The Fundamentals of Digital Semiconductor Testing

Guy Perry

Soft Test Inc.

246 Union Avenue

Los Gatos, California 95032

USA

Phone (408) 377-1888 Fax (408) 377-1866 Www.soft-test.com

Version 2.03 - September 1999 ISBN # 0-9658797-0-4

Preface

This manual has been created to provide a detailed understanding of the concepts of digital semiconductor testing of CMOS and TTL technologies using ATE test equipment. A brief overview of semiconductor manufacturing and the peripheral equipment used in production testing is also included.

The text presents procedures and guidelines for engineers to follow which will insure quality and reliability when creating or modifying test programs. Techniques are also included which will increase productivity and reduce the need for test system time.

To get the most out of this material the reader must have an understanding of basic electronics and digital circuits. If you need help in this area a short refresher can be found in Section One. The training class associated with this text presents a forum for questions, concerns, ideas and general discussions related to digital semiconductor testing.

I truly hope this information will help you to understand digital component testing and perhaps make your job a little easier.

Sincerely, Guy Perry

Table of Contents

Chapter 1 The Basics

Objectives

This section explains:

* Number formats
* What is: Voltage, Current, Resistance
* Various methods to measure resistance
* Digital Logic Pulses
* Simple Digital Circuits

This section is intended to be a short refresher for those who have not used their electronics training recently. It is not, however, intended to be a replacement for a complete basic electronics course.

Scientific / Engineering Notation

It may be necessary to express very large numbers or very small numbers when dealing with test issues. To simplify writing these numbers they are often expressed in scientific notation format. This format uses the mathematical properties of the powers of 10. In engineering texts and writing, abbreviations are used to represent levels of 1000. The following table shows the scientific notation, the computer notation, the abbreviation and names for commonly used numbers in the test industry.

Number	Name	Computer Notation	Power of 10 Equivalent	Abbreviation
1,000,000,000	Giga	10E+9	$1x10^{9}$	G
1,000,000	Mega	1E+6	$1x10^{6}$	M
1,000	Kilo	1E+3	$1x10^{3}$	K
10	Deka	1E+1	$1x10^{1}$	da
1	Unity	1E+0	$1x10^{0}$	
.1	Deci	lE-1	$1x10^{-1}$	d
.001	Milli	lE-3	$1x10^{-3}$	m
.000001	Micro	lE-6	$1x10^{-6}$	µ(u)
.000000001	Nano	lE-9	$1x10^{-9}$	n

Voltage

Voltage is the electrical difference of potential between two points. In a power supply the negative terminal has an accumulation of negative charge (a surplus of electrons), while the positive terminal has a positive charge (an accumulation of positive ions). This potential difference can be measured as voltage. "V" is the unit abbreviation for voltage, as in 5.0V (named for Alessandro Volta, an Italian physicist who first measured it). You sometimes create a voltage, called static electricity, when you walk across a carpet in a room with dry air. The formula symbol for voltage is E, short for EMF, short for ElectroMotive Force (which is probably what Volta called it). Some texts also use "V" as the formula symbol for voltage.

Current

Current is the flow of charge through a conductor. If a conductor (e.g., copper wire) is connected between the positive and negative terminals of a power supply, current will flow in an effort to balance the potential difference (the voltage) between the two points. This flow can be measured and is described in units called amperes or amps (named for Andre M. Ampere, a French physicist and contemporary of Volta who studied electricity). "A" is the unit abbreviation for current, as in 2.5mA (milliAmps). "I" is the formula symbol for current.

Resistance

Resistance is the opposing force to current flow, similar to friction which opposes mechanical force. The unit of measurement used to describe the amount of resistance a material offers is the ohm, (named after George Simon Ohm, a German physicist who did research on electricity). The unit abbreviation for ohm is the Greek letter omega (Q). The formula symbol for resistance is R, for the obvious reason.
Different materials have different amounts of resistance. The table below shows the typical resistance found in one foot of

Material	**ohms at 25C**
Silver	9.90
Copper	10.37
Gold	14.70
Aluminum	17.00
Nickel	47.00
Iron	74.00
Carbon	21,000.00

Note: the resistance of most materials changes with temperature. Some materials become more resistive at higher temperatures, called a Positive Temperature Coefficient of resistance (positive TC) and some become less resistive higher temperatures (negative TC).

Using Ohm's Law to Test Device Specifications

Ohm's law defines the relationship between current, voltage and resistance as:

	I = amperes	A
E = I * R where	E = volts	V
	R = resistance	Ω

This indicates that voltage is equal to the current multiplied by the resistance. When voltage and resistance are known, current can be found using the formula:
I = E / R
When current and resistance are known, voltage can be found using the formula:

E = I * R

When current and voltage are known, resistance can be found using the formula:

R = E / I

A test system is capable of supplying voltage and measuring current or supplying current and measuring voltage, therefore many tests are performed to determine the resistance of the device being tested. It may seem odd but the results of tests are not normally measured directly in ohms. The value of a resistor can be found by applying a voltage across the resistor and measuring the current flow through it. Once the voltage and current are known the resistance can be calculated using Ohm's law.

Measuring Resistance

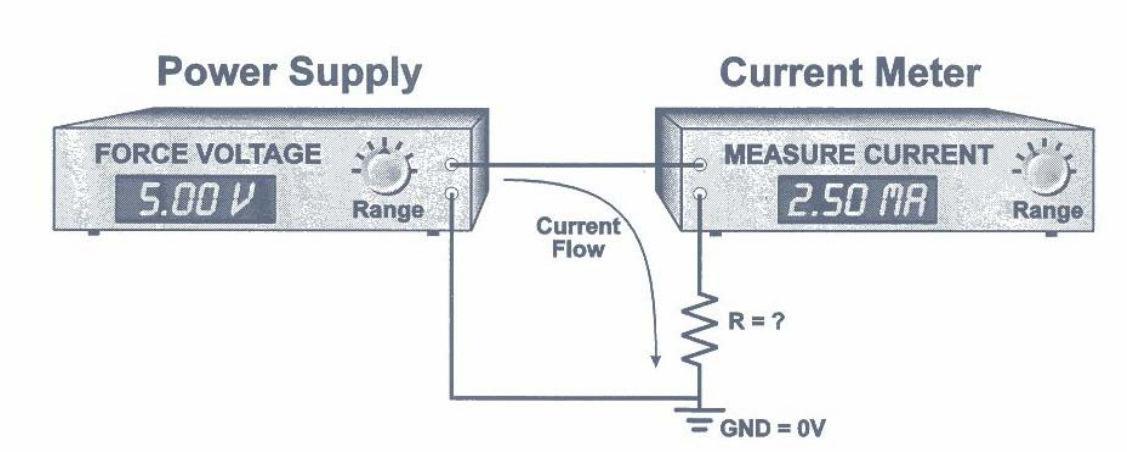


Figure 1-1

Resistance is measured by forcing a known voltage (E) and measuring the resultant current (I), then calculating R - E /1. Note that the resistance of the current meter must be very small compared to the resistance being measured.

In the example above, the test system's power supply provides 5.0V. The current meter measures the amount of current flow through the resistor and reports its findings. Let's say the current meter indicates a current flow of 2.5 x 10-3 amps (2.5mA). The resistance can be found using the following formula:

R = E / I

R = 5 / 2.5mA

R = 2000Ω

This works out to 2000, so the unknown resistor must be 2000Ω

Using Current Limits to Verify Resistance

Semiconductors are designed with certain specifications in mind, so rather than finding the value of an unknown resistor, a test is made to insure that the resistor value is within the design specification range. If the design specification states that the resistor value must be 2000Ω± 10%, the test can be made as follows:

The smallest acceptable value of resistance is 1800Ω(2000 - 2000 * 10%). When 5.0V is applied across an 1800Ωresistor, 2.77mA of current will flow. This can be determined using the following formula:

I - E / R

I = 5 / 1800

I = 2.77mA

The largest acceptable value of resistance is 2,200Ω(2000 + 2000 * 10%). When 5.0V is applied across a 2,200Ωresistor 2.27mA of current will flow.

To perform the test, an upper current limit of 2.77mA and a lower limit of 2.27mA is set or programmed into the test system. When the test is performed, the current measured by the current meter is compared to the programmed limits and a pass/fail decision is made.

Using Voltage Limits to Verify Resistance

In the example shown in Figure 1-1 a voltage was applied across the resistor and the resulting current was measured. The test can also be performed by forcing a known current through the resistor and measuring the resultant voltage as seen in Figure 1-2. If the resistor is exactly 2000Ωand a current of 2.5mA is forced through the resistor, a voltage drop equal to 5.0V will appear across the resistor.

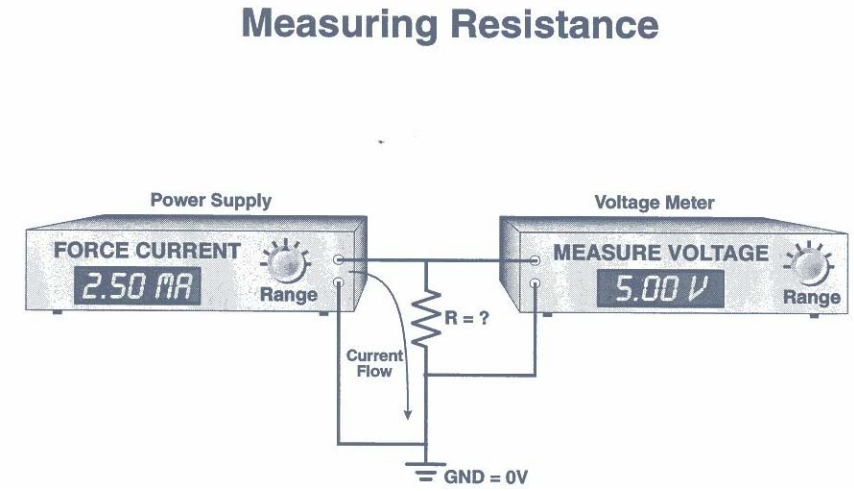


Figure 1-2

Resistance is measured by forcing a known current (I) and measuring the resultant voltage (E), then calculating R = E /I. Note that the resistance of the voltmeter must be very high compared to the resistor being measured.

In order to perform the test using this method the voltage limits must be defined. The lowest acceptable value of resistance, 1800Ωwill produce a voltage drop of 4.5V from this formula;

$E = I * R$
$E = 2.5x10^{-3} * 1800$
$E = 4.5$

The highest value of resistance, 2200Ω, will product a voltage drop of 5.5V. The test can be performed by forcing 2.5mA of current and comparing the measured voltage against the upper and lower voltage limits. If the measured value is within the limits the result of the test is pass otherwise it is fail.

Since most semiconductor devices operate between 3V and 5V, the value of current forced can be adjusted to yield a voltage within this range. For example, if the expected resistance value is 5000Ω(or 5KΩ) a current of 1mA can be forced through the resistor to produce a voltage drop of 5V across it.

Digital Numbers

Most computers in use today are digital computers. Digital refers to the way computers perform their various functions or operations by manipulating digits or numbers. Humans using the Arabic number system also began by manipulating digits, called counting on their fingers.

Base 10

We have a number system which uses the values 0,1,2, 3, 4, 5, 6, 7, 8, 9, 0 to represent no fingers showing up to 10 fingers showing. At 10, a new place of importance is added and the lowest place values start over. This is called a number system in base 10. If we had been born with 12 fingers, our number system would have been based on 12 rather than 10. (Dealing with the 360 degrees in a circle would be more natural for us if we used a base 12 number system.) The only thing which determines the number system we use is how many distinct values we use before adding a new place and starting the lowest place values over.

In the decimal system (base 10 where decem is Latin for 10 and December was the tenth Roman month), the next place value is a multiplier of a power of 10. For example, the number 24 represents

$(2 \times 10^1) + (4 \times 10^0) = 24_{10}$

and the number 100 represents

$(1 \times 10^2) + (0 \times 10^1) + (0 \times 10^\circ) = 100_{10}$

This numbering system gives us a very concise way to represent a large range of numbers. Compare this to the Roman Numeral system where 1996 is represented as MCMIVC or MCMLXXXXVI. (The Roman Numeral system has a more fundamental problem—there is no representation for no fingers showing, i.e. no zero.)

Base 2

We use 2 hands with 5 fingers each to give us 10 fingers for the decimal system. Think of a binary system (bi meaning 2 as in a bicycle with 2 wheels) as having hands with only one finger. We can represent this system with values 0, 1,10 where we start over after the 2nd digit, in effect adding another one-fingered hand to represent the next place. With this base 2 number system, each value place represents a power of 2 (rather than a power of 10 as in the decimal system). Notice that the only available values are combinations of 0 and 1. For example, the number 11 represents (when translated to base 10)

$(1 \times 2^1) + (1 \times 2^0) = 3_{10}$

and the number 101011 represents

$(1 \times 2^5) + (0 \times 2^4)$ -f $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 43_{10}$

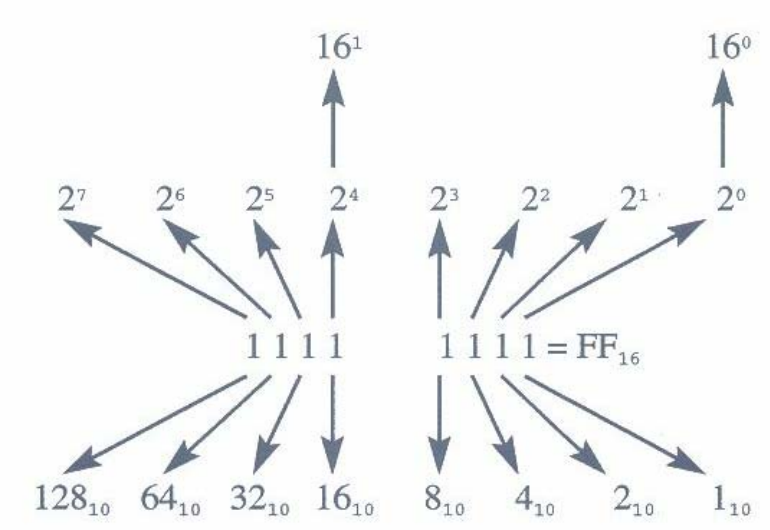
Note that the numbers 11_2 and 101011_2 are valid base 2 numbers in their own right. We translate them to decimal equivalents here only to show you what they represent in your most familiar number system.
It is often said that computers process ones and zeros by representing numbers as binary digits. A single binary digit is called a bit. All numbers are represented as binary values inside a digital computer because it is easier to design circuitry which uses one finger than circuitry which uses ten.

Base 16

Have you noticed that it takes a long string of Is and 0s to represent a number that is fairly short in decimal? Because of this, a shorthand way of representing binary numbers was developed, called Hexadecimal or hex (with hexa meaning 6, hexadecimal means 6 + 10, or 16). It is, in fact, a number system based on 16 values, but since $16 = 2^4$, it allows for a shorthand binary representation where 1 hex digit represents 4 binary digits (4 bits). Since we only have the number symbols 0 through 9, we use letters A, B, C, D, E, F for the other six symbols.

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
255	1111 1111	FF

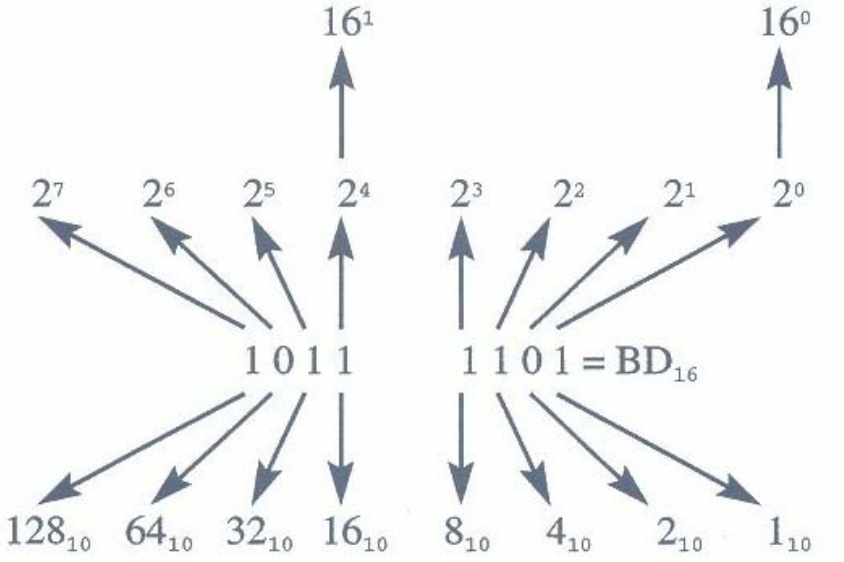
Notice that hex F = decimal 15 and hex FF represents decimal 255. As with base 10, the least significant value is represented by the right most digit. Significant digits increase in value by a power of the base for each place (10,2 and 16 respectively in the above table):



The hex number FF_{16} contains the equivalent of 8 bits. The easiest way to find the decimal value of a hex number is to first write the hex number as binary, then convert using powers of 2. Referring to the table of hex values, convert each hex digit into its equivalent four bits:

B_{16} = 1011 and D_{16} = 1101 so BD_{16} = $(1011\ \ 1101)_2$

Now convert this binary number into decimal using multiplication by powers of 2:



$(1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 189_{10}$

OR

$128+0 + 32+16 + 8+4+0+1 = 189_{10}$

Note: Remember that anything raised to the zero power =1.

Digital Logic

Two popular technologies are used to produce integrated circuits (ICs)—TTL (transistor-transistor logic) and CMOS (complementary metal-oxide semiconductor). In the past, TTL was faster than CMOS but used more power. (Power

dissipation of CMOS is directly related to the frequency at which it operates—the higher the frequency, the more power is used. TTL has a more constant power dissipation.) CMOS has now evolved to be the logic of choice for most new circuits because it can pack more circuitry into a given chip area.

The majority of a computer's circuitry is comprised of digital logic. This circuitry makes logical decisions which result in a yes/no or true/false answer. This shows why using binary arithmetic is so handy *in* a computer—a bit represents 2 states, 0 and 1, which can also represent yes/no and true/false. Simple digital circuits or logic blocks can be combined to form very complex logic functions. Examples of basic logic circuits are inverters (NOT), AND, OR, NAND and NOR gates and flip-flops (which are clocked gates).

Voltage Values Represent Logic Levels

Logic circuits read and write data based on logic levels; valid logic levels (zero and one) are determined by voltage values.

When input data is read into a circuit, a logic 0 is sensed if the voltage level is 0.8V or less and a logic 1 is sensed if the voltage level is 2.0V or greater (typical TTL values). Voltages between 0.8V and 2.0V do not represent valid input levels.

Logic Pulse

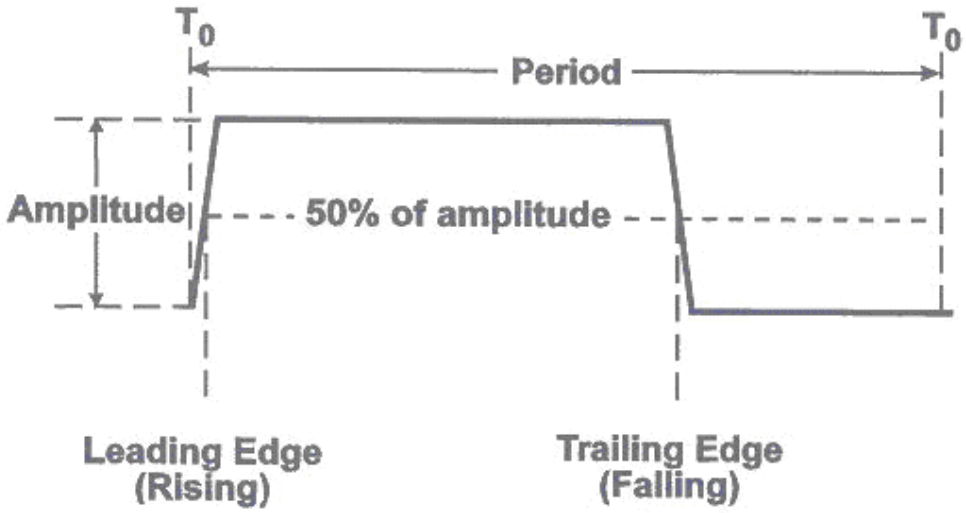
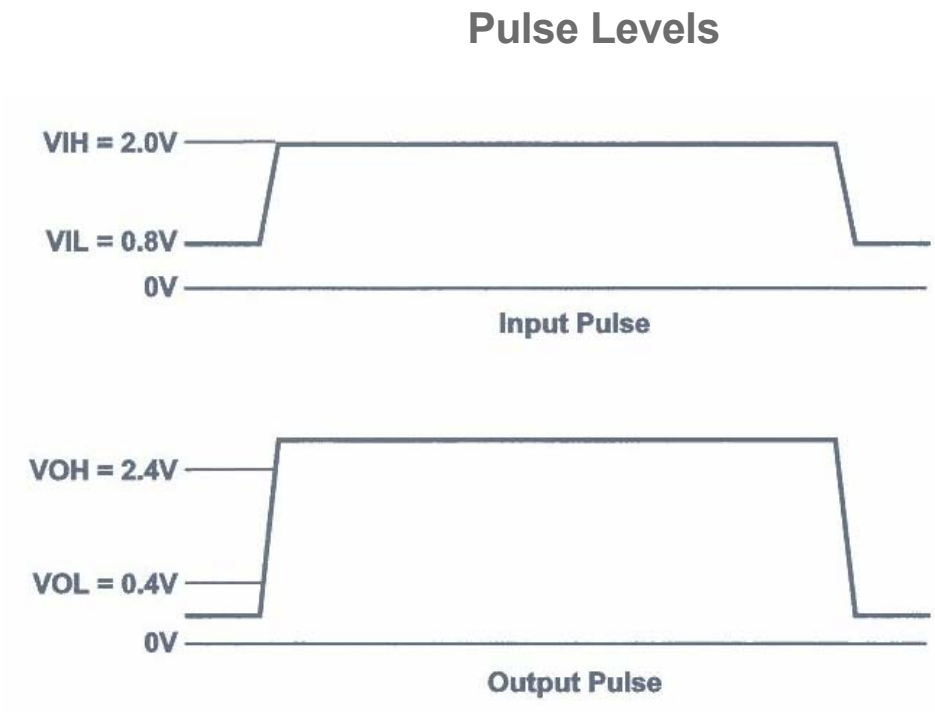


Figure 1-3

Period = 1 / Frequency
If Frequency = 10MHz then Period = 100nsec
The period is also known as the *cycle.*

When output data is produced by a circuit, a logic 1 is a voltage of 2.4V or greater and a logic 0 is a voltage of 0.4V or less (typical TTL values). Voltages between 0.4V and 2.4V do not represent valid output levels; a circuit receiving an invalid level will not know if it is receiving a 0 or a 1.

