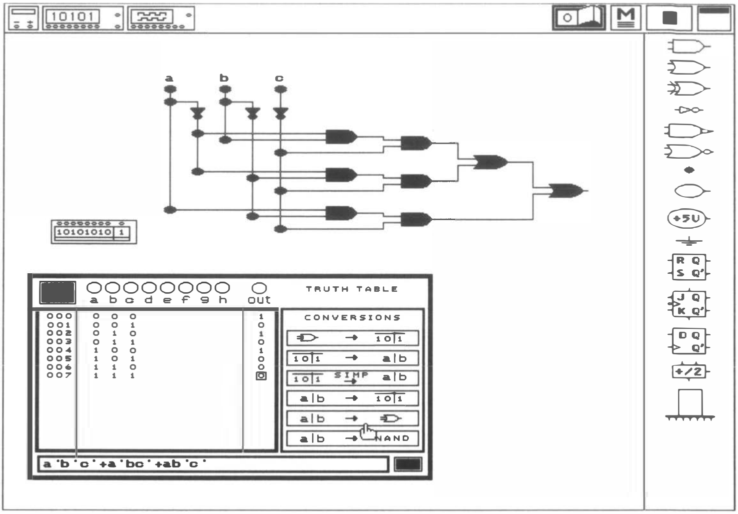
Click on this button to transform the truth table into a Boolean expression.

[to simplified Boolean expression]

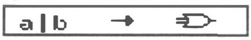
This will transform the truth table into a simplified Boolean expression. The simplification is performed by the Quine-McCluskey method, rather than the more familiar Karnaugh mapping technique. Karnaugh mapping works for only small numbers of variables and requires human intuition, while Quine-McCluskey is proven to be exhaustive for any number of variables but is too cumbersome for manual solution.

It is possible for this operation to use up all available memory.

[Boolean expression to truth table]



You can enter an Boolean expression into the text box on the bottom to the truth table. To convert the expression to a truth table, click on this button.



[Boolean expression to circuit]

This converts a Boolean expression to a circuit. You may place this anywhere on the workspace by clicking the first mouse button. All the components of the new circuit remain selected, so you can move them again, or do other operations easily. Deselect the components by clicking the last mouse button.

[circuit from only NAND gates]

Click on the button to generate a circuit fulfilling the conditions of the Boolean expression using only NAND gates.

Technical exposition

In order to display the activity of the circuit on the test equipment in the program, the activity must be simulated by mathematical means. This chapter explains some of the factors entering into the simulation and some of the mathematical models and techniques used. In this chapter, we assume that you already know something of the theory of electronic circuitry and the mathematics involved. This information is for teachers and advanced users, so you may want to skip it for now. You can make full use of ELECTRONICS WORKBENCH without knowing how its simulation works.

The methods ELECTRONICS WORKBENCH uses to simulate digital and analog circuits differ fundamentally. The digital simulation deals only with logic levels and does not attempt to portray electrical values directly. The analog simulation does determine electrical values at all points in the circuit, using an appropriate method as explained below.

digital simulation

To design and test a digital circuit, you typically:

- attach the outputs of the word generator to the inputs of the circuit you wish to analyze,
- attach the outputs of the circuit to the inputs of the logic analyzer,
- specify the sequence of values for each of the output terminals of the word generator,
- press a button on the word generator to cause it to go through all of its states,
- observe the resulting plots on the logic analyzer.

Digital simulation performs an analysis that determines the logic values at all points in the circuit based on:

- the values of all active inputs (word generator or TTL voltage sources);
- the previous state of the circuit.

A circuit whose outputs depend only on the present values of input sources is called “combinational.” If the outputs depend on the previous state of the circuit, the circuit is called “sequential.” Such circuits usually contain feedback, i.e. outputs that are connected to the inputs of earlier gates. It is not always possible to analyze sequential circuits, since they may exhibit race conditions which can cause unpredictable behaviour dependent on the propagation times of the various signals.

Even a purely combinational circuit may not be analyzable if it contains a “contradiction.” In reality, such circuits are either self-destructive (e.g. shorted circuits) or circuits in which a stronger gate dominates and overrules a weaker one. Both of these cases represent design errors, which are automatically caught and reported by ELECTRONICS WORKBENCH.

To understand how ELECTRONICS WORKBENCH performs digital simulation we must consider:

- the inputs to the circuit,
- the operation of the various circuit components and their interconnection,
- the method used by ELECTRONICS WORKBENCH to determine the total circuit behaviour from the behaviour of the parts,
- the methods for viewing the outputs of the circuit.

Circuit inputs

There are three basic kinds of input values provided by ELECTRONICS WORKBENCH:

Fixed values

The ground symbol represents a fixed logical 0 value and the TTL Voltage source represents a fixed logical 1 value (these are represented physically by voltage levels of 0 and +5 volts respectively).

Clock value

The clock value is available at the Clk terminal of the word generator. A complete clock cycle consists of two clock states: a 1 followed by a 0. After each clock cycle the word generator advances to the next set of values. ELECTRONICS WORKBENCH performs a separate analysis for each one of the two clock states, thus one complete pass through the 16 sets of values of the word generator causes ELECTRONICS WORKBENCH to perform 32 simulations. The result of each simulation is displayed on the logic analyzer (if it is attached).

The logic analyzer is implicitly connected to the clock, since its display must be synchronized with the signal produced by the word generator.

Word generator

The word generator has 8 output terminals. For each of the output terminals you can specify a sequence of 16 values. There are a number of ways you can control the time when the word generator is to switch to the next set of output values.

The word generator gives you three ways of going through its sets of values:

- "STEP" causes the word generator to go to the next state. This can be thought of as one clock cycle, i.e. two simulations.

- "BURST" causes the word generator to go through all of its output values once, i.e. 16 clock cycles are generated, causing 32 simulations of the circuit.

- "CYCLE" causes the word generator to commence cycling through its values until you press a mouse button.

A note about "Time"

ELECTRONICS WORKBENCH does not operate in "real time." That is, there is no relationship between your time and the time marked off by the word generator clock. This means that we cannot meaningfully talk about the clock rate. We simply assume that the clock rate is slow relative to the propagation delays in the components.

In fact, when you operate the word generator, you are, so to speak, taking time into your own hands. This is something we cannot do when we deal with real circuits and lab equipment, which demonstrates an important advantage of a computer simulation.

Components and connections

The digital circuit components fall into the following rough categories:

- Combinational logic components (AND, OR, XOR, NOT, NAND, NOR, Half adder)

- flip-flops (D, R-S, J-K)

- inputs: ground, TTL voltage source

- outputs: probe, 7 segment display

- connectors

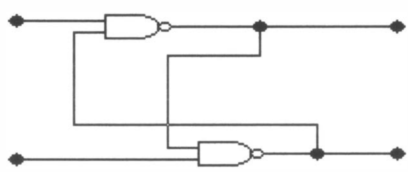
Combinational logic components

The output of a logic gate (such as an AND gate) is a simple function of the inputs.

Flip-flops

A flip-flop is a device that maintains an internal state. That is, its output depends both on its current and past inputs. This is the basis of computer random access memory.

Flip-flops are used in most sequential circuits. Note, however, that it is possible to construct a flip-flop out of simple combinational components. The trick involves feeding the output of a circuit back to its inputs, thus achieving a self-maintaining stable state.



Sequential use of feedback in a simple flip-flop.

Inputs: ground, voltage source

These components have a single output with a fixed value (0 Volts for ground, 5 Volts for the voltage source).

Strictly speaking, a voltage is always defined between two points. For convenience, we usually take one of the points as ground and then use the expression "voltage level at point X" to stand for "voltage between the point X and the ground." The ground itself, by this definition, has a voltage level of 0.

The ground component in ELECTRONICS WORKBENCH stands for a connection to the ground. The 5 Volt voltage source could be thought of as the "+" terminal of a 5 volt battery whose "–" terminal is connected to ground.

Outputs: probe, 7 segment display

These components provide a mechanism to view the logic level at various points in the circuit.

The probe lights up when its single input terminal is attached to a non-zero voltage level. In reality, this could be thought of as a light bulb with the other terminal attached to ground.

The seven segment display is simply a collection of 7 probes arranged in a pattern that can be used to represent digits.

Connectors

Connectors are completely passive electrically, and serve merely to connect two or more wires.

Computing the solution

The method used by ELECTRONICS WORKBENCH to determine the total circuit behaviour from the behaviour of the parts:

- repeatedly go through the circuit, computing the outputs of each component based on the inputs.

When an output value is computed, the value becomes the input to all components that are attached to the output with wires.

The process terminates when:

- a stable state consistent with the inputs to the circuit is reached, or

- the circuit fails to stabilize (indicating a race condition).

Certain assumptions about the components and circuit are made in order to carry out the digital simulation. The components are ideal, exhibiting none of the variations that may complicate actual circuit design. Moreover, we ignore the problems presented by propagation delays, so any circuit whose real-life behaviour depends on propagation delays is not simulated properly. In some circumstances, such circuits could result in race conditions in ELECTRONICS WORKBENCH, though their behaviour may depend on prior conditions in such a way that their actions may not be repeated.

Viewing the circuit output

Output from the circuit can be observed on electronic components intended for display or on test equipment. Probes and seven-segment displays can show the state of the circuit and the logic analyzer and digital multimeter can display signal levels throughout the circuit. See the sections of this manual describing their functions to read more.

boolean simplification

The automatic truth table uses straightforward methods to perform most of its transformations, but the process of Boolean simplification may be of interest.

Rather than the Karnaugh mapping technique usually taught to beginning students, ELECTRONICS WORKBENCH uses the Quine-McCluskey method, which is a tabular technique that guarantees the second order (two levels of gates) realization of a sum of products expression with the fewest products possible.

The Quine-McCluskey method begins by sorting the minterms (sum of product combinations) of a truth table according to the number of 1s they contain. The minterms are then compared to find those that differ in only one variable and the variable eliminated. This process is repeated until no further elimination of variables is possible. The remaining minterms that cannot be grouped are prime implicants, i.e. elements required in the final expression, though there may be redundancy among the expressions. The final step in simplification is to eliminate redundant expressions. Working with the binary representation of the expression allows the process of factoring a Boolean expression to be done automatically with this method.

Note that to simplify a Boolean expression in the program, it must be converted into a truth table. The simplification operates on a truth table, not on a Boolean expression directly.

Starting with a given truth table, the following steps illustrate the process used to produce a simplified Boolean expression.

Reduction & formation of prime implicants

We begin with the following arbitrary truth table. The Boolean expression of each minterm is given along with the number of 1s each minterm contains.

n	A	B	C	D	X	minterm	# of 1s
0	0	0	0	0	1	a'b'c'd'	0
1	0	0	0	1	0		1
2	0	0	1	0	0		1
3	0	0	1	1	1	a'b'cd	2
4	0	1	0	0	1	a'bc'd'	1
5	0	1	0	1	1	a'bc'd	2
6	0	1	1	0	0		2
7	0	1	1	1	0		3
8	1	0	0	0	1	ab'c'd'	1
9	1	0	0	1	1	ab'c'd	2
10	1	0	1	0	1	ab'cd'	2
11	1	0	1	1	1	ab'cd	3
12	1	1	0	0	1	abc'd'	2
13	1	1	0	1	1	abc'd	3
14	1	1	1	0	1	abcd'	3
15	1	1	1	1	1	abcd	4

This truth table produces the following Boolean expression:

$$f(A,B,C,D) = \Sigma m(0,3,4,5,8,9,10,11, 12,13,14,15)$$

where *m(n)* represents the list of minterms.

This expands to:

f(A,B,C,D) = a'b'c'd' + a'b'cd + a'bc'd' + a'bc'd + ab'c'd' + ab'c'd + ab'cd' + ab'cd + abc'd' + abc'd + abcd' + abcd

Step 1: rearrange the truth table

To begin the process of simplification, make a pass through the truth table and sort it in order of the number of ones in the binary representation of each minterm:

n	A	B	C	D	# of 1s
0	0	0	0	0	0
4	0	1	0	0	1
8	1	0	0	0	1
3	0	0	1	1	2
5	0	1	0	1	2
9	1	0	0	1	2
10	1	0	1	0	2
12	1	1	0	0	2
11	1	0	1	1	3
13	1	1	0	1	3
14	1	1	1	0	3
15	1	1	1	1	4

Sorting the truth table in this order makes it easy to find terms with common factors in the next step.

Step 2: Combine minterms into "1-cubes."

Next, compare the binary form of each term in group i with each term in group i+1. Going through the table methodically this way will let you find minterms that differ in one position only (in binary representation). This position represents a common factor.

For example, term zero from group zero above is compared with each term in group one:

n					# 1s		
0	0	0	0	0	0	⇒	0 X 0 0
4	0	1	0	0	1		

The position where they differ is X-ed out since it does not enter into the value of the combined expression.

The equivalent Boolean notation shows that the term factors out:

a'b'c'd' + a'bc'd' = (b'+b)a'c'd' = a'c'd'

Next, compare term zero with the next member of group one, and if they differ in one and only one position, combine them and X out that position.

n		# 1s		
0	0 0 0 0	0	⇒	X 0 0 0
8	1 0 0 0	1		

The Boolean form gives:

a'b'c'd' + ab'c'd' = (a'+a)b'c'd' = b'c'd'

Rows that don't combine with any other are prime implicants.

When you find a row that combines with another, you should "check off" the row to indicate that it has been used. Rows that are not used (checked off) in any combination are prime implicants and will be added to the bottom of the list. The process of simplification is to use up as many terms as possible this way. Those that are left provide the simplified expression.