With an output low guaranteed to be less than 0.4V and an input which recognizes anything up to 0.8V as a low, there is a 0.4V noise margin (0.8 - 0.4). This allows up 0.4V of noise to occur on the signal and still have it correctly recognized as a logic 0. The same noise margin exists with logic 1 levels—2.4V output level and 2.0V input level.



Levels represent normal min/max TTL specification values.

Figure 1-4

A logic 0 input level is referred to as VIL (Voltage In Low). A logic 1 input level is referred to as VIH, (Voltage In High). A logic 0 output level is referred to as VOL (Voltage Out Low). A logic 1 output level is referred to as VOH (Voltage Out High).

Inverter

The inverter is the most basic of logic gates. With one input and one output, it changes a logic 1 to a logic 0 and vice versa. Inverters have a single input and a single output. An inverter is shown in Figure 1-5, with its input and output logic shown as a waveform in Figure 1-6.

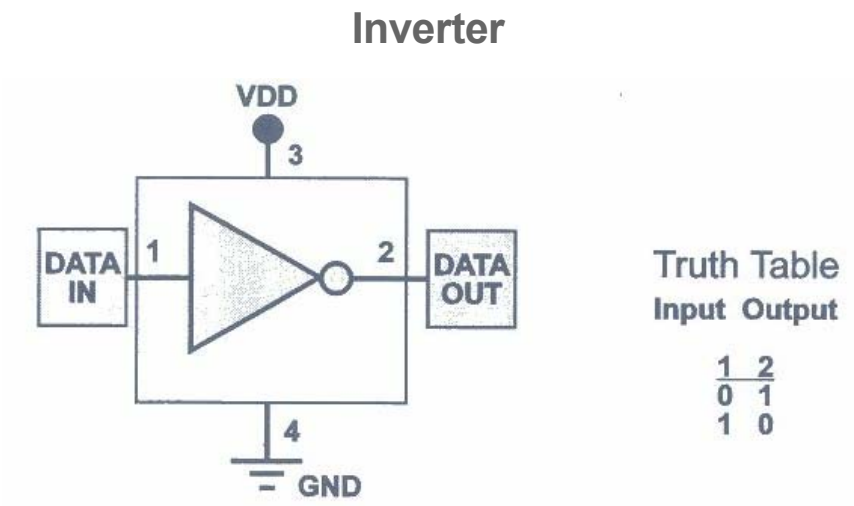


Figure 1-5

Logic 0 In = Logic 1 out

Logic 1 In = Logic 0 out

For VDD = 5.0V:

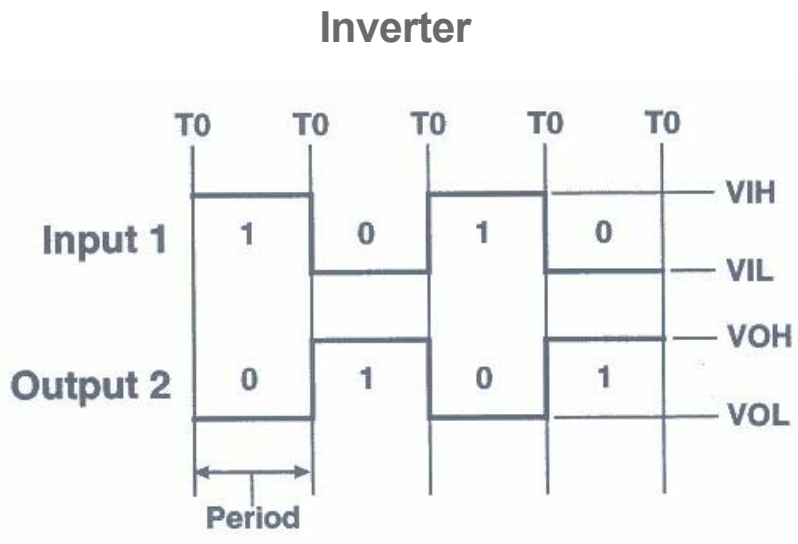
Any input voltage between 0.0V and 0.8V is a logic 0.

Any input voltage between 2.0V and VDD is a logic 1.

Any output voltage between 0.0V and 0.4V is a logic 0.

Any output voltage between 2.4V and VDD is a logic 1.

Figure 1-6 shows the waveforms for four test cycles. Each cycle begins at "T0" (time zero). In the first cycle, a logic one is applied to the input and a logic zero is expected on the output. In the second cycle a logic zero is applied to the input and a logic one is expected on the output. The pattern then repeats. The test period is the time duration of one cycle. VIL and VIH represent the voltage levels that will be applied to the input pin. VOL and VOH represent the expected output levels ◦



Functional Diagram of Inverter

Truth Table

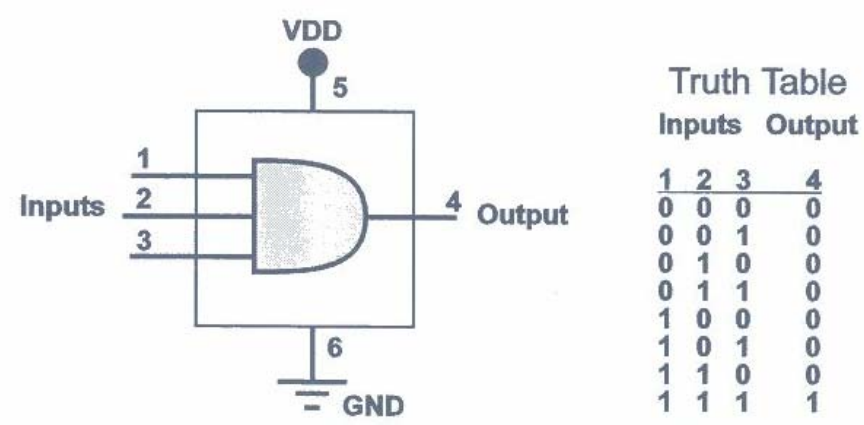
Figure 1-6

AND Gate

The AND gate provides a function in which the output value depends on the combination of all input values as given by the statement 'The output is a logic 1 only if ALL inputs are logic 1; the output is a logic 0 if ANY input is logic 0." AND gates have 2 or more inputs and one output. See Figure 1-7.

Note: Logic consisting of only basic gates such as AND, OR, etc. is sometimes called combinational logic.

AND Gate



Logic 0 on ANY input = Logic 0 out

Logic 1 on ALL inputs = Logic 1 out

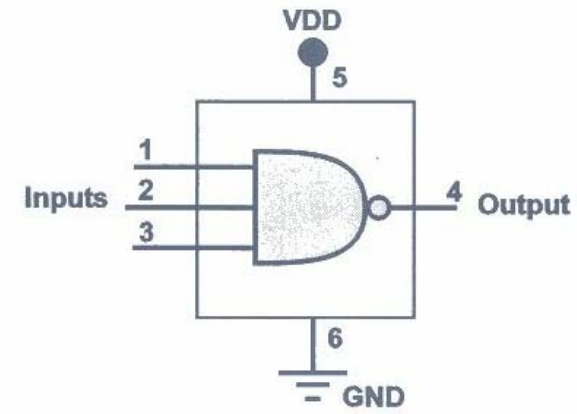
Figure 1-7

NAND Gate

The NAND gate provides a function in which the output value depends on the combination of all input values as given by the statement "The output is a logic 0 only if ALL inputs are logic 1; the output is a logic 1 if ANY input is a logic 0." NAND gates have 2 or more inputs and one output. See Figure 1-8.

This logic function is equivalent to an AND gate followed by an inverter. It exists because it is generally easier to build a semiconductor NAND gate than an AND gate and because it is often used to decrease total gate count when optimizing logic circuits.

NAND Gate



Truth Table

Inputs			Output
1	2	3	4
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

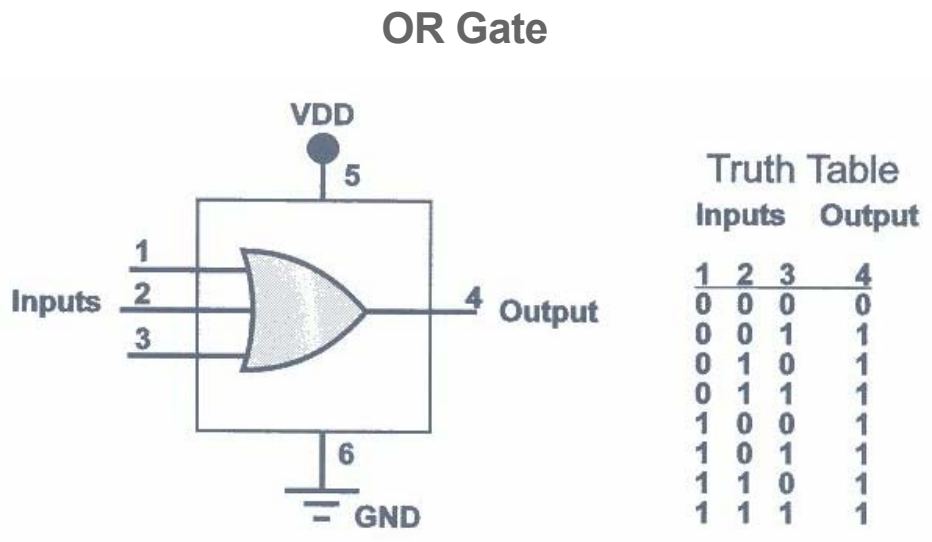
Logic 0 on ANY input = Logic 1 out

Logic 1 on ALL inputs = Logic 0 out

Figure 1-8

OR Gate

The OR gate provides a function in which the output value depends on the combination of all input values as given by the statement "The output is a logic 0 only if ALL inputs are logic 0; the output is a logic 1 if ANY input is a logic 1." OR gates have 2 or more inputs and one output. See Figure 1-9.



Logic 0 on ALL inputs = Logic 0 out

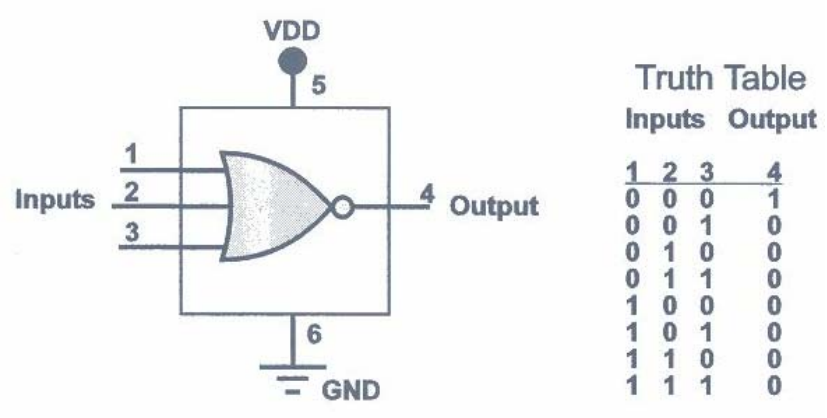
Logic 1 on ANY input = Logic 1 out

Figure 1-9

NOR Gate

The NOR gate provides a function in which the output value depends on the combination of all input values as given by the statement "The output is a logic 1 only if ALL inputs are logic 0; the output is a logic 0 if ANY input is logic 1." NOR gates have 2 or more inputs and one output. See Figure 1-10

NOR Gate



Logic 0 on ALL inputs = Logic 1 out

Logic 1 on ANY input = Logic 0 out

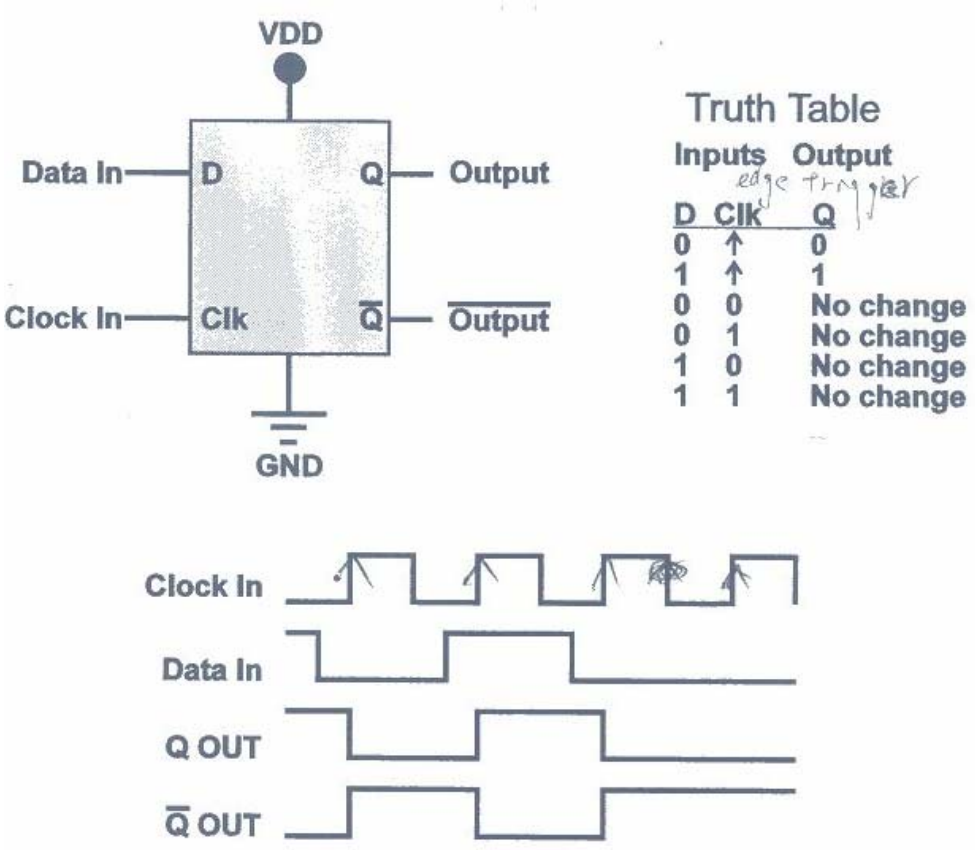
Figure 1-10

This logic function is equivalent to an OR gate followed by an inverter. It exists because it is generally easier to build a semiconductor NOR gate than an OR gate and because it is often used to decrease total gate count when optimizing logic circuits.

D Flip Flop

Flip flops are so named because the output changes (flips and flops) when a clock signal occurs. There are both level triggered and edge triggered flip flops. The most common and most useful is edge triggered, which means that its output only changes when a clock edge, or transition, occurs. The example in Figure 1-11 shows a D flip flop which is triggered by the positive going edge of the input clock signal. Flip flops have one or more data inputs, one clock input and one output. They may also have a Preset and/or Clear input and an inverted (Q/) output

D Flip-Flop



A positive clock edge causes the Data In

logic level to be latched at the Q output.

Figure 1-11

On the D type flip flop, the logic level which gets clocked to the output is the level which is present on the D input when the positive clock edge occurs. You can think of "D" as meaning Duplicate because the output duplicates the input after a clock edge triggers it. The D flip flop is the essence of a 1 bit memory, since whatever gets clocked to the Q output is remembered until the next clock occurs, no matter what logic level occurs on the D input in the meantime.

Other types of flip flops (not discussed here) are the T and the J-K

Chapter 2 Overview of Semiconductors and ATE

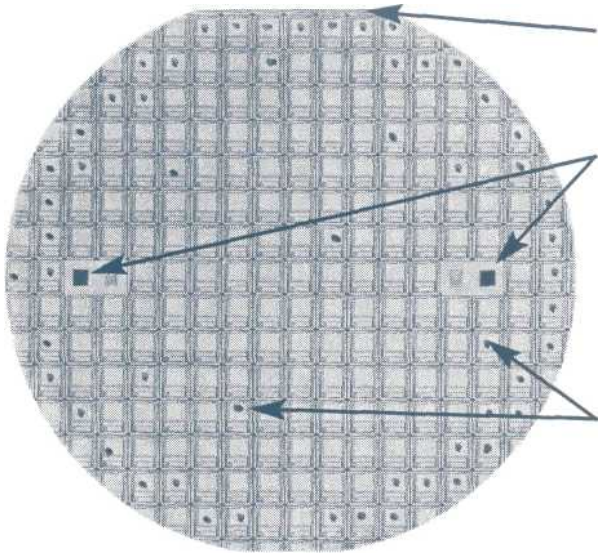
Objectives

This section explains:

- Wafers, Dice and Packages
- Basic concepts of Automatic Test Equipment
- Overview of Analog, Digital and Memory test systems
- Overview of Loadboards, Probers, Handlers and Temperature units

Wafers, Dice and Packages

The birth of the transistor in 1947 represents the start of the semiconductor industry. Since then, semiconductor manufacturing and fabrication techniques have advanced significantly. Many individual transistors can now be fabricated and interconnected to form complex "integrated circuits." Semiconductors called VLSI (Very Large Scale Integration) circuits, often containing millions of transistors, are presently being manufactured.



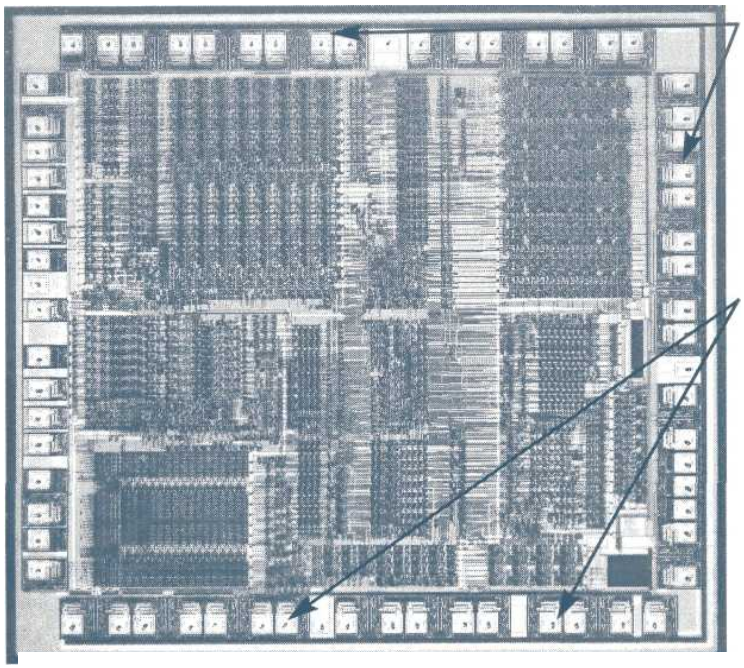
A wafer has a flat spot or notch (top of wafer) during the fabrication and testing process.

Wafers often have process monitor dice which are the same on all wafers regardless of the product. Because these process monitor dice are the same on all wafers, their electrical characteristics are known and checked at specific points during the fabrication process to verify that the process is being performed correctly.

Ink dots mark bad dice.

Figure 2-1 A wafer after it has been tested and inked
photo courtesy International Microelectronic Products, Inc.

Semiconductor circuits are initially manufactured in what is called wafer form. A wafer is a circular slice of silicon used as a foundation upon which many individual circuits are built. An individual circuit within a wafer is called a die, with dice being the plural form of the word. Each die is isolated from, and completely independent of, all other dice contained within the wafer.



Bond Pads

The light colored rectangles around the periphery of this die are bond pads. Probes make contact with the bond pads during wafer test. During the assembly process, bond wires are used to make electrical contact between a die and the pins of a package.

Probe Marks

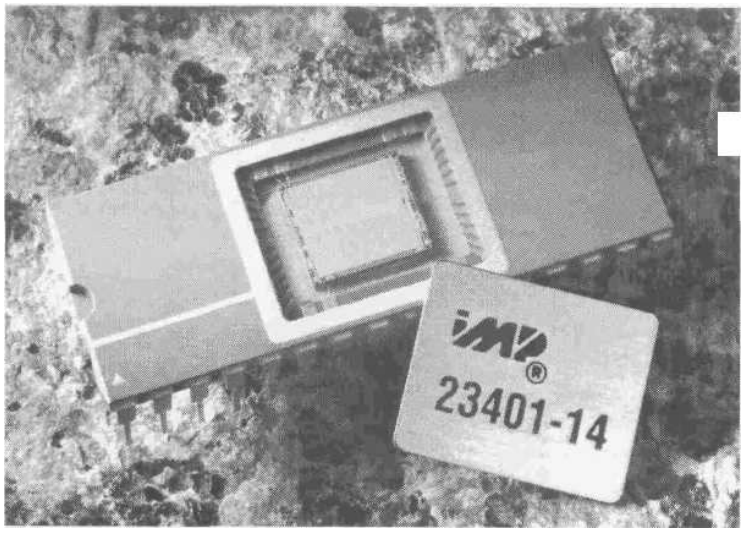
Notice the small dark spot on each bond pad. These are probe marks. The bond pads are made of soft aluminum, so when a probe makes electrical contact with the aluminum pad a small puncture, called a probe mark, is left behind. Probe marks can be used by the probe operator as visual verification that contact is being made to all die pads.

Figure 2-2 A single die sawed from a wafer

photo courtesy International Microelectronic Products, Inc.

When the manufacturing process is complete, each die must be thoroughly tested. Testing a wafer is called wafer probing or die sort. During this process, each die is tested to insure that it properly meets its device performance specification. This involves verifying voltages, currents, timings and functionality. When a die does not meet its specification, it is marked to indicate that it has failed the test process. Failures are typically indicated by placing an ink dot on defective dice.

After all dice on the wafer are probed, the wafer is cut to separate the dice. This is known as sawing the wafer. Any defective dice shown by an ink dot are scrapped (thrown away). Figure 2-2 shows an uninked die that has been sawed from a wafer. This die is now ready to be assembled into a package.



This picture shows a die mounted in a CERDIP package (Ceramic Dual Inline Package). The die is located in the center of the package. This area is called the die cavity.

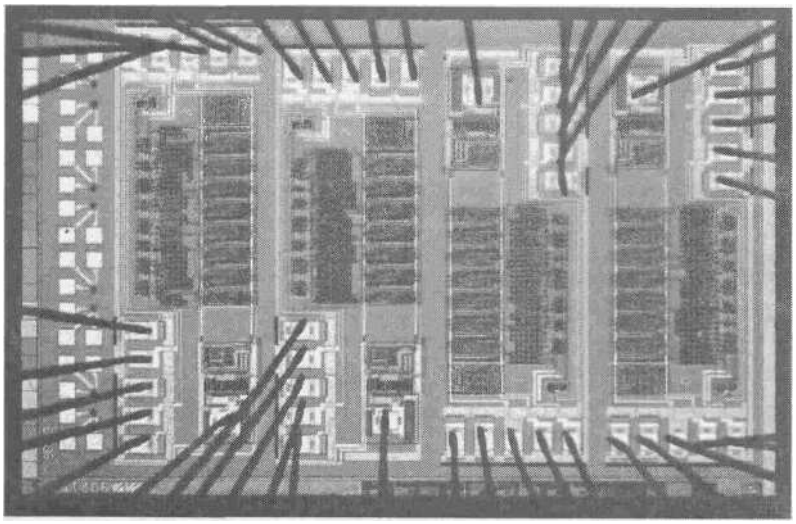
Notice the small bond wires which make the electrical connection between the bond pads of the die and the pins of the package.

The metal lid is used to seal the die inside the package.

Figure 2-3 Tested die mounted In a package

Photo courtesy International Microelectronic Products, Inc.

Once a die is assembled into a package, it is tested again in the process called package test or final test. If the probe card or wafer test equipment limits the speed or performance of the circuit, the package test may need to be more rigorous than the wafer test. The packaged device may be tested several times at various temperatures to guarantee parameters which are temperature sensitive. Commercial devices may be tested at 0° C, 25° C and again at 70° C. Military devices may be tested at -55° C, 25° C and 125° C



The black lines seen in this photo are bond wires. Bond wires make electrical connection between the bond pads of the die and the pins of a package.

The probes on the probe card make contact with the bond pads during wafer test. When viewed through a microscope during wafer test the probes appear similar to the bond wires in this picture

Figure 2-4 A die with bond wires attached to each bond pad

photo courtesy International Microelectronic Products, Inc.

Figure 2-4 shows a close-up view of a die and bond wires in the die cavity of a device package.

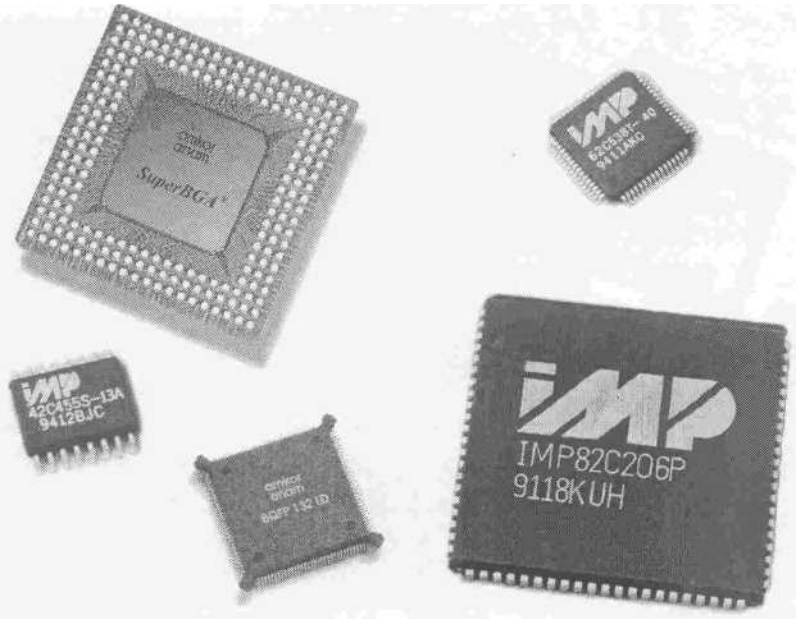


Figure 2-5 Several types of packages

photos courtesy International Microelectronic Products, Inc. and Amkor Electronics Inc.

Device packages are available in many different styles, figure 2-5 shows a few examples. Some common package types are listed below.

DIP	Dual Inline Package (dual indicates the package has pins on two sides)
CerDip	Ceramic Dual Inline Package
PDIP	Plastic Dual Inline Package
PGA	Pin Grid Array
BGA	Ball Grid Array
SOP	Small Outline Package
TSOP	Thin Small Outline Package
TSSOP	Thin Shrink Small Outline Package (this one is really getting small!)
SIP	Single Inline Package
SIMM	Single Inline Memory Modules (like the memory inside of a computer)
QFP	Quad Flat Pack (quad indicates the package has pins on four sides)
TQFP	Thin version of the QFP
MQFP	Metric Quad Rat Pack
MCM	Multi Chip Modules (packages with more than 1 die (formerly called hybrids)

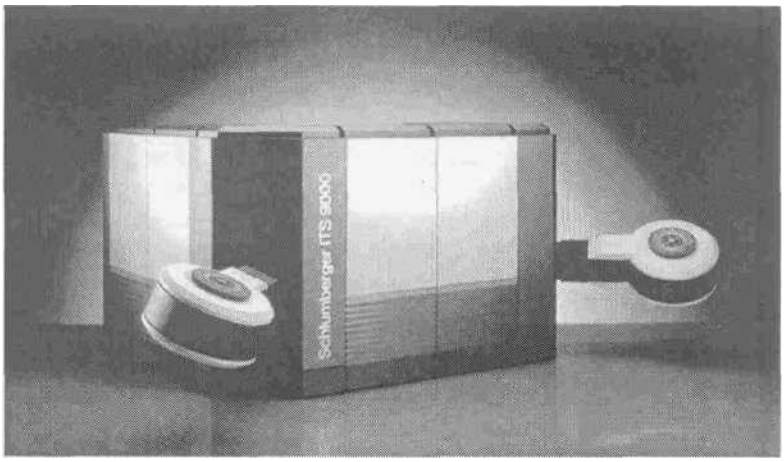


Figure 2-6 A digital test system with two test heads

Photo courtesy Schlumberger Technologies Inc.

ATE - Automated Test Equipment

As the complexity of integrated circuits has grown, the complexity of testing them has also grown. For some devices, the largest portion of manufacturing cost is testing. VLSI devices can require hundreds of voltage, current and timing tests. Millions of functional steps may be required to insure complete functionality of a VLSI device. To perform such complex testing, Automated Test Equipment (ATE) is used.

ATE is a collection of high performance computer controlled test instruments. A test system is the result of this merging of test instrumentation with a computer. The computer controls the test hardware by executing a set of instructions called the test program. Test systems typically produce very consistent test results which can be repeated quickly and reliably. To keep results correct and consistent, test systems are periodically calibrated, to verify (and adjust if necessary) the accuracy of forcing and measuring instruments.

When a test system is used to verify correct operation of an individual die within a wafer, a probe card physically and electrically connects the test system to the die. An interface circuit board called a load board or performance board connects the probe card to the instruments in the test system. The load board and probe card work together to enable electrical signals to pass back and forth between the test system and the die.

After dice are assembled into packages, they must be tested again. Testing packaged circuits may be done by manually inserting individual circuits into a device socket on the loadboard. This is called hand test. A faster way to test packaged circuits uses an automatic device handler. A handler has a set of contacts which connect the pins of the package to the loadboard. This provides a complete electrical path between the tester electronics and the die located inside the package. A handler can quickly pick up an untested device, insert it into the test site, then remove the tested device and place it in a particular bin location based on the pass/fail test results.

Semiconductor Technologies

Various methods are used to fabricate and manufacture digital semiconductor circuits. These methods are known as technologies. Some common technologies are: TTL (Transistor-Transistor Logic a.k.a. bipolar logic), ECL (Emitter Coupled Logic), SOS (Silicon on Sapphire), and CMOS (Complimentary Metal-Oxide Semiconductor). Although all technologies require testing, this text focuses on digital TTL and CMOS circuits

Digital and Analog Circuits

In the past there was a clear distinction between analog and digital circuit designs. Digital circuits manipulate electrical signals represented by zero and one logic levels. A zero is defined as a specific amount of voltage and a one is defined as another specific but different amount of voltage. All valid digital data is represented as either a zero or a one. A single one or zero level represents a bit of data. Any number can be represented by assigning weighted values to each data bit within a sequence of zeros and ones. The larger or more precise the value, the more data bits are required to represent the value. A byte is a group of eight bits. Digital values are often processed in bytes.

Unlike two level (zero/one) digital data, analog signals are continuous—an infinite number of values exists between any 2 signal levels. Analog circuits may use either voltage or current to represent data values. The

To help understand the fundamental difference between analog and digital values, think of a clock. The hands on an analog clock move continuously, therefore any and all time values can be determined and the precision of the reading is based to some degree upon the observer.

On a digital clock only incremental values are displayed. Any values of time which are less than the smallest incremental value (e.g. one minute or one second) are not displayed. When more precision is required of a digital clock, more digits are added, with each new digit representing a smaller incremental time value.

Some circuits contain both digital and analog circuitry. An A to D converter changes analog values to digital values; a D to A converter changes digital values to analog. Devices that contain both digital and analog circuitry are known as mixed signal devices. Another way of describing mixed circuits is based upon how much of the circuit is analog and how much is digital. The phrase Big D Little A indicates that the circuit consists primarily of digital circuitry. The phrase Big A Little D indicates that the circuit consists primarily of analog circuitry.

Types of ATE Systems

Although some ATE is considered to be general purpose, most test systems are designed to test a particular class or family of integrated circuits. Some of the more common classes of devices are memories, analog, mixed signal and digital , Each of these classes can also be subdivided, but for this discussion only four classes are considered.

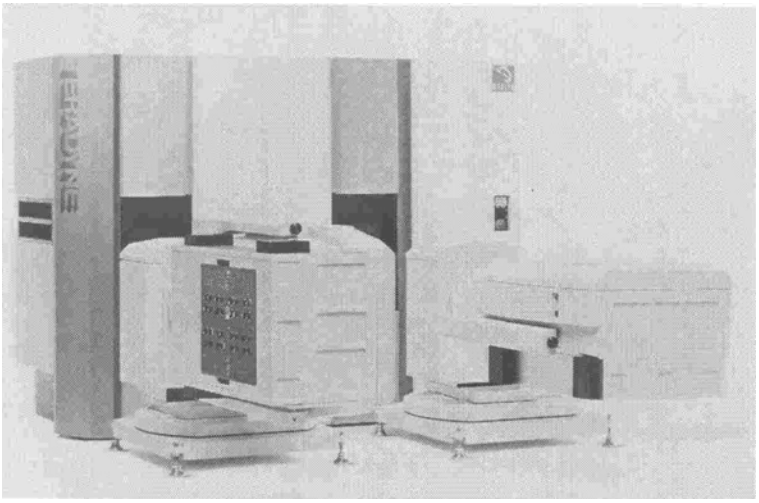


Figure 2-7 A Memory Tester

Photo courtesy of Teradyne Incorporated.

Memory Devices

Memory devices are considered digital and many DC test parameters are common to both memory and non-memory digital devices. Memory devices, however, require certain types of functional test procedures that are unique to memories. ATE memory test systems use an algorithmic pattern generator (APG) to generate the functional test patterns. The APG is capable of producing complex functional test sequences from hardware so they do not have to be stored as test vectors. Typical functional test patterns for a memory device are checkerboard, inverted checkerboard, walking ones, walking zeros, and butterfly, to name but a few. The APG generates these patterns each time a device is tested, unlike a non-memory tester which stores pre-generated patterns in vector memory then accesses the data during each test. Memory devices typically require long test times to exercise all of the required test patterns. In an effort to reduce test costs and increase test throughput Memory test systems often test numerous devices in parallel.

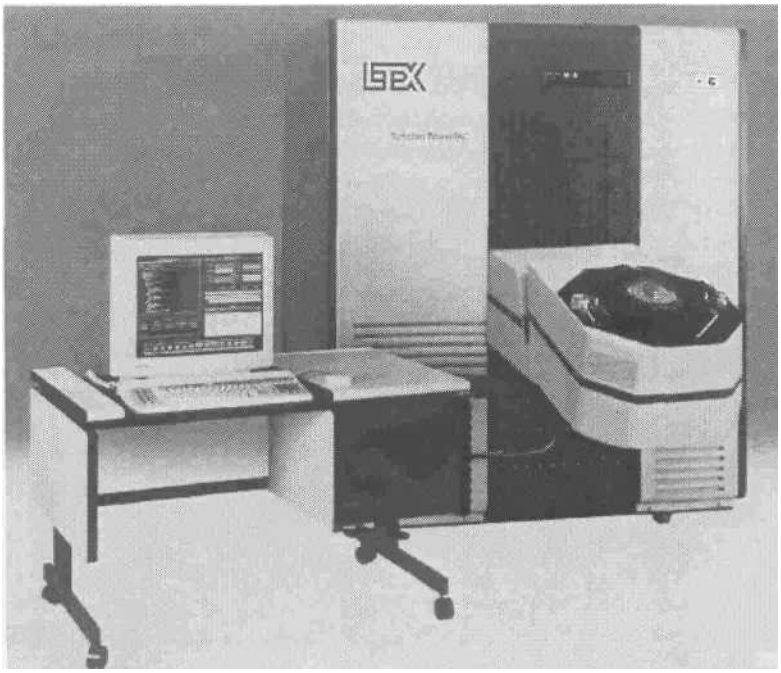


Figure 2-8 A Linear Tester
Photo courtesy LTX Inc.

Analog or Linear Devices

Analog device testing requires precise generation and measurement of both voltage and current. Force and measurement accuracy in the range of microvolts and picoamps are sometimes necessary. Analog devices require little (if any) digital stimuli in comparison to digital devices. The DC test parameters of analog circuits are also quite different from digital circuits. Verification of analog parameters requires specific test instrumentation. Analog test systems are often designed using rack mount instrumentation which is custom selected to test a specific device. Examples of analog test parameters are gain, input offset current and voltage, linearity, common mode rejection, power supply rejection, dynamic response, frequency response, settling time, overshoot, harmonic distortion, signal to noise ratio, response time, cross-talk, adjacent channel interference, accuracy and noise.

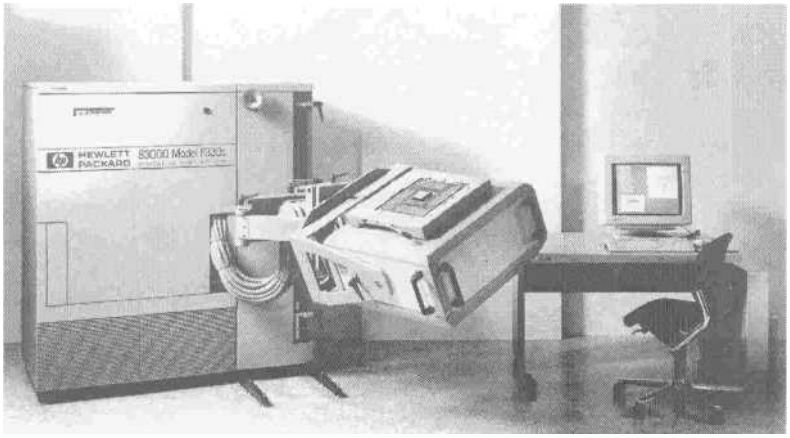


Figure 2-9 A Mixed Signal Tester

Photo courtesy Hewlet-Packard.

Mixed Signal Devices

Mixed signal device contain both digital and analog circuitry and thus required test systems which have equipment for testing both types of circuitry . Mixed signal testers come in two flavors: Big D Little A testers are designed for circuits that are primarily digital. They do DC and functional tests very effectively but support only minimal analog testing; Big A Little D testers are designed for circuits that are primarily analog. They accurately test analog parameters but are very limited in functional test capabilities.

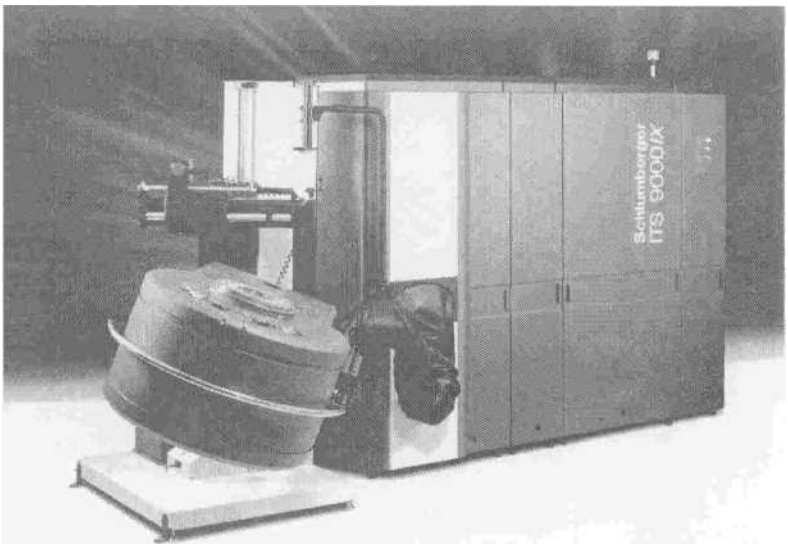


Figure 2-10 A Digital Tester
photo courtesy Schlumberger Technologies Inc.

Digital Devices

Devices with only digital circuitry are tested with digital test systems. These vary greatly in price, performance, size and options. Bench-top testers are available on the low end of the price/performance scale. They are designed for testing small pin count, low complexity devices. They typically run at a maximum speed of 20MHz and store only a limited number of test patterns. These testers are used to test small scale (SSI) and medium scale (MSI) integrated circuits.

At the high end of the digital tester scale are the super fast, super high pin count systems. These high performers may operate at test frequencies to 400MHz and offer up to 1024 tester channels. These systems have highly accurate timing resources and can store millions of functional test patterns. They are often used to characterize new VLSI circuits but their cost may prohibit their use in production testing.

In the middle of the tester scale we find the work horses of the semiconductor test industry. These testers are designed to provide the right mix of performance versus cost. The semiconductor industry is extremely sensitive to the cost of test, which can add substantially to the total manufactured cost of a device. Mid-performance testers operate from 50MHz to 100MHz and can be purchased with up to 256 tester channels. A variety of options are available on this type of test system.