This process produces the following table :

Combination	Binary	# of 1s
rows (0, 4)	0 X 0 0	0
rows (0, 8)	X 0 0 0	0
rows (4, 5)	0 1 0 X	1
rows (4, 12)	X 1 0 0	1
rows (8, 9)	1 0 0 X	1
rows (8, 10)	1 0 X 0	1
rows (8, 12)	1 X 0 0	1
rows (3, 11)	X 0 1 1	2
rows (5, 13)	X 1 0 1	2
rows (9, 11)	1 0 X 1	2
rows (9, 13)	1 X 0 1	2
rows (10, 11)	1 0 1 X	2
rows (10, 14)	1 X 1 0	2
rows (12, 13)	1 1 0 X	2
rows (12, 14)	1 1 X 0	2
rows (11, 15)	1 X 1 1	3
rows (13, 15)	1 1 X 1	3
rows (14, 15)	1 1 1 X	3

Step 3: Combine 1-cubes into 2-cubes.

Now repeat step 2 on this table. Again, combine rows that differ in only one position, but now make sure the Xs are in the same place in each of the combined rows. If two new rows contain all the same minterms, but in a different order, then a 2-cube is not formed for the second one since it contains all the same terms. You should check off the rows anyway to show that these values have been used.

rows (0, 4)	0 X 0 0	⇒	X X 0 0
rows (8, 12)	1 X 0 0		
rows (0, 8)	X 0 0 0	⇒	X X 0 0
rows (4, 12)	X 1 0 0		
rows (4, 5)	0 1 0 X	⇒	X 1 0 X
rows (12, 13)	1 1 0 X		

etc.

Note that the second combination has the same result as the first, since it is the same combination of terms, so it is not used in the final solution. Only one combination for each result is necessary, of course.

Completing this process gives the following table:

Combination (2-cubes)	Binary	# of 1s
rows (0, 4, 8, 12)	X X 0 0	0
rows (4, 5, 12, 13)	X 1 0 X	1
rows (8, 9, 10, 11)	1 0 X X	1
rows (8, 9, 12, 13)	1 X 0 X	1
rows (8, 10, 12, 14)	1 X X 0	1
rows (9, 11, 13, 15)	1 X X 1	2
rows (10, 11, 14, 15)	1 X 1 X	2
rows (12, 13, 14, 15)	1 1 X X	2
rows (3, 11)	X 0 1 1	prime implicant

Step 4: repeat

Repeat step 3 until it is impossible to combine any more rows. The maximum number of Xs possible is the number of inputs minus one (unless all possible minterms were contained in the original expression). This is also the number of times you must repeat the combination process.

The final table for this example is:

Combination (3-cubes)	Binary	# of 1s
rows (8, 9,10,11,12,13,14,15)	1 X X X	1
rows (0, 4, 8,12)	X X 0 0	
rows (4, 5,12,13)	X 1 0 X	
rows (3,11)	X 0 1 1	

None of these rows can be combined any more so they are all prime implicants.

Selecting a set of prime implicants

Now that we have reduced the implicants of the table, we must eliminate redundancy among them.

The final expression is a combination of some or all of the prime implicants obtained above, which together satisfy the original logic conditions. A prime implicant is essential if it produces at least one unique check mark in any column of the prime implicant table that we construct next.

Step 1: Construct a prime implicant table.

We will form a prime implicant table from all possible primes discovered during the reduction process. It is ordered in terms of cost (i.e. number of un-X-ed terms in the binary form).

Construct the table by cross-tabulating the necessary terms of the original truth table with the prime implicants discovered above (in this example, the 3-cubes). For each row, mark the column for every variable contained in the prime implicant.

	0	3	4	5	8	9	10	11	12	13	14	15
(8,9,10,11,12,13,14,15)					×	×	×	×	×	×	×	×
(0,4,8,12)	×		×		×				×			
(4,5,12,13)			×	×					×	×		
(3,11)		×						×				

Step 2: Find essential primes.

If there is only one mark in a column, then the corresponding prime implicant is an essential prime implicant, which must be included in the final realization. To find the essential prime implicants, read across each row of this table until you find a column with only one mark in it. Place another mark to the left of the prime implicant on that row, indicating that it is an essential prime implicant. Now, place a mark at the bottom of each column that contains a mark produced by this prime implicant. You may have a mark at the bottom of a column with more than one mark in it as a result of this.

		0	3	4	5	8	9	10	11	12	13	14	15
×	(8,9,10,11,12,13,14,15)					×	×	×	×	×	×	×	×
×	(0,4,8,12)	×		×		×				×			
×	(4,5,12,13)			×	×					×	×		
×	(3,11)		×						×				
		×	×	×	×	×	×	×	×	×	×	×	×

If all of the columns have a mark at the bottom, then all of the minterms are included in some essential prime and we are finished, as in this example.

Otherwise, form a reduced prime implicant table with the unmarked prime implicants on the left and the unmarked minterms along the top and then mark the intersecting members, as above. From this new table, select the least number of different combinations that contain all terms. In large problems or extreme cases this step may need two or more iterations.

Now we can easily produce the simplified Boolean expression from the essential prime implicants. The binary combined forms were produced as the 3-cubes above:

rows	(8,9,10,11,12,13,14,15)	1 X X X ⇒	a
rows	(0,4,8,12)	X X 0 0 ⇒	c’d’
rows	(4,5,12,13)	X 1 0 X ⇒	bc’
rows	(3,11)	X 0 1 1 ⇒	b’cd

The sum of products is then:

$$f(A,B,C,D) = a + c'd' + bc' + b'cd$$

analog simulation

Analog simulation will begin once the on/off switch is selected from the program control options. This is equivalent to turning on the power supply to the Workbench. The test instruments are thus activated, and the voltage and current sources are enabled to provide input signals to the circuit.

The analysis will attempt to simulate the response of the circuit to the input signals. If no errors are found in the circuit construction and layout, then the path the analysis follows depends on the test instruments present in the circuit, and on the Scope Output setting in the Preference Menu.

If the "Transient" preference is selected, a transient time domain analysis is performed with zero initial values assumed. If "Transient to SS" is selected, the calculation continues until steady state is achieved.

If the "AC SS" preference is selected:

1. DC analysis is performed to obtain the operating point. If no AC sources are present, this becomes the complete solution. The parameters for the linear AC models of the active components are computed from the DC operating point solution.

2. If not all AC inputs are sinusoidal, proceed directly to step 3. Otherwise AC analysis is performed. If, from this solution, the active components are found not to be operating in the linear region, then the analysis automatically proceeds to perform step 3.

3. A steady state time domain analysis is performed, with the solution of step 1 taken as the initial point solution. Choosing “Time SS” forces this calculation.

If the Bode plotter is present in the circuit:

A frequency response analysis is performed only after the “Transient” or “Steady state” response is calculated. The linear active model parameters are found from step 1 above.

Step 2 above is repeated for 100 frequencies of operation within the range specified on the Bode plotter face.

In order to obtain a frequency response plot, you must provide an AC source in the circuit, though not necessarily across the input terminals of the Bode plotter. The value of this source does not affect the frequency response; if it is not present, however, the results plotted will have no meaning.

Node voltages and branch current relationships

Passive components

Each common point created by wires and connectors is called a “node.” The circuit solution will have a separate voltage developed at each node. Each branch joining two nodes will have a separate current flowing through it.

The relationships between the terminal voltages, the branch current, and the component value are well understood for two-terminal, passive components. These diagrams give the branch relationships for the following passive components: voltage sources, current sources, resistors, capacitors, and inductors. The current flow is taken from left to right (into node 1).