To control the cost of test, it is critically important that the performance of the test system match the test requirements of the device. Using a high performance test system that offers more features than necessary to test a device will result in test costs which are too high. Using a low performance test system may result in inadequate test coverage. Finding the right balance between equipment cost and performance is essential to controlling the cost of test.

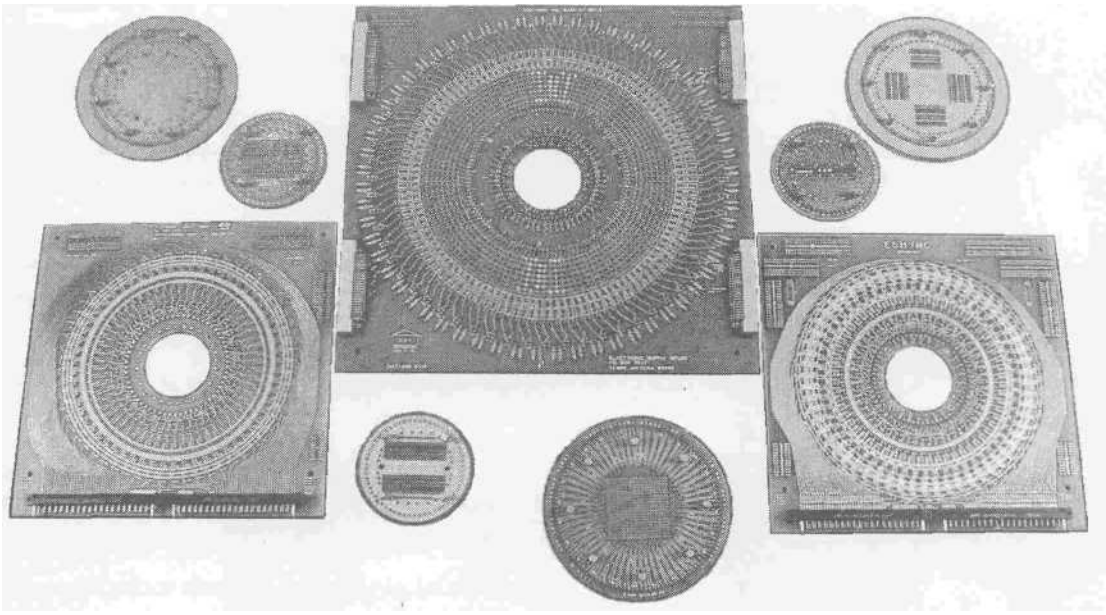


Figure 2-11 Tester Loadboards

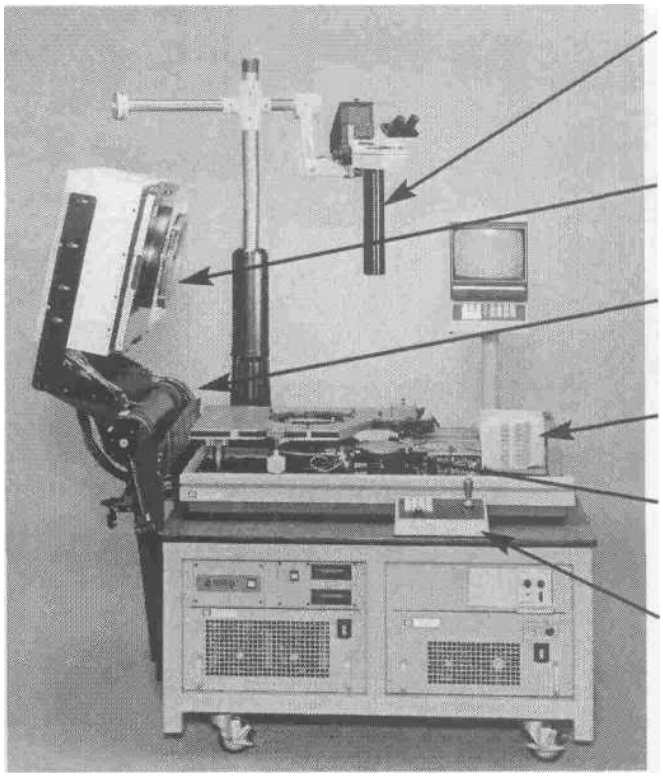
Photo courtesy ESH Inc.

Tester Loadboards

A loadboard or performance-board is a mechanical and electrical interface that connects the test system's test head to the device under test. The loadboard often mates to a wafer prober, device handler or other test hardware. The loadboard routes electrical signals from the pin electronics cards to the device under test.

During wafer test, the loadboard interfaces with the probe card. When testing devices by hand, the test socket is mounted on the loadboard. For high volume production testing, the loadboard interfaces with a device handler. Because there are many different types of equipment that the tester must mechanically and electrically mate with, there are a wide variety of loadboards.

When testing high speed or high current devices, a custom loadboard is required. These high performance custom circuit boards must be impedance matched to insure signal integrity; they often have matched line lengths (signal paths) to insure timing integrity. Custom loadboards can be very expensive and often take months to design and manufacture.



The black vertical tube in the center is a microscope and camera used to view the die to set probe alignment. When the test head is down the microscope goes through its center.

The test head (far left) supplies all the resources for testing the die. The test head is linked to the tester via a cable.

A torsion bar manipulator (under the test head) allows the test head to be gently lowered onto the prober.

A white boat of wafers sits at right center under the TV monitor

The wafer chuck (to the left of the boat) holds the wafer and positions each die underneath the probe card.

Manual alignment controls including a joy stick are visible in the foreground.

Figure 2-12 Wafer prober and a test head on a Torsion Bar manipulator

Photo courtesy Electroglas Inc.

Wafer Probing

During Wafer Test, the test head is mated directly to the wafer prober. The test head is flipped upside-down and connects with a prober interface assembly which contains the probe card (this type of testing is often called flip head sort). A single wafer is placed onto the chuck and the wafer is aligned with the probe card to insure a perfect fit. The probe card is then used to make contact with each die on a wafer. The probe card is not visible in Figure 2-12, but it is located in the center of the prober (near the chuck).

In the early years, wafers were loaded one at a time onto the chuck, and the alignment of the wafer was performed manually by a skilled probe operator. The probing area was open, as seen in Figure 2-12, allowing both light and contamination to reach the wafers. We now know that some test parameters can be affected by light, which may cause a device to fail. In recent years, contamination has become a larger concern due to the smaller geometry of the components on a die. Contamination can cause parameters to drift, devices to fail and in general, shorten the life of a device.

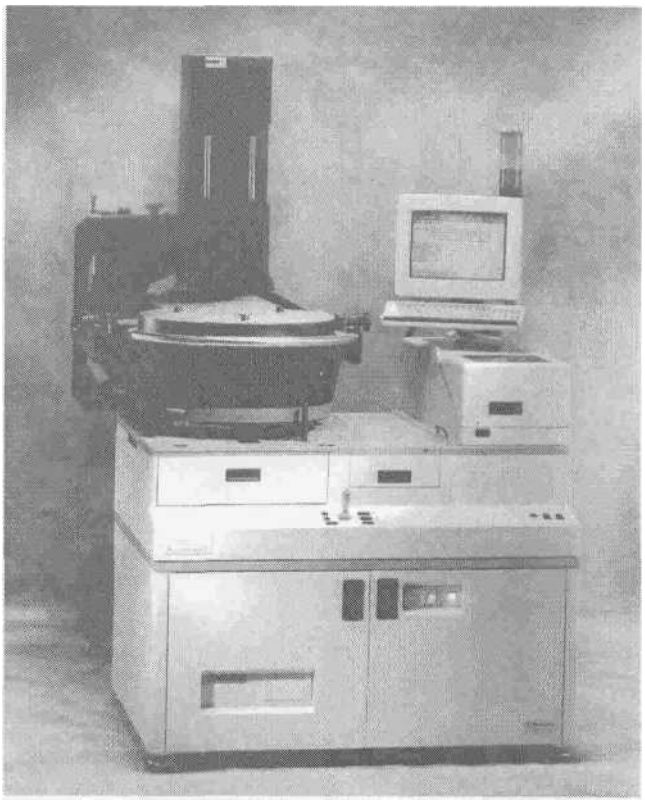


Figure 2-13 Fully Enclosed Prober with Test Head

Photo courtesy Electroglas Inc. and Credence Systems Inc.

Wafer probers are now designed so that light and contamination can no longer reach the wafers. The wafers are fully enclosed and protected the during the testing process.

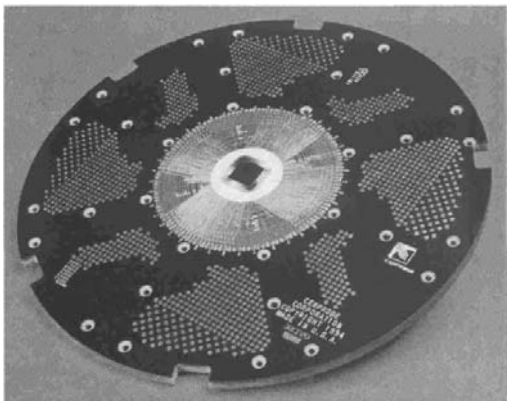
When the wafers leave the fabrication area they are placed in a protective container called a boat. A boat typically holds about 25 wafers. During the testing process the boat of wafers is placed inside the prober and all wafer handling and alignment functions are performed automatically by the prober.

Modern probers can ink defective dice, but many companies prefer to create wafer maps that store the locations of good and bad dice electronically. These maps are then used during the assembly process to separate the good dice from the defective dice. Electronic wafer mapping eliminates problems associated with prober inking attachments.

Modern probers are highly reliable and require very little attention. They have greatly eliminated the need for skilled probe operators

Probe Cards

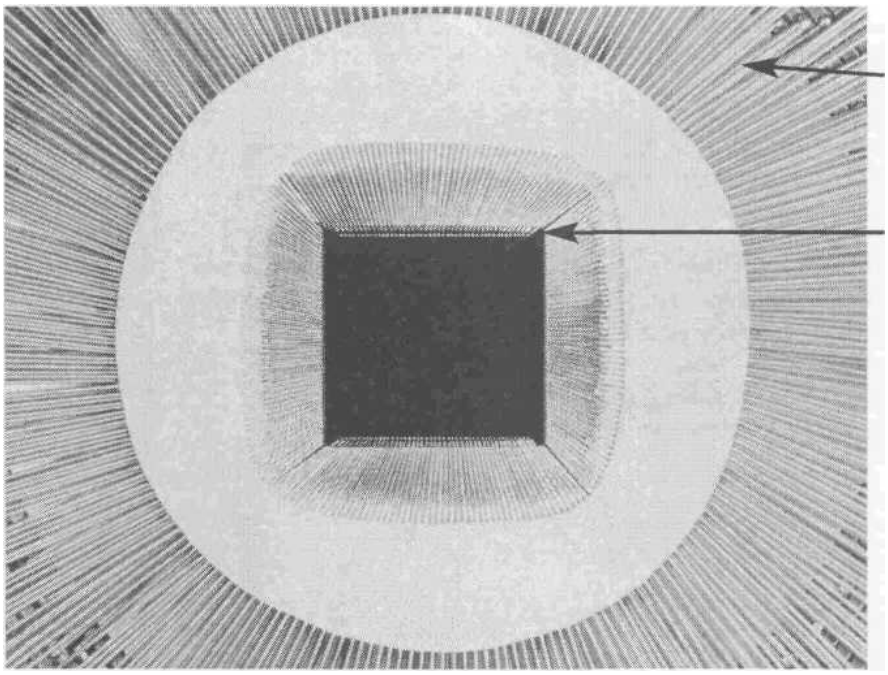
Probe cards connect the test head electronics to the individual pads of a die. In Figure 2-14, the probe card is physically part of the load board. In some cases the probe card attaches to the loadboard via a socket.



The probe card mechanically attaches to the test head.

Tester resources are routed through springy pogo pins that make electrical connection to the bottom of the probe card. Signals then travel through traces within the circuit board to the individual probes.

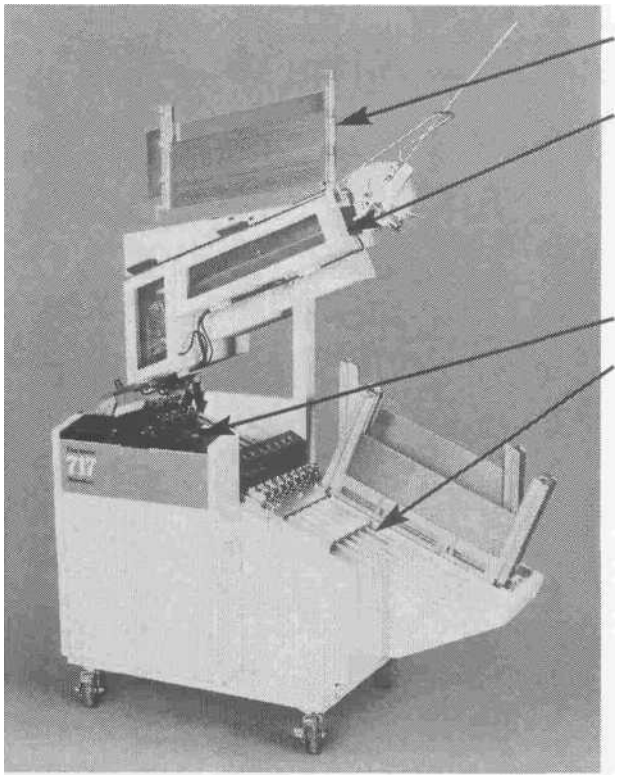
Figure 2-14 High Probe Count Probe Card
Photo courtesy Cerprobe Inc



Interconnect wires (outside of the white area) route the tester resources to the individual probes.

Each probe (inside the white area) makes electrical contact with a bond pad on the die.

Figure 2-15 Probes of a High Probe Count Probe Card
Photo courtesy Cerprobe Inc.



Untested devices are loaded into the top of the handler using rails or tubes.

Devices slide into a thermal chamber, which brings the devices up or down to the proper temperature.

One device at a time is loaded into the contactor mechanism (left middle), where electrical contact is made to the test head (not shown).

A user station allows handler setup and control.

Tested devices are loaded into specific bins or tubes (lower right) based on the results of the testing (called binning).

Figure 2-16 Device Handler with thermal chamber

Photo courtesy Daymark Inc.

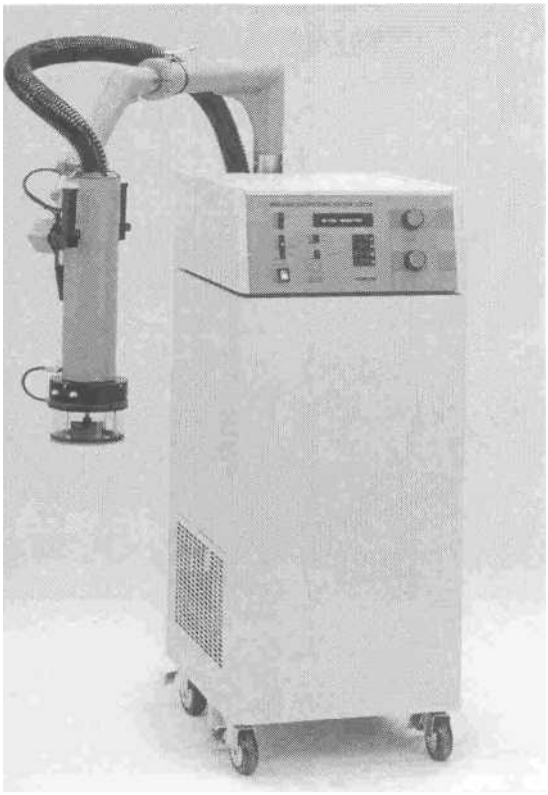
Device Handlers

Just as there are a wide variety of semiconductor package types, there are as many different styles of handlers to accommodate them. Figure 2-16 illustrates a handler which tests DIP packages (dual in-line package).

After device assembly is complete, packaged devices are placed in tubes or trays so that they can be taken to the test area. During production testing, untested devices are loaded into the handler storage bin. The handler then selects an untested device, moves it to the contact area, which interfaces to the test head, and inserts the device in the contactors. On a signal from the handler that all is ready, the test system performs the required tests and makes a pass/fail decision.

Information is sent from the test system to the handler regarding the proper bin assignment and the device is moved and stored in the appropriate bin location. Bins are used to group devices together based on their test results. A device handler must have a minimum of three bins: an untested bin, a good device bin and a reject bin. Other bins are also useful—for instance, it is convenient to separate rejects into categories such as: DC rejects, AC rejects and Functional rejects. There may also be various grades of good devices based on a parameter such as operating speed, e.g. 100MHZ, 166MHZ and 200MHZ devices. The maximum number of bins a handler supports is limited by its physical size.

Many handlers also offer environmental chambers for testing devices at temperatures different from room temperature.



A user control station allows temperature setting.

When the thermal chamber (the clear round chamber at the end of the tube at left) is placed over a device, the device is heated or cooled to a set temperature.

Figure 2-17 Temperature forcing system

Photo courtesy Thermonics Inc.

Temperature Forcing Units

Some devices must be tested at various temperatures. For example, a device specifications may state that the circuit must perform properly between -55° C and +125° C.

In the past, small hand held thermal probes were used to heat devices to 70° C or 125° C. This method was time consuming and troublesome because the handle of the thermal probe would become excessively hot and it was often difficult to make good thermal contact with the device under test. Hand held probes also do not accurately control device temperature.

For low temperature testing, dry ice or liquid nitrogen was often used to cool a device. Using dry ice is not an accurate method of cooling a device because the temperature of dry ice is approximately -60° C and device specifications typically call for -55° C or 0° C. Another problem with dry ice was simply having it available when it was needed.

Temperature forcing systems are generally the best method of heating or cooling devices when testing in a hand socket. Temperature forcing systems are programmable, accurate and fully self contained, requiring only external electrical power

Chapter 3 Introduction to Test

Objectives

This section explains:

- The purpose of testing
- Terms used to describe testing
- Basic rules of test engineering
- The components of a basic test system
- The functions of the PMU and Pin Electronics card
- The functions of a simple device and test program

Basic Terms

There are many terms used to describe semiconductor testing. Some basic terms are defined here, other terms are defined throughout this text. A complete list of terminology is located in the Glossary.

The DUT

The semiconductor device being tested is often called the *DUT* (Device Under Test). It is also sometimes called a *UUT* (Unit Under Test).

Terms that Apply to the DUT

Signals pass to and from the DUT via connection points called pins. A list of the various pin types, that can be found on digital devices follows:

Signal Pins	Input, output, Tri-State® and bi-directional pins (power and ground pins are not included in this definition). Signal pins have a structure which power pins.
Input Pin	A device pin that acts as a buffer between external signals and the internal logic of a device. The input senses the voltage applied to it and transmits a logic 0 or logic 1 level to the internal logic of the device.
Output Pin	A device pin that acts as a buffer between the internal logic of a device and the external environment. An output pin provides the correct voltages to produce a logic 0 or logic 1 level and also supplies the IOL/IOH current.
Three-State Output	A device pin that functions as an output pin and has the added capability of turning-off- (going to a high impedance state) Also called Tri-State® output.
Bi-directional Pin	A device pin that functions as an input, an output and is also capable of turning off (going to a high impedance state).

Power pin	A device pin that is connected a power supply or ground. VDD and VCC are typical examples of a power pins. VSS and ground are also identified as power pins. Power pins have a structure that is different from signal Power pins
VCC	The power pin that supplies voltage for a TTL. device.
VDD	The power pin that supplies voltage for a CMOS device.
VSS	The power pin that provides a return path for the power supplied to the VDD or VCC pin.
Ground	To connect a signal pin or other electrical node to the test system reference node or to VSS. On a device with only one power supply voltage, VSS is often called "ground."

The Test Program

The purpose of a semiconductor test program is to control the test hardware in a manner that will guarantee that the DUT meets or exceeds all of its design parameters. The design parameters are defined in detail in the device specification.

The test program is often segmented into various parts such as DC tests. functional tests and AC tests. DC testing verifies voltage and current parameters. Functional testing verifies correct operation of the various logical functions of the device. AC. testing verifies that the device can perform the logical operations within -specified timing constraints.

The test program controls the test system hardware so that each test gives a pass or fail result. If the test result is a pass, the device meets or exceeds the design specifications. If the result is a failure, the device does not operate within its specifications and should not be used in its intended application. The test program may also separate devices into categories based on their performance. This is called *binning.* A microprocessor that can function correctly at 150Mhz might be categorized as the best part or a "bin one." A microprocessor that can function correctly at l00Mhz is not as good as the one that works at 150Mhz, but it's not one that should be thrown away. It might be categorized as a "bin two" and sold to a different customer than the 150MHz device.

The test program must also be capable of controlling external hardware such as device handlers or wafer probers. It must collect test results and present the results in a summarized format. The test results provide valuable information to the test and product engineers and can be used to help increase yields.

What is The Correct Way to Test?

The question has been asked many times "What is the right way to create a test program?" Unfortunately, there is no single definition of the right way or the best way to test. What is correct for one situation may not be the best for another. Many factors influence the way a test program is constructed. Let's look at some of the more influential factors.

- What is the **purpose** of the test? The following list illustrates the most common uses of semiconductor test programs. Each of the items listed has unique requirements thus needs a unique test program.
  - **Wafer Test** — The testing of individual devices when they are still in wafer form. This is the first attempt at separating good dice from bad. This activity is also called wafer sort or die sort.
  - **Package Test** — Wafers are cut into individual dice and each die is assembled into a package. The packaged device is then tested to insure that the assembly process was correctly performed and to verify that the device still meets its design specifications. Package test is also called final test.

Quality Assurance Test — Performed on a sample basis to insure that the package test was performed correctly.

- **Device Characterization** — Device Characterization is the process of determining the operating extremes of individual device parameters.
- **Pre/Post Burn-In** — The testing of devices before and after they are "burned in" to verify that the process did not cause certain parameters to drift. This process weeds out infant mortality devices (those which have a defect that causes them to fail soon after they are first used)..
- **Military Testing** — Involves performing rigorous testing over a temperature range and documenting the results.
- **Incoming Inspection** — Testing of devices by a customer to insure the quality of the devices purchased before using them in an application.
- **Assembly Verification** — Verifies that the devices survived the assembly process and that they were assembled correctly. The tests performed during assembly verification are similar to that of package testing and may be a subset of package testing. This activity is usually performed offshore.
- **Failure Analysis** — The process of analyzing device failures to determine why the device failed. Determining the cause of a failure yields information that can improve device reliability.

♦ What are the **capabilities** of the test system? Test programs are designed to take full advantage of the test system's capabilities. Various test methods may be used depending on the hardware and software capabilities of the test system.

High performance testers

- **Highly accurate timing** — capable of accurate high speed measurements
- **Large vector memory** — eliminates the need to reload test patterns
- **Multiple PMUs** — capable of parallel measurements, reducing test time on DC tests
- **Programmable current loads** — simplifies test hardware, adds flexibility
- **Timing and levels per pin** — eases program development, reduces test times

Lower cost testers

- **Low speed / low accuracy** — may be inadequate to match device specifications
- **Small vector memory** — may require reloading test vectors, increases test times
- **Single PMU** — can only make serial DC measurements, requires long test times
- **Shared resources** (timing / levels) — increases complexity of test program, increases test times

♦ What are the **financial considerations?** This is probably the single most important factor in deciding what needs to be tested and what is the best way to meet those needs. Testing can add significantly to the manufacturing cost of a device, therefore many decisions regarding test are based solely on the selling price vs. test costs. For example, a device might be used in a game and sell for $0.15. The same device may also be used in a satellite and sell for $350.00. A unique specification will exist for each use, requiring two unique test programs.

$350.00 selling price per device can support expensive test costs $0.15

selling price per device can support only minimal test costs

♦ What is the **test philosophy** of the company developing the test program? A test philosophy is a consensus of opinion of what is the best method of testing within a given company. It is based on their particular requirements, the selling price of their devices and often it is influenced by past experiences.

Before test program development begins the test engineer must thoroughly consider each of the items mentioned above to determine the *best* solution. Developing test programs is not a matter of right or wrong, it is more of identifying the best solution for a give set of circumstances

The Test System

The test system is electronic and mechanical hardware used to simulate the operating conditions that a DUT will experience when used in an application, so that defective devices can be found. The test system is often known as ATE or Automated Test Equipment.

The test system hardware is controlled by a computer which executes a set of instructions (the test program). The tester must present the correct voltages, currents, timings and functional states to the DUT and monitor the response from the device for each test. The test system then compares the result of each test to pre-defined limits and a pass/fail decision is made.

A test system is actually a collection of power supplies, meters, signal generators, pattern generators and other hardware items which all work collectively under one main controller.

What's in a Test System?

Figure 3-1 shows the basic blocks that all digital test systems contain. Many new test systems contain far more hardware than what is shown here, but this diagram is a good starting point.

The CPU is the system controller. It contains the computer which controls the test system and provides a means of moving data into and out of the test system. Most new test systems offer a network interface as well as magnetic media for data transfers. The hard disk and CPU memory are used to store information locally; the video display and keyboard are used by the test operator to interact with the test system.

Basic Test System Components

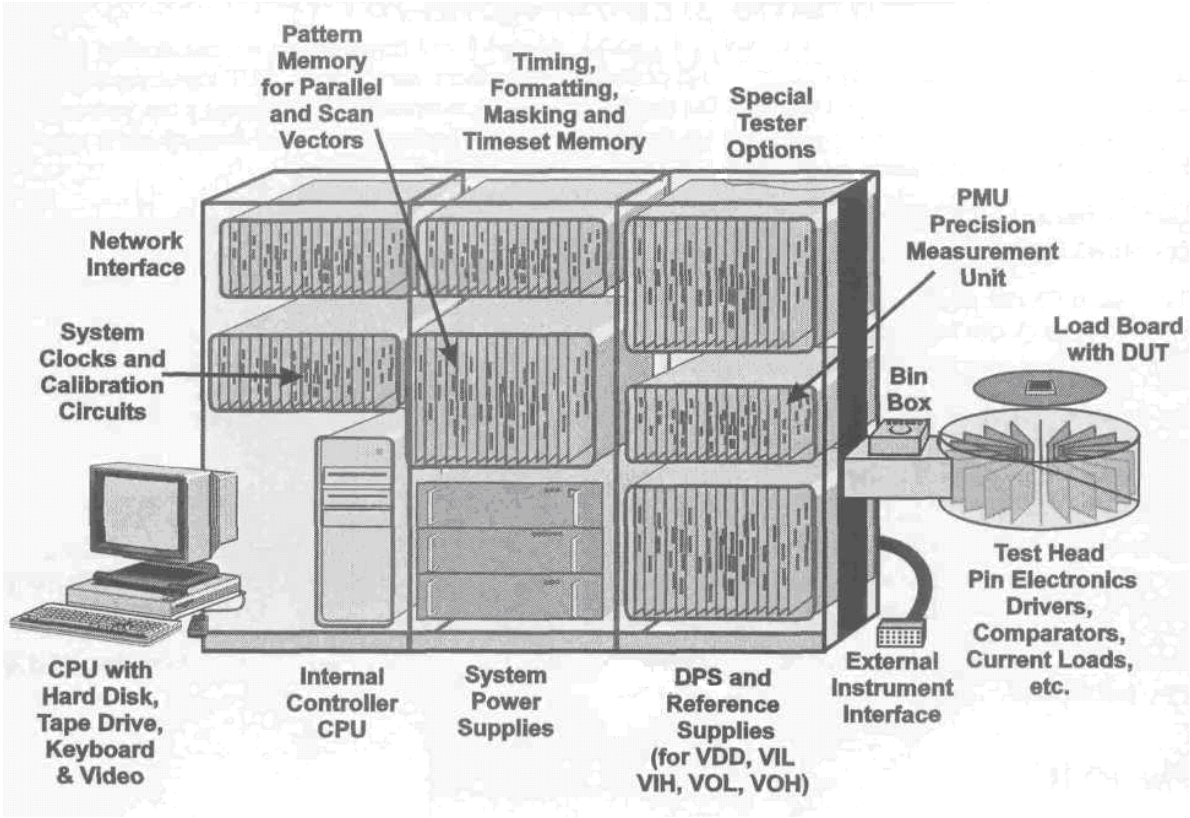


Figure 3-1

The DC subsystem contains the Device Power Supplies (DPS), the Reference Voltage Supplies (RVS) and the Precision Measurement Unit (PMU). The DPS supplies voltage and current to the DUT power pins (VDD/ VCC). The RVS supplies voltage references for logic 0 and logic 1 levels to the driver and comparator circuitry located on the pin electronic cards. These voltages set VIL, VIH, VOL and VOH. Less expensive and older test systems may have a limited number of RVS supplies, so only a limited number of input and output levels can be programmed at one time. When tester pins share a resource such as the RVS, that resource is considered a *shared resource.* Some test systems are said to have a *tester per pin* architecture which means that they have the ability to set input and output levels and timing independently for each pin. A *tester pin,* also called *tester channel,* is circuitry on the pin electronics card which supplies and/or detects voltage, current and timing for one DUT pin.

The DC subsystem also contains PMU circuitry (Precision Measurement Unit) to make accurate DC measurements. Some systems have a PMU per pin, located in the test head.

Each test system has high speed memory, called *pattern memory* or *vector memory,* to store test vectors or test patterns. *Test patterns,* also known as the *truth table,* represent the states of inputs and outputs for the various logical functions that the device is designed to perform. Input, or drive, patterns are supplied to the DUT by the test system from pattern memory. Output, or expect, patterns are compared against the response from the output pins of the DUT. During a functional test, vector patterns are applied to the DUT and the DUT's

responses are monitored. If the expected response data does not match the output data from the DUT, a functional failure occurs. There are two types of test vectors -- parallel vectors and scan vectors. Many test systems support both types.

The timing subsection has memory to store formatting, masking and timeset data for use during functional testing. The signal formats (wave shapes) and timing edge markers are used for DUT input signals and strobe timing for sampling DUT output signals. The timing subsection receives drive patterns from pattern memory and combines them with timing and signal format information to create formatted data which is sent to the driver section of the pin card and then to the DUT.

Special Options includes a variety of possibilities such as algorithmic pattern generators for memory test or specialized hardware modules used to perform analog tests.

The System Clocks provide a means of synchronizing the movement of information throughout the test system. These clocks often run at much higher frequencies than the functional test rate. Many test systems have calibration circuitry which can automatically verify and calibrate the system timing.

The External Interface communicates with other hardware such as wafer probers, device handlers or special test instruments. This interface typically uses a serial or GPIB (IEEE-488) protocol.

The test head contains pin electronics cards and interfaces to the DUT test hardware or loadboard.

The *Loadboard* is the physical interface between the test system and the DUT. It connects to the DUT through a test socket, a probe card or a device handler. It holds interface components required to test the DUT, such as relays, resistive loads or power supply decoupling capacitors.

Near the test head is the *test box* or *bin box.* The bin box typically contains *Start* and *Reset* buttons and displays pass/fail results.

The PMU

The Precision Measurement Unit (PMU) is used to make accurate DC measurements. It can force current and measure voltage or force voltage and measure current. Some test systems have only one central PMU that must be shared across all pin channels of the tester. Others have more than one PMU which accesses multiple channels, typically in groups of eight or sixteen. High end test systems have PMU per pin capability, which has a PMU on every tester channel.

Precision Measurement Unit (PMU)

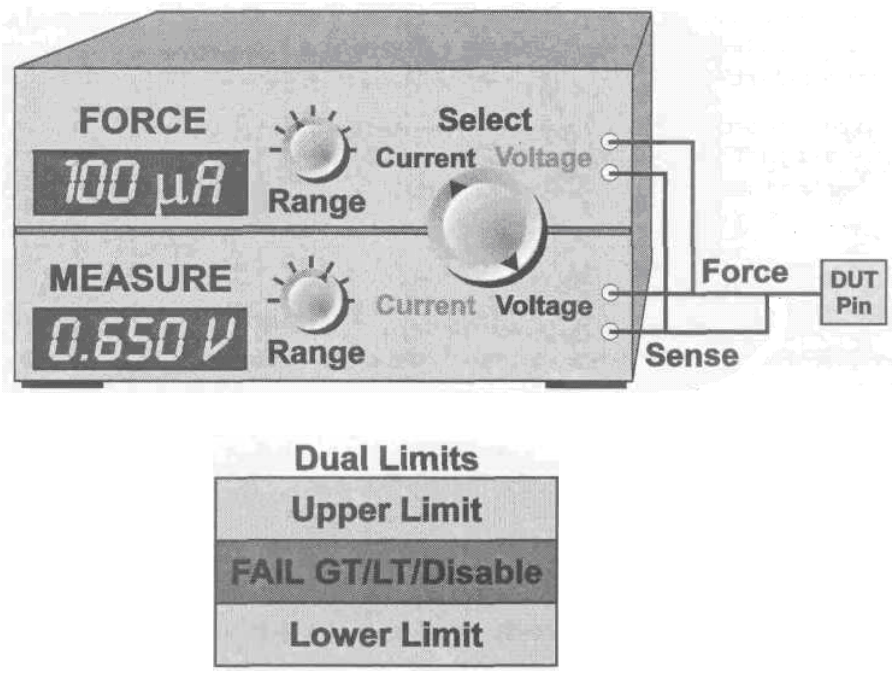


Figure 3-2

Force and Measurement Modes

In ATE, the term force (as in *force voltage* or *force current)* describes the application of a certain value of voltage or current by the test system. *Apply* can be substituted for the word force.

When programming the PMU, the force function is selected as either current or voltage. If current is forced, the measurement mode is automatically set to voltage. If voltage is forced, the measurement mode is automatically set to current. Once the force function is selected, the force value must be set.

Force and Sense Lines

To improve the voltage forcing accuracy of the PMU, a four wire system is used. *Four wire* systems use *2 force lines* to carry current and *2 sense lines* to monitor the voltage at the point of interest (usually the DUT). Ohm's Law states that a voltage is produced across a resistance when current flows through the resistance. All wire has resistance so, depending on the current through the force lines, the voltage at the DUT is different from the voltage at the PMU force unit output.

Using 2 separate (non-current carrying) sense wires to measure the voltage at the DUT keeps the voltage drop caused by current flow through the force lines from causing an error or offset in the voltage. The point at which the force and sense line are connected together is called the Kelvin Connect Point.

Range Settings

A PMU force range and measurement range must be selected. Proper range selection insures the most accurate test result. Be aware that the force and measure ranges have a limiting effect on the PMU. The force range will determine the maximum forcing capability of the PMU, so if the PMU is programmed to force 5 Volts and the 2 Volt force range has been selected, only 2 Volts will be forced. Likewise, if the 1mA measurement range is selected, the maximum current that can be measured is 1mA regardless of the actual current flow.

It is important to note that neither the force nor the measurement range of the PMU should be changed while it is connected to a DUT. Changing the range causes noise spikes that may damage the DUT. A noise spike is when a signal level abruptly changes its voltage level for a very short time. A noise spike is also called a *glitch.*

Limit Settings

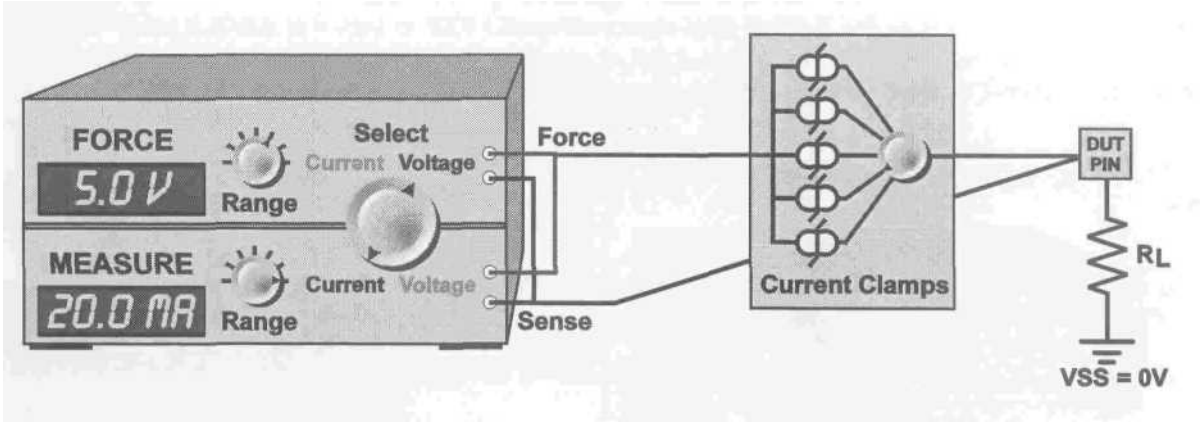
The PMU has two programmable measurement limits—an upper and a lower limit. The limits may be used individually (one limit enabled while the other is disabled) or they may be used together (both limits enabled).

The upper limit is used to make a *Fail Greater* Than comparison and the lower limit to make a *Fail Less* Than comparison. Failing the "Fail Greater Than" limit means the measured value was more positive than the upper limit setting. Failing the "Fail Less Than" limit means the measured value was more negative than the lower limit setting.

Clamp Settings

Most Precision Measurement Units have voltage and current *clamps* which are set from within a test program. A clamp is a circuit that limits the amount of voltage or current that is supplied by the PMU during a test in order to protect the test operator, the test hardware and the DUT.

Current Clamp



When forcing 5V and current limit is 20mA, set Iclamp at 25.0mA:
When RL = 250Ω, Iout = 20mA
When RL<= 200Ω, Iout = 25mA (clamp current)

Figure 3-3

When the PMU is used in Force Voltage mode, a current clamp **must** be set to limit the maximum current which flows during the test. When forcing voltage, a PMU delivers as much current as necessary to sustain the voltage. If a DUT pin is shorted to ground (or another supply), the forcing unit will increase the current to try to force the pin to the programmed voltage. This may result in a large enough current flow through the DUT pin to burn probe cards, circuit traces, pin electronics components, fingers, adjacent DUTs, etc.

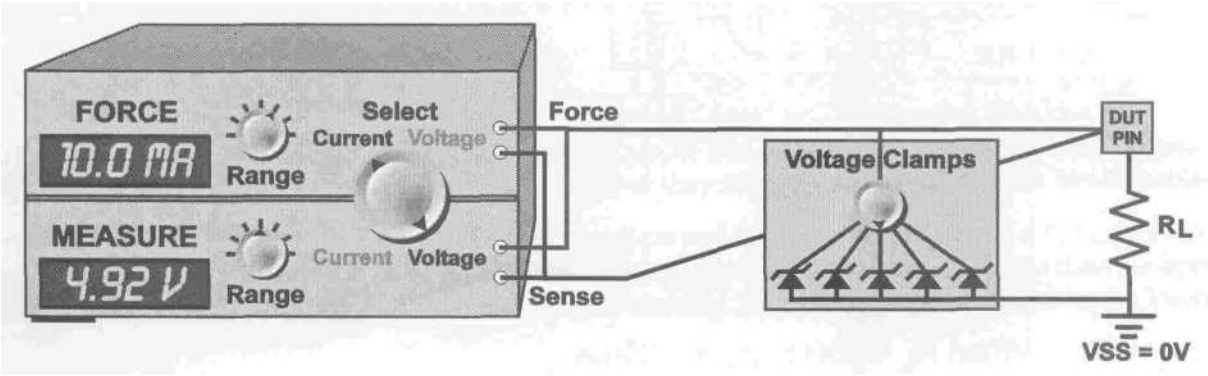
When forcing voltage, the PMU measures current. Because the current clamp limits the amount of current supplied by the PMU, the current clamp value must be set outside of the test limits otherwise the current clamp will prevent a "too much current" failure.

Figure 3-3 shows the PMU forcing 5.0V across a 250 Ohm load. In actual testing, the DUT is the resistive load. From Ohm's Law (I=E/R) we know that the 250 Ohm load will restrict the current flow to 20mA during this test. The device specification may state that the maximum acceptable current is 25mA. This means the fail limit would be set to **Fail GT 25mA** and the current clamp could be set to **30mA.**

If a defective DUT presents a load of 10 Ohms, the resulting current will be 500mA unless a current clamp is programmed to limit the current. A current flow of 500mA may cause damage to the test system, the interface hardware or the DUT. However, if a current clamp is programmed to 30mA the maximum current flow would

be limited by the damp circuit to a much safer value. "Why 30mA?" you may ask. 30mA is greater than the fail limit of 25mA, allowing the test to fail when a defective device is encountered, but the current will be limited to a safe value. **The clamp value must always be set outside of the fail limits; if not, the test will never fail.**

Voltage Clamp



For I_{force} = 10.0 mA:
When R_L = 500Q, VOUT = 50V
When R|_ = open, VOUT = COMPLIANCE or CLAMP voltage.

Figure 3-4

When the PMU is used in the Force Current mode, a voltage damp **must** be programmed to limit the maximum voltage that is produced during the test. When forcing current, a PMU will increase the voltage at its output to try and force the programmed current. If the pin is an open circuit, the voltage may increase to a value that is dangerous to the DUT, the hardware, the operator, etc.

When a test is made with the PMU in the current forcing mode we expect to measure a voltage within a certain defined limit range. If the DUT is defective, the actual voltage measured may be far greater than the expected voltage. To be sure that the voltage produced by the PMU stays within safe limits, a voltage damp should be programmed.

The voltage damp will limit the amount of voltage supplied by the PMU, so the voltage damp value must be set outside of the test limits to make sure the voltage can reach the value required by the DUT. When forcing current a voltage damp must always be set.