Voltage sources

$V_1 - V_2 = V$ (current unknown)

$v_1 - v_2 = v$ (current unknown)

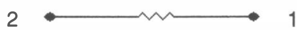
Current sources

current = I

current = i

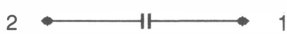
Resistor

$$\text{current} = \frac{V_2 - V_1}{R}$$



Capacitor

DC: current = 0 (open circuit)



linear AC (frequency f): current = $2\pi fC(v_2 - v_1)$

Inductor

DC: current = $(v_2 - v_1)$/MIN_REAL (short circuit)

linear AC (frequency f): current = $(v_2 - v_1)/2\,\pi fL$

Active components

This relationship for active and multi-terminal components is not as straightforward. In order to facilitate the analysis, certain equivalent circuits have been developed which model the behaviour of nonlinear components. Modeling involves transforming the active or multi-terminal component into a combination of two-terminal, passive components whose behaviour can be predicted.

The analog component models that are used for the common nonlinear devices are described in a section below: the diode, the transistor and the opamp, as well as the multi-terminal transformer. Model parameters are described along with some typical values.

You should note that by specifying values for either the frequency components or the series resistors in the nonlinear models, you are adding components to the circuit, and thus you should expect the simulation time to increase. Combining the series resistors with other resistors which may already be present at the diode or transistor terminals can help to reduce computation length.

DC operating point analysis

This analysis involves building a sparse modified nodal admittance matrix unique to the circuit configuration. Entries to the matrix are based on a simple application of Ohm's Law and Kirchhoff's Current Law, using the DC models described below in the Analog Component Models section.

The solution of the matrix is calculated using a partial pivoting algorithm. For simple circuits, this is a one-solution operation. However, if active components are present in the circuit, some model component values will depend on the solution values. In this case, the two steps of building the matrix, and solving it, must be repeated until a stable solution (with the precision specified in the preference menu) is achieved.

For the DC operating point solution, the AC sources are given zero values, and steady state is assumed, i.e. capacitors are open circuits and inductors are short circuits.

AC frequency analysis

This analysis also performs the two steps of building the admittance matrix and solving it using a partial pivoting algorithm. In this case, however, the matrix is complex, i.e. it has both a real and an imaginary component.

DC sources are here given zero values. AC sources, capacitors, and inductors are represented by their frequency models. Active components are represented by linear AC models, derived from the operating point solution.

Time domain analysis

When a non-sinusoidal input waveform is used, or an component such as a diode, transistor, or opamp in the circuit is outside its linear region, then it becomes necessary to perform a time domain analysis of the circuit.

In this analysis, an input cycle is divided into intervals, and a DC analysis is performed for each time point in the cycle. The solution for the voltage waveform at one node will be given by the value of that voltage at each time point over one complete cycle.

DC sources have constant values; AC sources have a time-dependent value. Capacitors and inductors are represented by energy storage models, which use the trapezoid rule to calculate the quantity of energy transfer over an interval of time, and whose matrix entries are dependent on the solution values for the previous time point.

Trapezoid rule

To approximate the value of the integral of the differential equations used in the time domain solution, the trapezoid rule is applied:

$$v_{n+1} = v_n + \frac{h}{2}\left(\frac{dv_{n+1}}{dt} + \frac{dv_n}{dt}\right)$$

where v_{n+1} is the present unknown voltage value, v_n is the previous time point solution, and h is the timestep, or interval length.

The steady state solution is reached when the RMS voltage values are equal, within a certain tolerance, for two consecutive cycles, at each node in the circuit.

Due to the nature of the nonlinear components, each time point may involve solving the admittance matrix several times before converging to a solution. The point solution is reached when the difference between consecutive voltage values is less than the analog precision specified in the preference menu.

The complete solution includes the voltage values for each point in the cycle at each node in the circuit.

According to the scope output preference, the first cycle will become the solution if a transient solution is selected. This allows the user to review the transients of the initial circuit response. If a steady state solution is selected, the analysis will proceed for at least two cycles. Analysis will stop when

$$\left|\frac{v_1 - v_2}{v_2}\right| < 5e\text{-}4$$

is true for each node voltage, where v_1 and v_2 are the RMS voltages for two consecutive cycles. The solution becomes the final cycle of points calculated.

Please note that for circuits with lightly damped responses, you should be prepared for many cycles to pass before steady state is achieved.

Sparse modified nodal admittance matrix

The circuit becomes expressed as a system of equations:

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$$

where **X** is a vector of unknowns with dimension *n*, the number of unknowns. **B** is a vector of constants, also with dimension *n*. **A** is the admittance matrix with dimension *n* x *n*.

The unknowns include each node voltage (excluding ground), as well as the voltage source currents. **B** will contain the voltage and current source constants, and the entries in **A** will be determined by Ohm's Law, Kirchhoff's Current Law, and the voltage differentials across the voltage sources.

Ohm's Law: The current through a component with impedance Z is equal to the voltage differential across the component divided by the impedance, $i = (v_1 - v_2)/Z$.

Kirchhoff's Current Law: The sum of the currents leaving a node is equal to zero.

Voltage Differential: $v_1 - v_2 = V_{in}$.

The matrix is deemed sparse because it contains more zeros than non-zeros. Making use of a linked list, the zero values do not need to occupy a memory location, hence a sparse matrix approach will require less memory consumption.

Partial pivoting algorithm

The sparse matrix approach requires that the Gaussian elimination method be used to solve the system of equations $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$. This follows a pattern of multiplying and subtracting row and column values in such a way that the admittance matrix is reduced to an upper triangular matrix, and the unknowns can be calculated by a simple backwards substitution procedure.

In order to reduce the roundoff error incurred by this method, a partial pivot algorithm is employed. This involves interchanging rows so that each main diagonal element has a value greater than or equal to the column values below it.

Viewing the solution

Once either the complex or the time domain analysis has obtained a solution, the multimeter and the oscilloscope can be used to examine the solution voltages. In volt or decibel mode, the multimeter leads can be moved freely from point to point to obtain either a DC or an AC voltage reading. In the same manner, the scope leads can be moved from point to point to view the various waveforms. Re-evaluation of the solution is necessary only when a component value or input waveform is changed, or if increased detail is desired in the first cycle of the transient response.

Multimeter output

The multimeter leads cannot be moved about in amp or in ohm mode because they have separate models and moving the leads will invalidate the solution. The voltmeter is modeled by a large impedance and therefore moving its leads should not have a noticeable effect. The multimeter models are given in the section below. The values can be altered by the user, and should be taken into account when using the voltmeter in conjunction with large impedances, or the ammeter with small impedances.

In AC mode, the RMS value of the alternating signal is displayed with any DC bias eliminated. In DC mode, the average value of the total signal is displayed.

Oscilloscope output

The oscilloscope basically plots the output straight from the results. If the scope is ungrounded, 1 volt, 60 Hz noise is added to the output waveforms. Which part of the cycle is first displayed depends on the triggering level, edge, and channel. In order to bypass triggering and display immediate output, attach the trigger channel to ground and select the EXT trigger channel.

The DC channel setting displays the sum of the AC and DC waveforms, while the AC setting eliminates the DC component. The AC setting performs the same function as a coupling capacitor in series with the scope lead. Since the oscilloscope is simply a display instrument, and is not recognized as part of the circuit, it will not affect the DC biasing of the circuit.

DO NOT place a coupling capacitor in series with a scope lead, as the oscilloscope will not provide a path for current and the analysis will consider the capacitor improperly connected.

analog component models

Capacitor models

Characteristic equation

The current through the capacitor is equal to the capacitance multiplied by the change in voltage across the capacitor, i.e.

$$i = C \frac{dv}{dt}$$

DC operating point model

The capacitor is represented by an open circuit. This is in fact modelled by a very large impedance.

AC frequency model

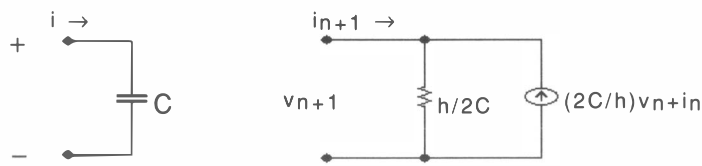
The capacitor is modeled by an impedance with its imaginary component equal to 2π multiplied by the frequency of operation of the circuit multiplied by the capacitance value, i.e. $2\pi fC$.

Time domain frequency model

Applying the trapezoid rule to the characteristic equation, it becomes:

$$i_{n+1} = \frac{2C}{h} v_{n+1} - \left(\frac{2C}{h} v_n + i_n \right)$$

where v_{n+1} and i_{n+1} are the present unknown voltage across and current through the capacitor, v_n and i_n are the previous solution values, and h is the time step.



Capacitor time-domain energy-storage model.

Inductor models

Characteristic equation

The voltage across the inductor is equal to the inductance multiplied by the change in current through the inductor, i.e.

$$v = L \frac{di}{dt}$$

DC operating point model

The inductor is represented by a short circuit. This is in fact modeled by a very small impedance.

AC frequency model

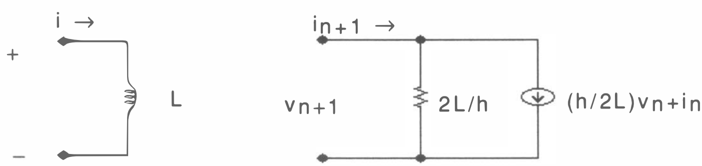
The inductor is modeled by an impedance with its imaginary component equal to the negative inverse of 2π multiplied by the frequency of operation of the circuit multiplied by the inductance value, i.e. $-1/2\pi fL$

Time domain frequency model

Applying the trapezoid rule to the characteristic equation, it becomes:

$$i_{n+1} = \frac{h}{2L} v_{n+1} + \left(\frac{h}{2L} v_n + i_n \right)$$

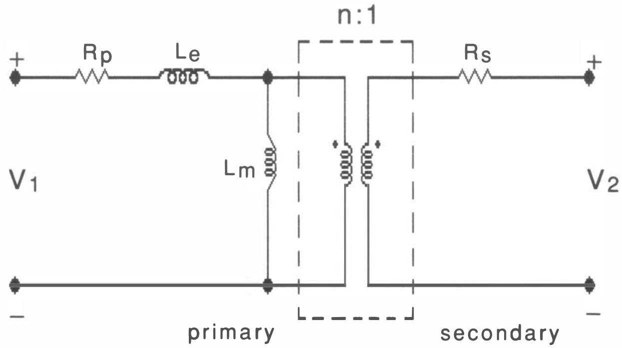
where v_{n+1} and i_{n+1} are the present unknown voltage across and current through the inductor, v_n and i_n are the previous solution values, and h is the time step.



Inductor time-domain energy-storage model.

Characteristic equation

$V_1 = \quad nV_2(\text{ideal})$



Parameters

n : turns ratio; if n > 1 it is a step-down transformer, if n < 1 it is a step-up transformer.

L_e : leakage inductance

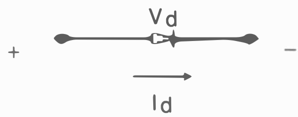
L_m : magnetizing inductance

R_p : primary winding resistance.

R_s : secondary winding resistance; resistance in both windings is ideally zero, but if there are source or load resistances, choose R_p and R_s to maximize power output.

Characteristic equation

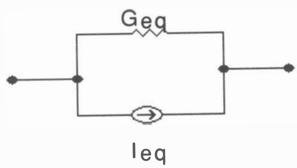
$$I_d = I_s \left(\exp \frac{V_d}{nV_t} - 1 \right)$$



DC operating point model

$$I_{eq} = I_d - G_{eq}V_d$$

$$G_{eq} = \frac{I_s}{nV_t} \exp \frac{V_d}{nV_t}$$

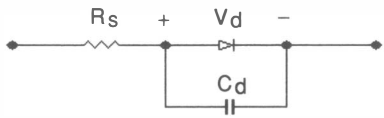


AC frequency model

$$C_D = C_d \Big|_{op}$$

The illustration below for the time domain frequency model shows the model used for AC frequency analysis if you substitute C_D in place of C_d.

Time domain frequency model



$$C_d = \frac{\tau_t I_s}{nV_t} \exp \frac{V_d}{nV_t} + C_{j0} \left(1 - \frac{V_d}{\phi_B} \right)^{-m}$$

Equation parameters

I_s : saturation current; typically 10^{-14} to 10^{-15} Amperes, but cannot be zero.

V_t : thermal voltage; assumed equal to 0.025 Volts.

n : emission coefficient; assumed equal to 1.

Optional parameters

R_s : ohmic resistance model; typically 1 Ω.

C_{j0} (Cd) : zero bias junction capacitance; typically 0.1 to 10 pF.

ϕ_B (P) : junction potential; typically 0.5 to 0.7 Volts.

τ_t (T) : transit time; typically 1 ns.

m : junction grading coefficient; typically 0.33 to 0.5.

Zener diode model

The zener diode model is the same as the diode model above, except in the reverse breakdown region, where another diode is simulated.