Figure 3-4 shows the PMU forcing 10mA through a 500 ohm load. In actual testing, the DUT will present the resistive load. Based on Ohms law (E=I/R) we know that 10mA flowing through a 500 Ohm load will develop a voltage drop of 5 volts. In this example, assume the device specification states that the maximum acceptable voltage is 5.25 volts. This requires a fail limit of **"Fail GT 5.25 volts"** and the voltage clamp could be set to **6 volts.**

In the course of device testing, a defective DUT may present a high resistance or worse yet, an open circuit. When the PMU attempts to force current into an open circuit (or a very high resistance) the result will be the maximum voltage that the PMU is capable of producing. Think of it this way, if 10mA is forced into 500 Ohms the result is 5 Volts; if 10mA is forced in 5000 Ohms the result is 50 Volts; if 10mA is forced into an open circuit (infinite Ohms) the result is an infinitely high voltage. In this example, if a voltage clamp is programmed to 6 Volts, the test will fail when the measured voltage is greater than 5.25 Volts but the PMU voltage will be limited to the programmed value of the clamp.

The Pin Electronics

The *pin electronics* (also called the Pin Card, PE, PEC or I/O card) is the interface between the test system resources and the DUT. It supplies input signals to the DUT and receives output signals from the DUT.

Each test system has its own unique design but generally the PE circuitry will contain:

- *Driver* circuitry to supply input signals.
- I/O switching circuitry for turning drivers and current loads on and off.
- *Voltage Comparator* circuitry for detecting output levels.
- A connection point to the PMU.
- Programmable current loads.
- Possibly additional circuitry for making high speed current measurements.
- Possibly a per pin PMU

Although there are many variations of this design, Figure 3-5 represents a single tester channel on a typical pin electronics card for a digital test system.

Pin Electronics

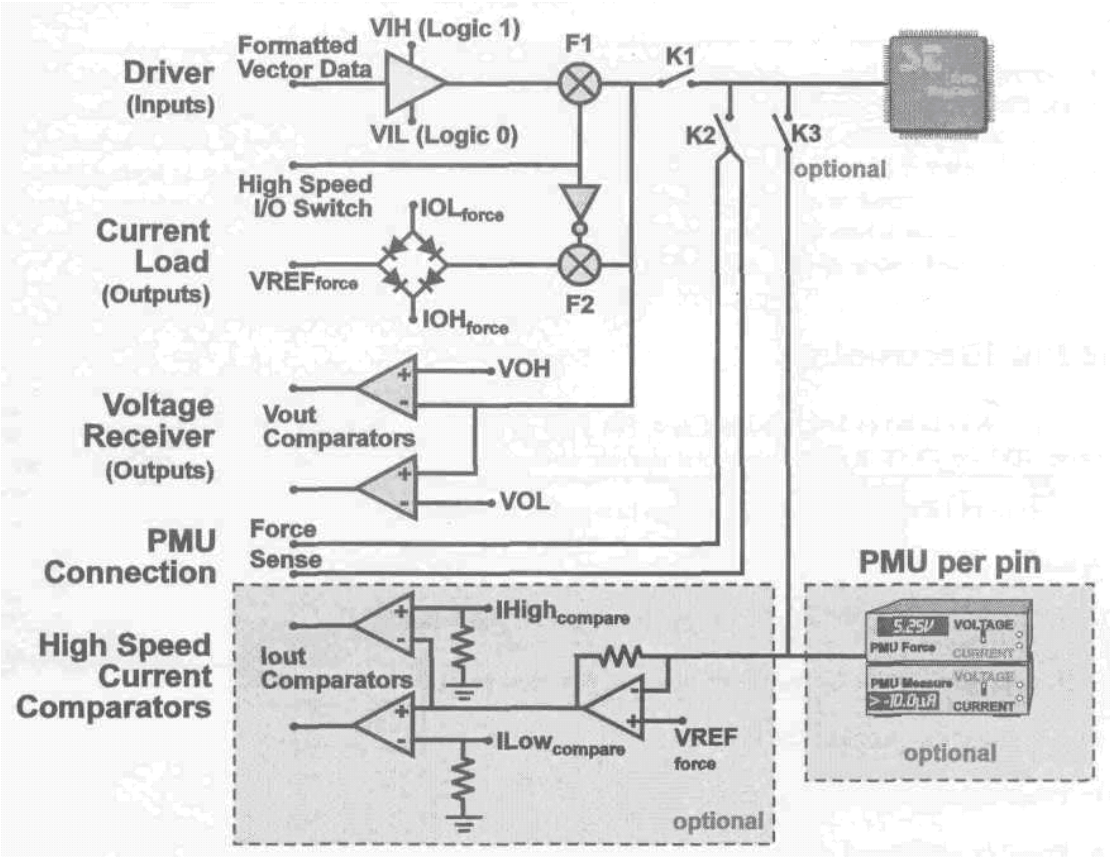


Figure 3-5

The Driver

The driver circuitry receives formatted signals, called FDATA, from the high speed section of the test system. As the signal passes through the driver, the VIL/VIH references from the Reference Voltage Supplies (RVS) are applied to the formatted data. If the FDATA instructs the driver to drive to a logic 0, the driver will drive to the VIL reference. VIL (Voltage In Low) represents the maximum guaranteed voltage value that can be applied to an input and still be recognized as a logic 0 by the DUT circuitry.

If the FDATA instructs the driver to drive to a logic 1, the driver will drive to the VIH reference. VIH (Voltage In High) represents the guaranteed minimum voltage value that can be applied to an input and still be recognized as a logic 1 by the DUT circuitry.

When the tester channel is programmed as an input, FI FET turns on and the K1 relay is closed allowing the signal from the driver to pass through to the DUT. When the tester channel is programmed as an output or is in a "don't care" mode the FI FET is turned off and the signal from the driver will not pass through to the DUT.

The FI FET is a Field Effect Transistor used as a very high speed switch. It isolates the driver circuitry from the device under test. The FI FET is used during IO switching, which is when the DUT alternates between

receiving data from the test system (reading data) and supplying data to the test system (writing data). The same pins function as both inputs and outputs. If the tester channel is programmed as an input, the FET is on. If the tester channel is not programmed as an input the FI FET is off, which prevents the driver signal from reaching the device under test. It is important to insure that the DUT and the driver are not trying to drive a voltage onto the same tester channel at the same time. This is called an *I/O conflict ox bus contention.*

Current Loads

The *Current Loads,* also known as *Dynamic Loads* or *Programmable Current Loads,* act as a load to the DUT outputs during functional tests and can be programmed to supply positive and negative currents. *Positive current* flows from the test system into the device and *negative current* flows from the DUT into the test system.

The dynamic loads provide both IOH (Current Out High), which is the amount of current that a DUT output must source when driving a logic 1, and IOL current (Current Out Low), which is the amount of current that a DUT output must sink when driving a logic 0.

After the IOL and IOH current values are set by the test program, the VREF voltage is used to set the *switching point* of the IOL and IOH currents. The switching point is the output voltage above which IOH flows and below which IOL flows. When the output voltage from the device under test is more negative than VREF, IOL current flows. When the output voltage from the device under test is more positive than VREF, IOH current flows. The current loads are also used during the functional Tri-State® tests and the functional opens and shorts test.

The F2 FET is also a field effect transistor used as a high speed switch. The F2 FET isolates the current load circuitry from the device under test and is used during IO switching. If the tester channel is programmed as an output, F2 is on, allowing current to flow to and from the device under test. If the tester channel is programmed as an input, F2 is off.

The Voltage Comparators

The Voltage Comparators are used during functional testing to compare the output voltage of the device under test to reference voltages supplied by the RVS. The RVS supplies a reference for a valid logic 1 (VOH) and a valid logic 0 (VOL). If the DUT output voltage is equal to or less than VOL, it is recognized as a logic zero. If the DUT output voltage is equal to or greater than VOH, it is recognized as a logic one. If the output voltage is greater than VOL but less than VOH, it is considered to be a Tri-State® level or a bad output.

The PMU Connection

When the PMU connects to a device pin, the K1 relay is opened first, then the K2 relay is closed. This sequence isolates the PMU from the I/O circuitry of the pin electronics card.

The High Speed Current Comparators

Some test systems offer a way to measure small currents quickly as an alternative to having a PMU for each tester channel. Current comparators are used to make high speed leakage measurements. The K3 relay connects the *current comparators* to the pin under test. If the test system has PMU per pin capability, the current comparators are not necessary.

The PPPMU

Some test systems offer a per pin PMU (PPPMU) that allows the test system to measure either voltage or current on every pin simultaneously. Like the PMU the PPPMU can force current and measure voltage or force voltage and measure current. PPPMUs typically lack all the capability of the standard system PMU.

Basic Rules of Test Engineering

The following set of rules will apply in most instances. They are defined here because they seem to be recurring problems found in many test programs. If you should ever feel a need to intentionally violate these rules, make certain that you fully understand the consequences. Some of these rules may seem obvious, but it is easier than you think to violate these rules, depending upon the exact test hardware being used.

- Never functionally test an input pin as if it were an output. This can be accomplished by accidentally setting the output compare mask on an input pin. Remember, inputs are not directly tested for pass/fail results during functional testing. Signals are applied to inputs, outputs are tested.
- Never connect a tester pin driver to an output pin of a DUT. The result of this action will cause the test system and the device pin to both drive voltage and current at the same point, at the same time.
- Never float an input pin. A valid logic 0 or 1 level must always be supplied to an input pin. Floating CMOS inputs may result in device *latch-up,* an SCR effect which can destroy the device by drawing too much current.
- Never supply a voltage which is above VDD or below ground to an input pin or output pin. This may cause a CMOS device to latch up.
- Always, when forcing voltage, set a current clamp to limit the amount of current provided by the test system.
- Always, when forcing current, set a voltage clamp to limit the amount of voltage provided by the test system.
- Never change a force range of the tester when connected to a device pin. Never change the forcing mode (voltage/current) of the PMU when connected to a device pin.

Introduction to Test — Review

1. Digital test programs are often segmented into three separate sections. These sections are:

2. Positive current flow is defined as: -
   a) Current flowing from the test system into the DUT
   b) Current flowing from the DUT into the test system
   c) Current flowing from a higher voltage to a lower voltage

3. A device pin that functions as both an input and output is called a:

4. Which resource of the test system hardware is used to provide voltage and current to power the DUT?
   a) RVS
   b) DPS
   c) Driver

5. Which resource of the test system hardware is used to perform very accurate voltage and current measurements?

6. Which type of testing verifies correct operation of the various logical operations or functions of the device?

7. When using the PMU to force current, which clamp should be used?
   a) A current clamp
   b) A voltage clamp
   c) Neither, clamps are not used when forcing current

8. When using the PMU to force voltage, which clamp should be used?
   a) A current clamp
   b) A voltage clamp
   c) Neither, clamps are not used when forcing current

Chapter 4 Device Specifications

Objectives

This section explains:

- Why a device specification is written.
- How to read a device specification.
- How to determine limits for each test.
- How to determine which tests to perform

Semiconductor testing is based upon a specification created specifically for each unique device design. The specification will define the intended functions and operational parameters of the device. To understand the various tests defined in the following chapters, it is first necessary to gain a basic understanding of device specifications.

Some type of device specification must exist before a plan for the test program can be developed. A *test specification,* a *design specification* or a *data book specification* may be used as a reference when developing the test plan. The specification is the foundation upon which the test plan is developed. Test program development should not begin until a specification and test plan have first been defined.

Basic Terms

As we begin to discuss the various types of specifications, there are a few terms related to testing a device that we need to understand:

Preconditioning	Setting a device into the proper logic state so that a test may be performed. A functional vector sequence is often required to prepare the DUT for a DC test.
Static	The DUT is in a fixed state of operation, with no input or output signals changing.
Dynamic	The DUT is actively changing states. Dynamic tests are associated with executing functional test vectors.

The Design Specification

The term design specification describes a document which defines the intended functions and performance characteristics of a new circuit design. The specification includes voltages, currents, timings and a description of the device functions. The design specification may be created by the sales and marketing department, the design engineering department or in some cases the end user. The creation of the design specification precedes the manufacturing of the silicon; after the device is produced it must be characterized and the actual performance of the device can be compared with the intended design requirements.

The Test Specification

The term test specification describes a document which shows step-by-step the process required to fully test the circuit. The test specification is normally produced by the semiconductor manufacturer for its own internal use through a cooperative effort of the design, test and product engineering departments. It defines the exact conditions (voltages, currents, timings and test patterns) to be used in program development and production testing. These conditions, when applied to the device under test, will guarantee that the device meets the intended design specification. If during the course of test development, it becomes necessary to modify a test parameter in order to get the device to function properly, the test/product engineer will immediately notify all appropriate parties of the modification (in writing). This modification will then be reflected in the published device specification.

The Device Specification

The terms device specification or data book specification refers to a published document which describes the functionality of a device along with its performance characteristics. Device specifications can normally be found in semiconductor device catalogs (also called data books). These specifications will define voltages, currents and timings but are often very general and lack the detail required to develop a complete test program. The device specification also serves as a contract between the seller and buyer, in that the semiconductor manufacturer guarantees that the device will perform as defined in the specification.

The device specification as it appears in a semiconductor device catalog is often used for program development. A typical device specification for a 256 x 4 static RAM will be used throughout this manual as an example.

Test Conditions and Limits

When testing a digital semiconductor device three types of tests are performed: DC, Functional and AC. The conditions and limits for each test must be determined. The device specification is reviewed and used as a starting point for test program development. Not all test conditions will be detailed in the specification. Also, conditions may exist in a device specification which can not be reproduced on a digital test system or may not apply to digital testing. It is important to understand the "intent" of each parameter and the standard procedures used to verify individual parameters. Standard procedures for each test will be presented in detail in the following chapters.

Parameters that Apply to Parametric Testing (DC)

When developing the DC test plan the following information must be determined. This information should be defined in the device specification.

- Input voltages and currents (VIL, VIH, EL, EH)
- Output voltages and currents (VOL, VOH, IOL, IOH, IOZH, IOZL)
- Power supply currents and voltages (VCC, VDD, VDDMIN, VDDNOM, VDDMAX, ICC, IDD)
- Pass/fail limits for each parameter

Note: Many DC tests require that the device be properly preconditioned. For example, in order to perform the VOH test (voltage out high) the device but first be preconditioned so that the outputs are set to the logic one

state. The device specification will typically not include preconditioning information. The information in the following chapters which describes each DC test will however specify all required preconditioning.

Parameters that Apply to Functional and AC Testing

When developing the AC test plan the following information must be determined. This data should be defined in the device specification.

- Input conditions (VIL, VIH)
- Output conditions (VOL, VOH, IOL, IOH)
- Power supply settings (VDDMIN, VDDNOM, VDDMAX)
- Tunings (frequency, pulse widths, setup and hold times, delays)
- Pass/fail limits for each parameter

Note: It may be possible to take a quick look at the AC (timing) specification and determine whether or not the target test system will be adequate to test the device. For example, if the device specification defines the operating frequency at 50MHz and the test system operates at a maximum of 20MHz, compromises will need to be made, or another test system may be selected for use. Other considerations are: minimum pulse widths and timing accuracies needed to make setup, hold and delay time measurements.

Logical Functions

The functionality of the device may be shown within the device specification as a truth table for simple logic devices. For complex devices the general functionality or behavior of the device may be described in written form. Complex devices may take many thousands or millions of instructions (input combinations) to fully test the device. Detailed test patterns for complex devices are most often developed by using computer aided design tools during the design process and are not included as part of the device specification.

Reading Device Specifications

By working through the following pages you will begin to gain an understanding of how the various parameters are defined in a device specification and used within the test program. Review the device specification for the 256 x 4 static RAM. On the pages immediately following the specification you will find a brief description of each DC test listed in the specification. Read the device specifications carefully and determine the correct values to be used within the test program.

Although the complete device specification is shown, only the DC section will be considered at this time. The AC section will be discussed in a later chapter.

256 x 4 Static Read/Write RAM Device Specifications

256 x 4 Static RAM

Logic Diagram

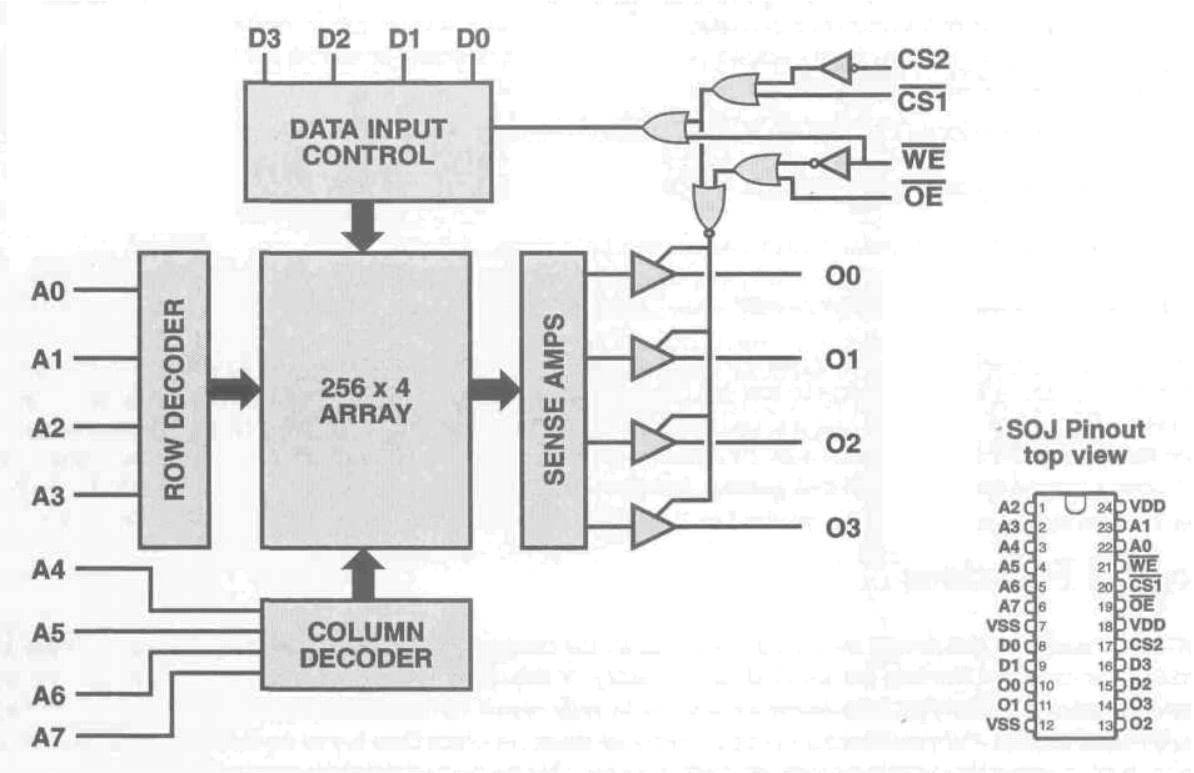


Figure 4-1

Functional Description

The 256 x 4 static RAM is a high speed CMOS 256 word by 4 bit static RAM. CMOS technology provides excellent performance at low power consumption. Flexible design makes it easy to use the 256 x 4 static RAM and expand to larger memory sizes. Very low access time makes the RAM useful in CPU caching applications and others which require fast data access.

Writing to the device is done by taking CS1/ and WE/ low and CS2 high. Data on D0-D3 is written into the memory location specified by the address on A0-A7. Outputs O0-O3 are in a high impedance state during the write cycle.

The RAM is read by making CS1/ and OE/ low and CS2 and WE/ high. Data at the address set by A0-A7 appears on outputs 00-03 within the maximum access time.

When CS1/ or OE/ are high or WE/ or CS2 are low, the output pins are in a high-impedance state.

256 x 4 RAM Specifications

Maximum Ratings

Storage Temperature	-55°C to 125°C	DC Input Voltage[1]	-0.5V to 7.0V
Ambient Temperature with Power Applied	-55°C to 125°C	Output Current into Outputs (Low)	20mA
Supply Voltage to Ground Potential (Pins 24 and 18 to Pins 7 and 12)[1]	-0.5V to 7.0V	Latch-Up Current	200mA
DC Voltage applied to Outputs in High Z State[1]	-0.5V to 7.0V		

Operating Range[2]

Range	Ambient Temperature[2]	VDD
Commercial	0°C to 70°C	5V ± 5%
Military	-55°C to 125°C	5V ± 10%

D.C. Characteristics (Over the Operating Range)

Parameter	Description	Test Conditions		Min	Max	Units
VOH	Output HIGH Voltage	VDD = Min, IOH = -5.2mA		2.4		V
VOL	Output LOW Voltage	VDD = Min, IOL = 8.0mA			0.4	V
VIH	Input HIGH Voltage			2.2	V_{DD}	V
VIL	Input LOW Voltage[1]			0.0	0.8	V
IIL, IIH	Input Load Current	Vss ≦ Vin ≦ VDD		-10	+10	µA
IOZ	Output Current (High Z)	Vss ≦ Vout ≦ VDD, Outputs Disabled		-10	+10	µA
IDD	Power Supply Current	VDD = Max Iout = 0mA $f = f_{MAX}$	Commercial		120	mA
			Military		150	mA

Capacitance[3]

Parameters	Description	Test Conditions	Max	Units
Cin	Input Capacitance	Ta = 25oC, f = 1MHz, VDD = 5.0V	8	pF
Cout	Output Capacitance	same as above	8	pF

Logic Table[4]

Inputs					Outputs	Mode
$\overline{OE}$	$\overline{CS1}$	CS2	$\overline{WE}$	D0-D3	O0-O3	
X	H	X	X	X	Z	Not Selected
X	X	L	X	X	Z	Not Selected
L	L	H	H	X	L/H	Read Stored Data
X	L	H	L	L	Z	Write "0"
X	L	H	L	H	Z	Write "1"
H	L	H	H	X	Z	Output Disabled

256 x 4 Static RAM A.C. Characteristics

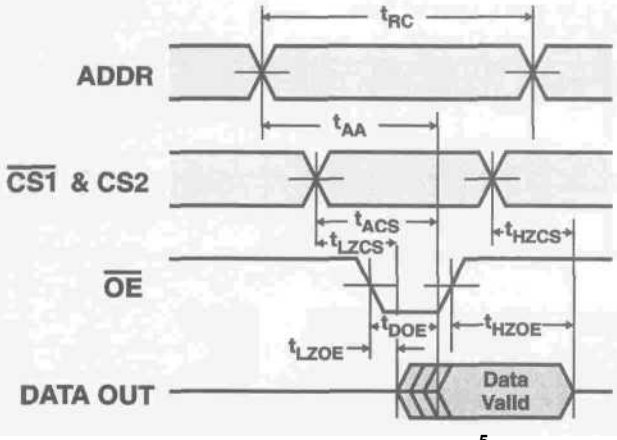


Figure 4-2 Read Timing[5]

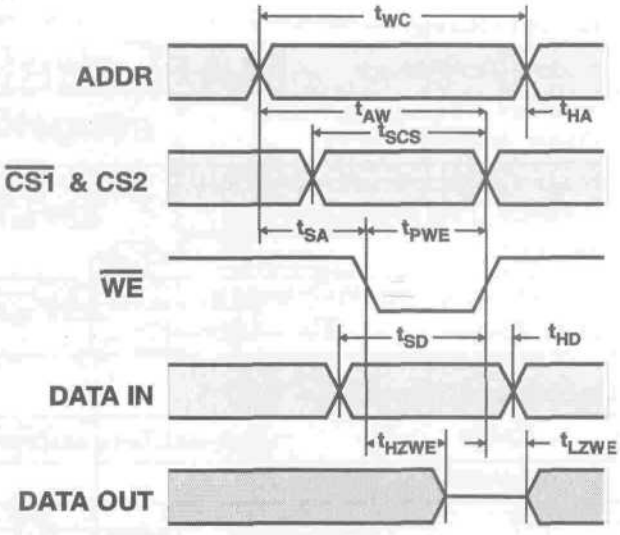


Figure 4-3 Write Timing[5]

READ Cycle				
Parameter	**Description**	**Min**	**Max**	**Units**
t_{RC}	Read Cycle time	15		nsec
t_{AA}	Address to Data Valid		15	nsec
t_{ACS}	Chip Select to Data Valid		10	nsec
t_{DOE}	OE Low to Data Valid		10	nsec
t_{HZCS}[6]	Chip Select to High Z		8	nsec
t_{HZOE}	OE High to High Z		8	nsec
t_{LZCS}[6]	Chip Select to Low Z	2		nsec
t_{LZOE}	OE Low to Low Z	2		nsec
WRITE Cycle				
Parameter	**Description**	**Min**	**Max**	**Units**
t_{WC}	Write Cycle time	15		nsec
t_{HZWE}	WE Low to High Z		8	nsec
t_{LZWE}	WE High to Low Z	2		nsec
t_{PWE}	WE Pulse Width	11		nsec
t_{SD}	Data Set-up to Write End	11		nsec
t_{HD}	Data Hold from Write End	1		nsec
t_{SA}	Address Set-up to Write Start	2		nsec
t_{HA}	Address Hold from Write End	2		nsec
t_{SCS}	CS Low to Write End	11		nsec
t_{AW}	Address Set-up to Write End	13		nsec

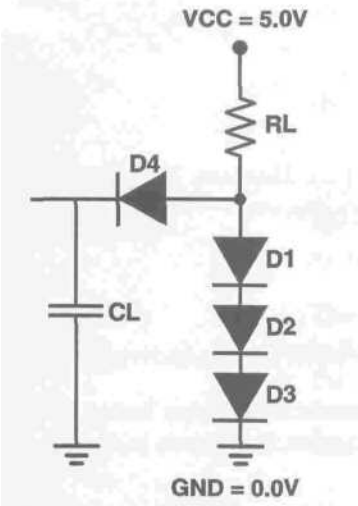


Figure 4-4 AC Test Load

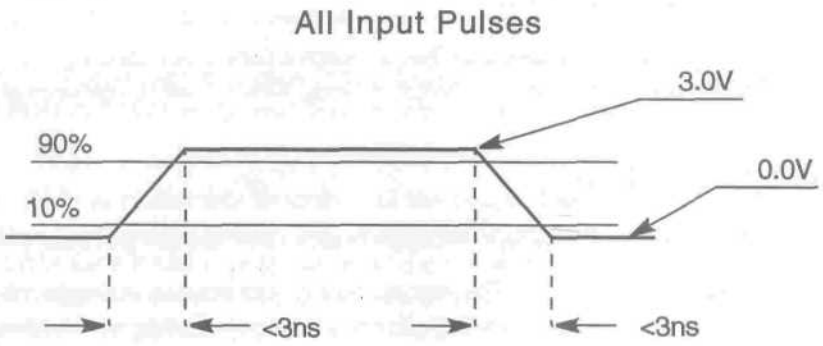


Figure 4-5 Input Waveform

Note: The output test load shown in Figure 4-4 is required on all output pins when performing AC timing measurements. This type of load is normally connected and disconnected using a mechanical relay located on the load board.

Note: Input levels should be set according to Figure 4-5 for the AC tests. Set VIL = 0.0 and VTH = 3.0.

Footnotes:

1. VILmin = -3.0V for pulse durations < 20nsec.
2. TA is the "instant on" temperature.
3. Tested initially and after any design or process change that may affect these parameters.
4. H = Logic 1, L = Logic 0, X = Don't Care, High Z = High Impedance.
5. Measurements are referenced to 1.5V Output and Input conditions.
6. For any single device, ^{t}HZCS is always less than ^{t}LZCS.

Interpreting the Device Specification

Functional Description

The device specification normally begins with a functional description and block diagram. This section gives a general overview of the device—what it does, how to use it, features and benefits, etc.

DC Specifications

The DC section defines the operating parameters of the device. Lets examine each section individually.

Maximum Ratings — The extremes of temperatures, voltages and currents which may be applied to the device. If any Maximum Rating is violated the device may cease to function and the manufacturers warranty would be voided.

Operating Range — The operational range of the VDD voltage and also the operational temperature range. Make certain to notice that the specification defines both commercial and military parameters, indicating that the device may be purchased for either application. Also notice that the VDD range for the commercial grade device is 5V ± 5%. That makes VDDMIN, which is the lowest supply voltage at which the device is guaranteed to still operate correctly, 5V - 5% or 4.75V and VDDMAX, which is the highest supply voltage at which the device is guaranteed to still operate correctly, 5V + 5% or 5.25V. The VDD range for the military grade device is 5V ± 10%, so VDDMIN is 5V - 10% = 4.5V and VDDMAX is 5V + 10% = 5.5V

D.C. Characteristics — The values for the various parameters listed. This section contains the parameter names (VOH/VOL/etc.), a brief description of the parameter, the test conditions which are applied when testing the parameter, the Minimum and Maximum guaranteed operating points, and finally the units of measurement that applies to each parameter. The Minimum and Maximum guaranteed operating points are used to establish the pass/fail limits used with in test program.

There are several DC parameters that relate to the DUT inputs. VIH (Voltage In High) is the minimum guaranteed voltage that can be applied to an input and still be recognized as a logic 1 by the DUT circuitry. VIL (Voltage In Low) is the maximum guaranteed voltage value that can be applied to an input and still be recognized as a logic 0 by the DUT circuitry. IIH (Input Leakage High) is the maximum amount of current that is allowed to flow into an input pin when a logic high voltage is applied to the pin . IIL (Input Leakage Low) is the maximum amount of current that is allowed to flow out of an input pin when a low voltage is applied onto the pin.

There are several DC parameters that relate to the DUT outputs. VOH (Voltage Out High) is the minimum guaranteed voltage that will be produced by the output when driving out a logic 1. VOL (Voltage Out Low) is the maximum guaranteed voltage that will be produced by the output when driving out a logic 0. IOH (Current Out High) is the amount of current that an output must source when driving a logic 1. The output must be capable of supplying the specified IOH current while maintaining the correct VOH voltage. IOL (Current Out Low) is the amount of current that an output must sink when driving a logic 0. The output must be capable of accepting the specified IOL current while maintaining the correct VOL voltage

There are also DC parameters that relate to the DUT power pin(s). ICC which is the current consumed by the circuitry of a TTL device or IDD which is the current consumed by the circuitry of a MOS device.

Capacitance	The amount of capacitance of each input and output pin. This information is useful to system design engineers but capacitance is not tested on digital test systems.
Logic Table	The Logic Table or truth table describes all the proper logic states of a device. The logic table describes the functionality of a device in general terms. As an example the logic table for a RAM vice shows how data is written into the RAM and read from the RAM. It does not indicate the data written into and read from the RAM; this depends on the application in which the RAM is used

.

AC Specifications

The AC specifications consist of timing diagrams and the individual parameter values that are associated with the timing diagrams. The AC section defines the type of AC load circuit which is to be used when testing the output timing parameters and it also defines the waveform characteristics of input signals. The AC specifications are used to determine how the functional timing will be structured within the test program and it is also used to define which AC timing parameters will be tested.

Device Specifications and Test Conditions

The following is a brief description of each DC test listed in the device specification. Read the description of each test and determine the correct values to be used for device testing. Use the **COMMERCIAL** VDD specifications when determining the VDD values.

Note: This exercise may seem difficult because you do not yet have a detailed understanding of each test parameter and the method used to verify that parameter. The goal of this exercise is to begin the process of understanding what information is commonly found in a device specification and how that information relates to the test program.

VOL/VOH (voltage out low/high parameters)

The parameter values shown below are commonly used, and referred to as standard for a given technology. As these values may vary for any given device, the parameter values defined in the device specification are always the values to be used in test program development. The VOL/IOL and VOH/IOH specifications are interrelated. When testing these parameters, both current and voltage must be tested to the specification simultaneously unless otherwise noted. The device specification may also specify the VDD value to be used during this test. To correctly perform the VOL/VOH test, the DUT must be preconditioned to insure that the outputs are in the correct logic state.

Standard TTL VOL = 0.4V
Standard TTL VOH = 2.4V
Standard CMOS VOL = 0.1V (GND +.1) VDD = 5.0 (no current load)
Standard CMOS VOH = 4.9V (VDD -.1) VDD = 5.0 (no current load)

IOL/IOH (current out low/high parameters)

These parameters define the amount of current that an output can sink when driving a logic 0, or the amount of current an output can source when driving a logic 1. The IOL/VOL and IOH /VOH specifications are interrelated. When testing these parameters both current and voltage must be tested to the specification simultaneously. The device specification may also specify the VDD value to be used during this test. To correctly perform the IOL/IOH test, the DUT must be preconditioned to insure that the outputs are in the correct logic state.

Note: It is possible to have CMOS inputs/outputs with TTL compatible levels, verify and enter the exact values from the specification.

Defining the Conditions for the VOH/IOH Test (voltage out high)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development The answers to the following questions can be found in the in Chapter B (Answers) near the end of the book, but don't peek!

Parameter	Description	Test Conditions	Min	Max	Units
VOH	Output HIGH Voltage	VDD = Min, IOH = -5.2mA	2.4		V

1. VDD voltage will be set to_____volts.

2. IOH current will be forced. The correct value of IOH is_____.

3. Voltage will be measured during the test. The test will fail if the measured voltage is less than _____volts.

Defining the Conditions for the VOL/IOL Test (voltage out low)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development.

Parameter	Description	Test Conditions	Min	Max	Units
VOL	Output LOW Voltage	VDD = Min, IOL = 8.0mA		0.4	V

1. VDD voltage will be set to_____volts.

2. IOL current will be forced. The correct value of IOL is_____.

3. Voltage will be measured during the test. The test will fail if the measured voltage is greater than_____volts.

VIL/VIH (voltage input low/high parameters)

The parameter values shown below are commonly used, and referred to as standard for a given technology. These values may vary for any given device, the parameter values defined in the device specification are always the values to be used in test program development.

Standard TTL VIL = 0.8V
Standard TTL VIH = 2.0V
Standard CMOS VIL = 1.5V (VDD * 0.3) VDD = 5.0
Standard CMOS VIH = 3.5V (VDD * 0.7) VDD = 5.0

Note: It is possible to have CMOS inputs and outputs with TTL compatible levels.

Defining the Conditions for the VIL/VIH Test (voltage in low & high)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development.

Note: The VIH/VIL parameters must be verified by executing a functional test twice. Once with VDD set to the minimum value and then again with VDD set to the maximum value.

Parameter	Description	Test Conditions	Min	Max	Units
VIH	Input HIGH Voltage		2.2	V_{DD}	V
VIL	Input LOW Voltage[1]		0.0	0.8	V

1. VDD voltage will be set to _____ and_____ volts.

2. The VIH (input high level) will be set to _____volts.

3. The VIL (input low level) will be set to _____ volts.

IIL/IIH (current input low/high parameters)

Input current tests low and high. To perform this test a low voltage is forced onto each input and the resulting current is measured. The test is then repeated, this time a high voltage is forced onto each input and the resulting current is measured. VDD is normally set to the maximum VDD value, the voltage applied to the inputs during testing is dependent upon the design and technology. Be sure to verify the values for VDD, Vin (voltage applied) and IIL/IIH (current limits).

Defining the Conditions for the IIL/IIH Test (input leakage test)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development.

Note: VDD is normally set to maximum. When forcing a high voltage on the input, the voltage applied is normally VDD maximum. When forcing a low voltage on the input, the voltage applied is normally VSS or zero volts.

Parameter	Description	Test Conditions	Min	Max	Units
IIL, IIH	Input Load Current	Vss ≦ Vin ≦ VDD	-10	+10	µA

1. VDD voltage will be set to_____.

2. When forcing a high voltage on the inputs Vin will be set to_____volts.

3. Current will be measured during the test. The test will fail if the measured current is greater than

   _____

4. When forcing a low voltage on the inputs Vin will be set to_____volts.

5. Current will be measured during the test. The test will fail if the measured current is less than

   _____.

IOZ (current output high Z parameter)

Output high impedance leakage current. First the output under test is preconditioned to the off or high-Z state. Current is then measured while forcing voltage (usually VDD maximum and ground). From the device specification find the VDD value, the IOZ current limits and the forcing voltages.

Defining the Conditions for the IOZ Test (output currents - high impedance)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development.

Note: VDD is normally set to maximum. When forcing a high voltage on the outputs, the voltage applied is normally VDD maximum. When forcing a low voltage on the outputs, the voltage applied is normally VSS or zero volts.

Parameter	Description	Test Conditions	Min	Max	Units
IOZ	Output Current (High Z)	Vss ≦ Vout ≦ VDD, Outputs Disabled	-10	+10	µA

1. VDD voltage will be set to _____.

2. When forcing a high voltage on the outputs Vout will be set to_____volts.

3. Current will be measured during the test. The test will fail if the measured current is greater than_____.

4. When forcing a low voltage on the outputs Vout will be set to_____volts.

5. Current will be measured during the test. The test will fail if the measured current is less than_____.

IDD Dynamic Current

The dynamic IDD test requires a continuous vector pattern or a preconditioning sequence which leaves the device in an active state. A continuous loop or subroutine is often created specifically for the IDD dynamic test. The device specification may define this sequence, if not, some experimenting may be required to locate or develop the proper vector sequence. Make certain to verify the correct input levels and VDD level to be used during testing.

Defining the Conditions for the IDD Test (total supply current)

The following information has been copied from the device specification. Read the device specification and determine the correct values to be used for test program development.

Parameter	Description	Test Conditions		Min	Max	Units
IDD	Power Supply Current	VDD = Max f = 66Mhz Iout = 0mA,	Commercial		120	mA
			Military		150	mA

1. VDD voltage will be set to_____.

2. The output current loading (Iout) will be set to_____.

3. This test requires the execution of a functional test loop during the measurement of the IDD current. The frequency of the functional test will be_____.

4. During this test the IDD current will be measured. The test will fail if the measured current is greater than_____.

Device Specifications Review

1. Standard TTL input levels are:
   a) VIL = 1.5V (VDD * 0.3) VDD = 5.0
      VIH = 3.5V (VDD * 0.7) VDD = 5.0
   b) VIL = 0.4V
      VIH = 2.0V
   c) VIL = 0.8V
      VIH = 2.4V
   d) None of the above

2. Which DC test forces voltage onto input pins and then measures the current flow?

3. From the sample device specification in chapter four, find the commercial IDD current limit for the 256 x 4 device. The maximum current flow limit is:

4. When an output is in the logic 0 state, what test is performed to guarantee the correct output voltage level?
   a) VIL
   b) VOL
   c) IOZL
   d) None of the above

5. From the sample 256 x 4 device specification, find the commercial operating temperature range:

6. In the sample 256 x 4 RAM device specification, what current limit values would be used in the IIL/IIH test?

7. In the sample 256 x 4 device specification, the values for IOL and IOH are equal.
   a) True
   b) False

Chapter 5 Opens and Shorts - PMU Method

Objectives

This section explains:

- The purpose of testing for opens and shorts.
- Serial/Static DC test method.
- Advantages and disadvantages of the test method.

Why Test for Opens and Shorts?

The opens/shorts test (also called continuity or contact test) verifies that, during a device test, electrical contact is made to all signal pins on the DUT and that no signal pin is shorted to another signal pin or power/ground.

Device cost is directly related to how long it takes to test each device. One of the best ways to reduce average test time per device is to reject bad devices as soon as possible. The opens/shorts test determines very quickly whether a device has shorted pins, missing bond wires, a pin damaged from static electricity, a manufacturing defect, etc.

The opens/shorts test can also point to test system related problems such as a wafer test probe card or a device test socket which is not making correct contact.

Opens and Shorts Serial Static Method

Test conditions for an opens/shorts test are not normally defined in the device specification or data book specification, but there are standard values which apply to most standard device types. These standard values are presented in the guidelines here.

To test for opens/shorts, first ground all device pins including power and ground pins. Next, connect the PMU to a single device pin and force a current which will forward bias one of the protection diodes. (See Figure 5-1 and Figure 5-2.) A negative current will forward bias the diode to ground; a positive current will forward bias the diode to VDD. A current in the range of 100µA to 500µA should be adequate. Once forward biased, the voltage drop across the protection structure can be sensed (typically 0.65 volt). The sensed voltage may change between various technologies and designs, but 0.65V is good for most silicon based devices.

Since the PMU will be programmed to force current, a voltage clamp must be programmed to limit the voltage produced when an open pin is tested. A typical clamp value for the opens/shorts test is 3V. When an open pin is tested, the measured result will be the clamp voltage (3V).

The advantage of the Serial/Static opens and shorts test is that it produces DC measurements. When a failure occurs, the exact measured voltage may be examined from the datalogged results. The voltage value clearly indicates whether the failure was caused by a shorted condition or an open condition. The disadvantage is the time required to make the individual DC measurement on each pin.