In this region, an inverted diode is simulated. The value of I_S remains the same, while the thermal voltage is calculated by:

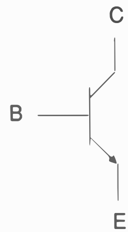
$$V_T = \frac{V_{zt}}{\ln\left(\frac{I_{ZT}}{I_S} + 1\right)}$$

V_{ZT}: test voltage

I_{ZT}: test current

Bipolar transistor model

Characteristic equations



$$\beta_F \rightarrow \alpha_F = \frac{\beta_F}{1+\beta_F}$$

$$\beta_R \rightarrow \alpha_R = \frac{\beta_R}{1+\beta_R}$$

$$I_{CC} = I_S\left(\mathrm{xp}\,\frac{V_{BE}}{V_t} - 1\right)$$

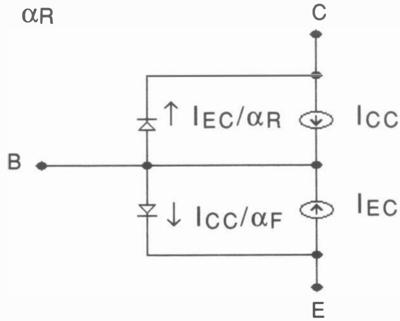
$$I_{EC} = I_S\left(\exp\frac{V_{BC}}{V_t} - 1\right)$$

DC operating point model

To determine the DC operating point, a simplified Ebers-Moll model is used with the capacitors treated as open circuits.

$$I_E = -\frac{I_{CC}}{\alpha_F} + I_{EC}$$

$$I_C = I_{CC} - \frac{I_{EC}}{\alpha_R}$$



AC frequency model

Using parameters derived from the DC operating point analysis, a hybrid-π model is used for AC frequency analysis while the transistor is in the forward active region of operation.

$$C_\pi = C_{BE} \Big|_{op}$$

$$C_\mu = C_{BC} \Big|_{op}$$

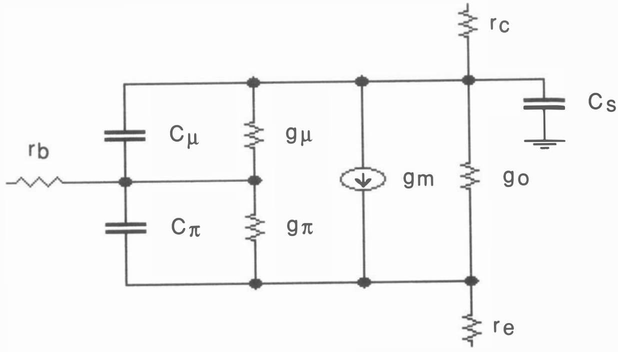
$$C_S = C_{sub}$$

$$g_\pi = \frac{I_B}{V_t} \Big|_{op}$$

$$g_m = \frac{I_C}{V_t} \Big|_{op}$$

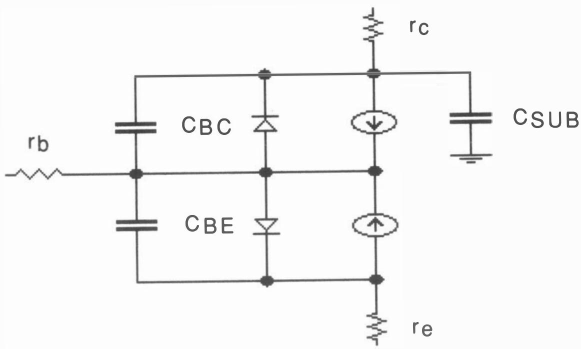
$$g_o = \frac{I_C}{V_A} \Big|_{op}$$

$$g_\mu = \frac{g_o g_\pi}{g_m} \Big|_{op}$$



Time domain frequency model

If the AC frequency solution does not remain in the linear region of operation for the transistor, then a point by point time domain solution must be performed to model the nonlinear behaviour. This method uses the complete Ebers-Moll model, with the capacitors in the model represented by their energy storage model derived with the trapezoid rule.



$$C_{BE} = \frac{\tau_F I_S}{V_t} \exp \frac{V_{BE}}{V_t} + C_{je0} \left(1 - \frac{V_{BE}}{\phi_e} \right)^{-me}$$

$$C_{BC} = \frac{\tau_R I_S}{V_t} \exp \frac{V_{BC}}{V_t} + C_{jc\,0} \left(1 - \frac{V_{BC}}{\phi_e} \right)^{-mc}$$

Necessary parameters (must be non-zero)

I_S : saturation current; typically 10^{-14} to 10^{-15} Amperes.

β_F : forward current gain coefficient; typically between 10 and 500.

β_R : reverse current gain coefficient; typically 1 to 5.

V_A : Early voltage; typically 50 to 100 V.

V_t : thermal voltage; assuming room temperature, fixed in program at 25 mV.

Optional parameters

r_b : base ohmic resistance; typically 1 Ω to 1 kΩ.

r_c : collector ohmic resistance; typically 10 Ω to 100 Ω.

r_e : emitter ohmic resistance; typically 1 Ω.

C_{sub} : substrate capacitance; typically 1 to 2 pF.

C_{je0}, C_{jc0} : zero bias junction capacitance; typically 0.1 to 10 pF.

f_e, f_c : junction potential, typically 0.5 to 0.7 Volts.

m_e, m_c : junction grading coefficient; typically 0.33 to 0.5.

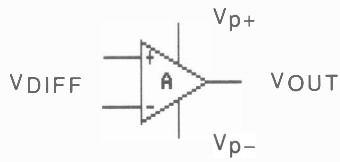
τ_F : forward transit time; 0.3 ns to 80 ps.

τ_R : reverse transit time; typically 1 to 20 ns.

Operational amplifier model

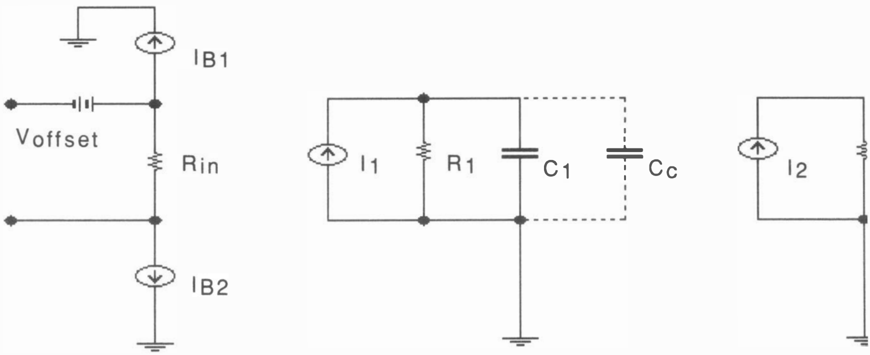
Characteristic equation

$V_{OUT} = AV_{DIFF}$



Model

The same model is used for AC, DC and time domain analyses.



$C_1 = 1/(2\pi\ R_1\ f_p)$

$f_p = \log^{-1}(\log f_u - (A_{OL}/20)\ )$

$R_1 = 1\ k\Omega.$

$I_{B1} = I_{BIAS} + I_{OFFSET}/2$

$I_{B2} = I_{BIAS} - I_{OFFSET}/2$

Parameters

A : open loop gain, distributed across stages.

R_i : input resistance.

R_o : output resistance.

V_{pm} : power supply voltage.

V_{SW} : maximum voltage swing.

V_{offset} : offset voltage.

I_{bias} : bias current.

I_{offset} : offset current.

f_u : unity bandwith, i.e. the bandwith that produces unity gain.

f_{p2} : a third stage may be introduced by specifying the location of a second pole; $R_2 = 1k\Omega$, $C_2 = 1/(2\pi\ R_2\ f_p)$

C_C : compensation capacitance; increasing this will shift the dominant pole to the left.

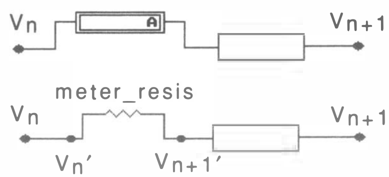
SR : slew rate, achieved by limiting current I_1.

Digital multimeter modes

Ammeter

The meter_resis, in series with the circuit, should be kept at a very low value, e.g. $1\mu\Omega$.

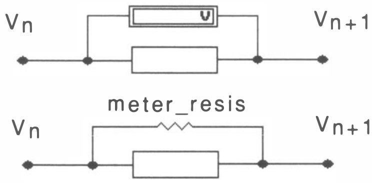
current = $(V_n' - V_{n+1}')$/meter_resis



Voltmeter

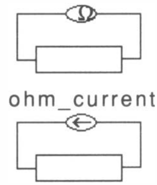
The meter_resis, in parallel with the circuit, should be kept at a maximum, e.g. 1 GΩ.

voltage = $V_n - V_{n+1}$



Ohmmeter

resistance = $(V_n - V_{n+1})$/ohm_current



Decibel

dB reading is:

20 x $\log_{10}$(volt reading/decibel standard)

where the decibel standard is set on the label option for the multimeter.

component parameter reference tables

This section contains tables of the parameters you can adjust in ELECTRONICS WORKBENCH. You can set the values of these parameters by choosing Model from the Label box for the component.

The first column of the tables shows the symbol used in the program and the second the generally accepted symbol for the parameter named in the third column. Some typical values may appear in the final column.

Transformer model parameters

n	n	turns ratio	n > 1 step down n < 1 step up
Le	L_e	leakage inductance	ideally zero
Lm	L_m	magnetizing inductance	ideally zero
Rp	R_p	primary winding resistance	ideally zero
Rs	R_s	secondary winding resistance	ideally zero

Diode model parameters

IS	I_S	saturation current	10^{-14}—10^{-15}A;cannot be 0
rd	R_S	ohmic resistance	1nΩ
Cj	C_{jo}	zero bias junction capacitance	0.1—10pf
P	ϕ_B	junction potential	0.5—0.7V
T	τ_t	transit time	1nS
m	m	junction grading coefficient	0.33—0.5

Bipolar junction transistor parameters

BF	β_F	forward current gain coefficient	10—500
BR	β_R	reverse current gain coefficient	1—5
IS	I_S	saturation current	10^{-14}—10^{-15}A
rb	r_b	base ohmic resistance	1Ω–1kΩ
re	r_e	emitter ohmic resistance	1Ω
rc	r_c	collector ohmic resistance	10Ω–100Ω
Cs	C_{sub}	collector substrate capacitance	1–2pF
Ce	C_{jeo}	zero bias b-e junction cap.	0.1–10pF
Cc	C_{jco}	zero bias b-c junction cap.	0.1–10pF
TF	τ_F	forward transit time	0.3ns–80ps
TR	τ_R	reverse transit time	1–20ns
Pe	ϕ_e	b-e junction potential	0.5–0.7V
Pc	ϕ_c	b-c junction potential	0.5-0.7V
me	m_e	b-e junction grading coefficient	0.33-0.5
mc	m_c	b-c junction grading coefficient	0.33-0.5
VA	V_A	Early voltage	50-100V

Operational amplifier model parameters

A	A	voltage gain	very large (ideally ∞)
Ri	R_i	input resistance	very large (ideally ∞)
Ro	R_O	output resistance	very small (ideally 0)
Vpm	V_{pm}	power supply voltage level	5—15V
Vsw	V_{sw}	maximum voltage swing	10 —30V
Vos	V_{offset}	offset voltage	
Ibs	I_{BIAS}	bias current	
Ios	I_{OFFSET}	offset current	
SR	SR	slew rate	
fu	f_u	unity bandwith.	
fp2	f_{p2}	second pole	
Cc	C_C	compensation capacitance	

Customizing Help

WARNING

This is a procedure for experienced computer users. You must be able to use a text editor that produces plain ASCII files with no formatting characters imbedded in them and you must understand the directory structure. Never work on the original help files. Be careful.

You have complete control over the system called by "Help" from the menu. The text file may be modified or created anew with any content and any cross-references you wish.

The system consists of entries with keywords which are linked to buttons (containing the same string as the keyword) in other entries in the text file. You can create the entries and cross-reference buttons easily with a text editor. The special format is explained below.

If you select an object and press F1, a help window about it will open. ELECTRONICS WORKBENCH uses the names of its objects to call help when F1 is pressed. The entry on the selected object is presented directly when F1 is pressed. By using the name of any of these objects as a keyword for an entry, that entry will be called up directly when the object is selected and F1 is pressed. If you want to write your own help for a component, just use the component's name in your entry's keyword list.

You could use the system in an entirely different way, to create interactive lessons. By presenting a menu as the index, you can introduce different topics. Any button on a screen can call any entry you wish. Tests or lessons using ELECTRONICS WORKBENCH could be set up this way.

names of components in ELECTRONICS WORKBENCH

The names of components and instruments are contained in the files EWBA.MSG and EWBD.MSG. The help system uses these names for keyword entry into the help files EWBA.HLP and EWBD.HLP. The order of the names and other messages in the file is important, but you can modify the text on any line if necessary. The format (line numbers, quotes) must be preserved. The names in EWB?.MSG and the keywords in the two EWB?.HLP files must agree or the help system will not work correctly.

file name and path

The information presented by the help system is contained in an ASCII file named EWBA.HLP or EWBD.HLP. ELECTRONICS WORKBENCH expects this file and a file with the same name and the extension ".TBL" to be in the current directory or in the directory specified on the command line with the "-s" parameter.

file format

The help files (EWB?.HLP) have the following format:

~keyword[| keyword...] | |

title

text...{button[: keyword]}...text.

The tilde (~) marks the beginning of an entry or subject in the help system. There must be at least one keyword here and the line must end with | | (two vertical bars). Additional keywords may be added, separated by single vertical bars. You can use these as synonyms to call the same entry from different words (singular and plural forms, for example). The synonyms may be longer than one line, but must end with "| |."

The next line is reserved for the title of the window. This must be a line by itself, no more than 50 characters long.

Any text following the title line will appear in the window. The width of the window will fit the longest line. If there are too many lines to fit on one screen, arrowheads on the top row allow you to page the text forward and back.

Within the text that is displayed on the screen, you can create buttons that will open a new window on another entry in the file. To make a button, mark a word or phrase (on a single line) with curly braces, like {this}. The button will show up as bold or highlighted text on the screen.

The string inside the braces must be a keyword for some entry in the file, or the keyword may optionally follow a colon (:). Otherwise you will get an error message when the button is used in the program. A keyword placed after a colon will not show on the screen. This makes it possible to have a short abbreviation as the keyword to refer to the entry and allows you to use many different buttons to call a single entry.

Using a single keyword for each entry and placing it after a colon keeps the index for the system smaller and consumes less memory.

If the colon is not present, the help system takes the entire bracketed phrase as the keyword for the index.

Example

This example from the help system produces the home screen and two others that are cross-referenced from it. You can compare it to its display in the program to see how the format works.