This can also be performed as a functional test. See Opens and Shorts - Functional Method in Chapter 7

Opens/Shorts Test
Serial/Static Method, VDD Diode

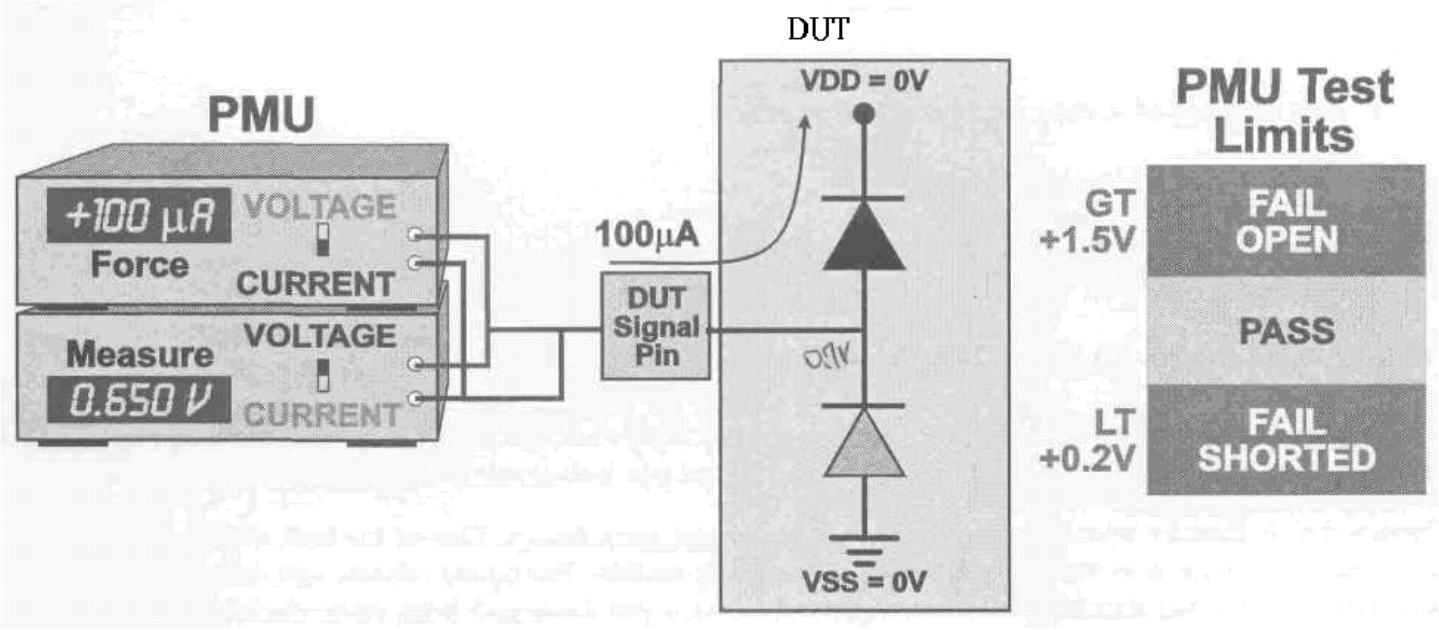


Figure 5-1

Ground all pins (including VDD).
Set Voltage Clamp 3.0 volts
Using PMU, force +100µA, one pin at a time.
Measure resultant voltage.
Fails test (open) if voltage measured is greater than +1.5V.
Fails test (shorted) if voltage measured is less than +0.2V.

To test the upper diode, which is connected to VDD, use the PMU to force a positive current of approximately +100µA. Set the upper test limit of the PMU. to fail if the measured result is greater than 1.5V to detected an open. Set the lower PMU limit to fail if the measured result is less than 0.2V to detect a short. This test method is used to test signal pins (inputs and outputs) but not power pins such as VDD or VSS.

Opens/Shorts Test
Serial/Static Method, VSS Diode

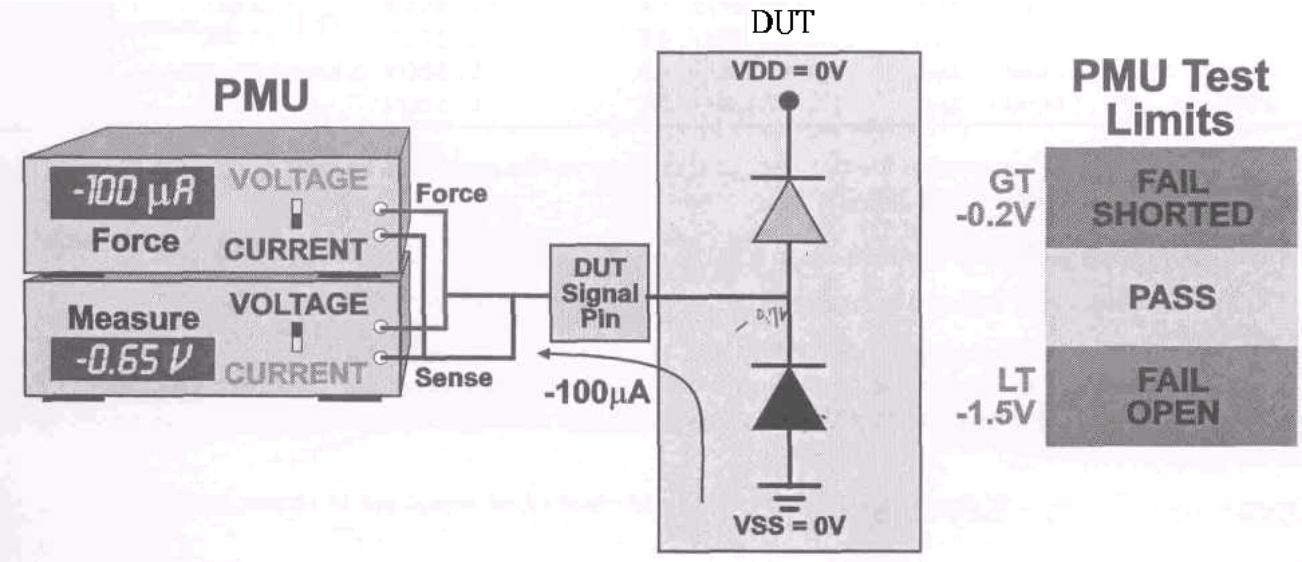


Figure 5-2

Ground all pins (including VDD).
Set Voltage Clamp-3.0 volts
Using PMU, Force -100µA, one pin at a time.
Measure resultant voltage.
Fails test (shorted) if voltage measured is greater than -0.2V.
Fails test (open) if voltage measured is less than -1.5V.

To tset the lower diode, which is connected to VSS, use the PMU to force a negative current of approximately -100µA. Set the lower test limit of the PMU to fail if the measured result is less than -1.5V to detect an open. Set the upper PMU limit to fail if the measured result is greater than -0.2V to detect a short. This test method is used to test signal pins (inputs and outputs) but not power pins such as VDD or VSS.

Power and ground pins may also be tested for an open condition, but their structure is different from that of a signal pin. Test the power pins, observe the measured value on a good device, and set the test limits accordingly.

Datalog of: Opens and Shorts using the PMU

Pin	Force/rng		Meas/rng		Min	Max	Result
Pin1	-0.100mA/	2mA	-641mV/	8V	-1.500V	-200mV	PASS
Pin2	-0.100mA/	2mA	- 3.0V/	8V	-1.500V	-200mV	FAIL
Pin3	-0.100mA/	2mA	-633mV/	8V	-1.500V	-200mV	PASS
Pin4	-0.100mA/	2mA	-004mV/	8V	-1.500V	-200mV	FAIL

To begin trouble shooting, enable the datalogger and observe the measured result. In this case the measured result will be one of three possibilities:

1. The result will show the measured voltage of a working diode, approximately 0.65V.
2. The results will indicate a shorted condition, approximately 0V.
3. The result will indicate an open condition, which will be limited by the clamp voltage. If no clamp voltage is set, the measured result will be the highest voltage possible, limited only by the range of the test hardware. Not clamping the voltage can further damage the DUT.

Opens and Shorts - Key Points

Purpose: to detect open or shorted device pins and verify proper connections between the test system and the DUT

- Test one pin at a time, DC serial/static test method
- Force current, measure voltage
- Test requirements not found in device specifications
- This test is very time consuming (slow and costly)

Opens and Shorts Review

1. Device pins must be set to a certain condition for the opens and shorts test. To be properly conditioned, they are:
   a) Programmed to VDD
   b) Programmed to ground
   c) Floating
   d) None of the above

2. When performing the opens and shorts test using the PMU:
   a) A positive current is forced
   b) A negative current is forced
   c) A or B
   d) None of the above

3. In the datalog example of the opens and shorts test pin2 is failing because it is:
   a) open
   b) shorted

4. In the datalog example of the opens and shorts test pin4 is failing because it is:
   a) open
   b) shorted

5. In the datalog example of the opens and shorts test, pin 2 measures as -3.0V. This reading is caused by:
   a) The PMU voltage clamp
   b) The pin being shorted
   c) The maximum voltage range of the PMU
   d) None of the above

Chapter 6 Verifying DC Parameters

Objectives

This section explains:

- Ohm's Law
- The various types of DC tests
- How and why each DC test is performed
- Advantages and disadvantages of various test methods
- How to calculate the resistive value being tested
- How to trouble shoot each test
- Key points of each test

Basic Terms

As we begin looking at DC testing there a few basic terms that must be understood. A complete list of terminology is located in the Glossary.

Hot Switching	Occurs when a relay is opened while current is flowing through it, or when current immediately begins to flow after a relay is closed (i.e. when the two terminals of a relay are at different voltages when the relay is dosed). Opening or closing a relay while current is flowing through it may result in damage to the relay. This can be avoided with careful test programming.
Latch-up	A high current condition which exists within a CMOS device caused by applying an improper voltage level to a signal pin, VDD or ground. This condition can weaken the device or cause a catastrophic failure.

Binning

Binning is a way of categorizing or sorting the tested devices into their appropriate groupings. Two types of binning may exist within a test program, hard binning and soft binning. Hard binning controls the physical hardware which selects the actual location (rube or tray) where the DUT will be placed after testing, Soft binning controls software counters which keep track of the various pass and fail categories. The number of hard bins is limited by the external handling equipment used, soft bins are usually unlimited. An example of binning categories follows:

Bin#	Category
01	Good Device 100MHZ
02	Good Device 75MHZ
10	Opens and shorts reject
09	Gross IDD reject
08	Gross Function reject
03	75MHZ Function reject
07	Function VIL/VIH reject
06	DC VOL/VOH reject
05	Dynamic/Static IDD reject
04	IIL IIH/ IOZ Leakage reject)

The number of individual bins is often determined by the equipment used for device handling during testing. Auto-handling equipment must have at least two bins (good and bad) but may have many bins as the example above shows.

The test program must provide the proper information to the auto-handler via the hardware interface. This is accomplished by setting a bin light from within the test program. When the auto-handler receives the end of test signal it reads the bin light and deposits the tested device into the appropriate tray or tube.

Program Flow

The Program flow and it's relationship to DC testing is important. Many DC tests require preconditioning, meaning the device logic must be set into a specific logic condition before the DC measurement can be made. Therefore, before the DC tests are performed, the functional tests must be verified. If the device can not function properly the preconditioning will fail and the results of the DC tests will be meaningless. The following Test Flow Diagram illustrates a typical program flow, notice that the Gross Functional Test is performed before the DC tests. This insures that all of the device functions are operating properly and any preconditioning required for the DC measurements will be successful.

The Program flow is most important to the production test program. There are a number of items to consider when planning the test flow, but test throughput is generally the most important. Other items to consider may be: what amount of information needs to be gathered during test and will the test procedure include speed grading or multiple pass binning (more than one good bin). Develop a flow diagram which includes pass/fail branching and binning; Make sure the flow will satisfy all test requirements.

Test Flow Diagram

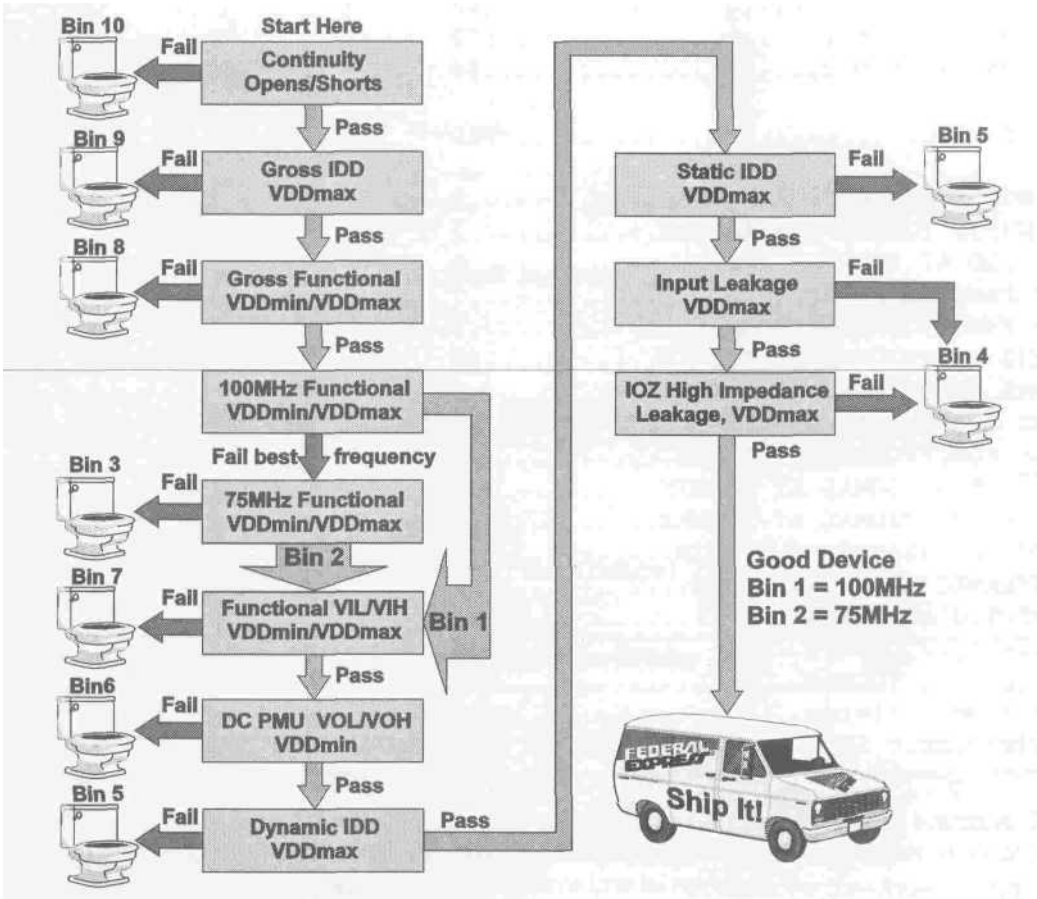


Figure 6-1

After the program has been used in production for some period of time, the test summaries should be reviewed and if necessary the program flow should be modified. Tests that fail most frequently should be executed first, tests that seldom fail should be executed last. See Figure 6-1

Test Summary

The test summary provides statistical information which indicates the results of testing. The summary should indicate the total tested, total passing/failing and the number of failures in each test category. The test summary can provide valuable insight into yield issues, so include as much information as possible in the test summary. The ability to extract a partial summary (without clearing the counters) is often desirable during production testing. See Figure 6-2.

```
********************  TEST SUMMARY  ************************

  TEST TEMP: 25C    LOT NUMBER: 285-8089    DATE:          10/12/98

                    TOTAL UNITS TESTED:  TOTAL GOOD    UNITS:
                                100                     80

*********************************************************************

                                                TOTAL UNITS      % OF TOTAL
TOTAL TESTED ..... .....100
TOTAL PASSED .BIN 1 ..... 30                        30
TOTAL PASSED.BIN 2 ..... 50                         50

TOTAL FAILED ..... 20                               20

CONTINUITY (SHORTS) FAILURES.....1                  1
CONTINUITY (OPENS)  FAILURES.....2                  2
GROSS IDD AT VDDMAX.....0                           0
GROSS FUNCTIONAL AT VDDMIN.....7                    7
GROSS FUNCTIONAL AT VDDMAX.....0                    0
100 MHZ FUNCTIONAL AT VDDMIN..... 50
100 MHZ FUNCTIONAL AT VDDMAX .....0
75 MHZ FUNCTIONAL AT VDDMIN.....3                   3
75 MHZ FUNCTIONAL AT VDDMAX.....0                   0
VIL/VIH FUNCTIONAL AT VDDMIN.....1                  1
VIL/VIH FUNCTIONAL AT VDDMAX.....0                  0
VOL/VOH DC STATIC AT VDDMIN .....3                  3
IDD DYNAMIC AT VDDMAX.....1                         1
IDD STATIC AT VDDMAX.....2                          2
IIL/IIH AT VDDMAX .....0                            0
IOZL/IOZH AT VDDMAX.....0
Power Supply Alarms .....0
Average Static IDD..... 26.8uA
```

Figure 6-2

DC Tests and the Hidden Resistance

Most DC parameters are verified by forcing current and setting a voltage limit or by forcing voltage and setting a current limit. What is actually being verified is the amount of resistivity the silicon offers.

When a test is performed by forcing a current and measuring a voltage, the voltage value is produced by the resistance of the silicon. In a similar way, when a voltage is forced and a current is measured it is the amount of resistance which determines the current.

A semiconductor is designed to precise specifications. These specifications provide details on how conductive or resistive the various paths through the silicon must be. The amount of conductivity may also change depending on the function the device is performing. The device may be fully-conducting, semi-conducting or non-conducting.

Ohm's Law is used to calculate the tested resistance value for each DC parameter. To verify or debug a DC test, a resistor may be substituted for the DUT to eliminate the possibility of a defective DUT. The value of the substituted resistance is calculated from the forced and expected measurement values of voltage and current. The pin being tested is disconnected and the resistor placed between the test socket for that pin and VDD/ VCC or VSS/ground.

Ohm's Law

R = resistance E = voltage I = current

To Calculate Resistance:

R = E / I resistance (voltage divided by current)

To Calculate Voltage:

E = I * R voltage (current multiplied by resistance)

To Calculate Current:

I = E / R current voltage divided by resistance)

Parameter	Description	Test Conditions	Min	Max	Units
VOL	Output LOW Voltage	VDD = Mm, IOL = 8.0mA		0.4	V

A VOL specification can be used as an example to calculate resistance: VOL = 0.4V, IOL = 8.0mA. This specification states that an output gate driving a logic 0 must produce a current of 8.0mA at a voltage no greater than 0.4V. Given this information, we can use Ohm's law to calculate the maximum output resistance that the device pin can have and still be within the design specification. The formula below indicates that this maximum resistance is 50Ω When the device was designed, the silicon design specification stated that the output could not have more than 50ΩThat information, however, does not appear in the device specification; instead we see specifications for voltage and current values.

Example:

R = E / I E = 0.4 (VOL) I = .008 (IOL)

R = .4 .008

R = 50 Ω

NOTE: Remember that in many cases a resistor can be used in place of the DUT to verify if a test is working correctly. This can be a very effective debug tool. One of the main goals of debugging is to make sure the only errors are those of the DUT and not test program or load board errors. Using a resistor guarantees a good "DUT-equivalent."

VOH/IOH

VOH represents the minimum voltage (V) produced by an output (O) when the output is in the logic 1 (High) state. IOH represents the current sourcing capabilities (I) of an output (O) when the output is in the logic 1 (High) state. The following table illustrates the VOH/IOH specification for the 256 x 4 Static RAM:

Parameter	Description	Test Conditions	Min	Max	Units
VOH	Output HIGH Voltage	VDD = 4.75V, IOH = -5.2mA	2.4		V

Why Test for VOH/IOH?

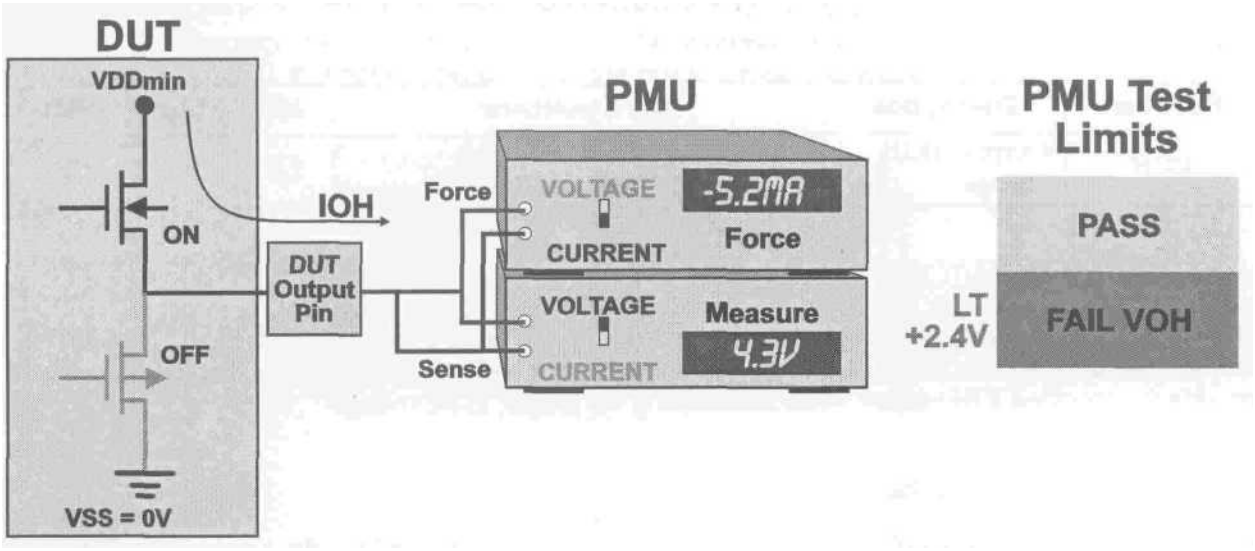
The VOH/IOH test measures the resistance of an output pin when the output is in the logic 1 state. This test insures that the resistance of the output meets the design parameters and guarantees that the output will provide the specified IOH current while maintaining the proper VOH voltage.

VOH/IOH—Serial/Static Test Method

These parameters may be verified either statically or dynamically (dynamic VOH is discussed in chapter seven). To perform a static (DC) test, the device is preconditioned to set the output(s) into the logic 1 state, the DC measurement system (PMU) is connected to the pin under test, the IOH current is forced and the resultant voltage is measured and compared to the