~home | |
Overview

ELECTRONICS WORKBENCH
(C) 1988, 1991 Interactive Image Technologies

An Electronic Digital/Analog Circuit
Design and Simulation Program

{ • What you can do with EWB:purp}
{ • Parts of the screen:scrn}
{ • Building a circuit:wksp}
{ • Simulating a circuit:go}
{ • File operations:F9}
{ • Program controls:ctrl}
{ • List of topics:tpx}

Point and click to get more.

~purp | |
Purpose
Electronics Workbench can be used to

• {construct:wksp} a schematic for an electronic
  circuit
• {simulate:go} the activity of that circuit
• display the circuit activity on simulated
  {test instruments:inst}
• and {print:prt} a copy of the schematic, the
  instrument readings and component list

~scrn | |
Appearance
The program models a workbench for electronics.

The large central area on the screen acts as a
{breadboard:wksp} for circuit assembly.

On the right side of this {workspace:wksp}
is a {bin of parts:bin}.

Above the workspace is a shelf of
{test instruments:inst} and {program controls:ctrl}.

You may wish to read about {common operations:ops} for the
{workspace:wksp}.

{List of topics:tpx}

installing in ELECTRONICS WORKBENCH

The file containing the help system should be in the current directory or in the directory specified on the command line with the "-s" parameter when ELECTRONIC WORKBENCH is started. This normally means the same directory as the executable program files. See the chapter on program setup and requirements for more detail.

ELECTRONICS WORKBENCH requires an index to use the help file. Create this index from the help file by executing the program EWBNDX.EXE. The current directory should contain the program and the text file; at the system prompt type:

```
ewbndx filename[enter]
```

where *filename* is any legal filename with the extension ". HLP".

Note that the extension ".HLP" must be part of the filename, but you should not type it on the command line with EWBNDX. You can use a full path in front of the filename.

EWBNDX will create a new file with the same filename and the extension ".TBL". You can create pairs of help files with any name and then rename them for use with ELECTRONICS WORKBENCH.

The help system in ELECTRONICS WORKBENCH will use the files named EWBA.HLP and EWBA.TBL for the Analog Module and EWBD.HLP and EWBD.TBL for the Digital Module.

To locate the system files, you can use the parameter "-s<pathname>" on the command line when you start EWBNDX.EXE. Otherwise it looks for some system files in its current directory. If you want to copy EWBNDX.EXE to another directory to work with various help files, use this parameter.

If you want to set up a space to work on custom help files, you might find it convenient to copy EWBNDX.EXE along with the .MSG and the .FNT files to a subdirectory. Then you can use EWBNDX with no command-line parameters.

TAKE NORMAL PRECAUTIONS TO BACK UP THE ORIGINAL FILE BEFORE MAKING ANY CHANGES TO THE HELP SYSTEM. WORK ONLY ON A COPY. BE CAREFUL.

Program setup and requirements

summary

Minimum system requirements

- IBM PC or AT-compatible microcomputer
- 640 Kilobytes memory
- hard disk—2 Megabytes for installation
- high resolution graphics adapter and display (EGA or VGA)
- Microsoft-compatible mouse
- MS-DOS or PC-DOS version 3.0 or later

ms-dos

The following information assumes that you are familiar with your hardware and basic MS-DOS operations and directory organization. If you are uncertain about your hardware or operating system, consult the documentation provided with it, its supplier, or its producer.

installing and running the Professional Version

Electronics Workbench provides state-of-the-art graphics. It requires a hard disk to run. Because high-resolution color graphics take time to display, we recommend using an IBM AT-compatible computer or better.

The program is supplied on four 360 Kilobyte, 5.25" disks or on two 720 Kilobyte, 3.5" disks.

To install the professional version of ELECTRONICS WORKBENCH on your hard disk you will need approximately 2 Megabytes of free space.

Make A: the current drive.

Insert Disk 1 in drive A: and type this command—

```
install[enter]
```

Now follow the instructions on the screen.

After ELECTRONICS WORKBENCH is successfully installed, run the analog simulation with this command—

```
analog[enter]
```

Run the digital simulation with this command—

```
digital[enter]
```

It is a good idea to collect circuits and libraries for both modules in subdirectories. The file selector box filters the filenames so you see only the correct ones. You must use MS-DOS commands to create subdirectories before running ELECTRONICS WORKBENCH.

displays

The professional version of ELECTRONICS WORKBENCH is intended to work with high-resolution EGA and VGA color displays. It will also support CGA, MCGA and Hercules video display adapters, but all pictures will be monochrome.

These display cards must be compatible at the hardware level with the IBM or Hercules standard. System compatibility with MS-DOS, but running on hardware that is not fully compatible, is not sufficient. ELECTRONICS WORKBENCH writes directly to the screen display memory in all modes.

See the documentation for your display card for more information.

If you need to force the program to use a particular graphics mode for some reason (using an LCD screen with an overhead projector, for example), you can start the program with the following command-line switch—

```
/display=<display>
```

where "<display>" is one of the following:

mvga	Multi-Color Graphics Array (640 x 480 monochrome)
cga	Color Graphics Adapter (640 x 200 monochrome)
ega	Enhanced Graphics Adapter (640 x 350)
hgc	Hercules Graphics Card (720 x 348 monochrome)
vga	Video Graphics Array (640 x 480)

In all cases, your graphics adapter card must be able to display the selected resolution or ELECTRONICS WORKBENCH will not operate. If you have problems using any mode other than that automatically selected by the program, consult your documentation or your dealer.

system files

ELECTRONICS WORKBENCH normally expects to find its system files in the same directory where the executable files (EWBA.EXE and EWBD.EXE) are found.

However, you can use an optional command-line parameter to enable ELECTRONICS WORKBENCH to find the system files in a different directory if you wish. Just start the program with—

```
ewba /s=<pathname>
```

or

```
ewbd /s=<pathname>
```

where “<pathname>” designates the directory where the files from the system disk are located.

You may wish to create a batch file with a line like this. Such an arrangement could be useful on a network.

ELECTRONICS WORKBENCH writes a temporary file to the hard disk when printing. This file will be written in the current directory, or in the directory indicated by an environmental variable named TEMP that can be controlled with the MS-DOS SET command. See your MS-DOS manual for more information about environmental variables and the SET command. If ELECTRONICS WORKBENCH is running on a network, its users must have write privileges in the system files directory in order to configure the printer.

You can add the home directory of ELECTRONICS WORKBENCH to your search path so you can start it wherever you are on a hard disk. You may wish to start the program while a directory containing circuit files is current, for example. To add a directory to your search path, see your MS-DOS manual for instructions on using the PATH command.

Note

The same bitmap images are used for all display standards, so the proportions of the images in the program will be slightly different on different displays. The displayed area of the workspace will differ on different displays, though the workspace has the same overall dimensions. Printing uses the same bitmaps as well. On dot-matrix printers, you may prefer the results if you choose portrait orientation from the printer configuration menu.

mouse

ELECTRONICS WORKBENCH requires the use of a Microsoft-compatible mouse. It operates with the Logitech Mouse, the Microsoft Mouse and the IBM PS/2 mouse.

Index

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes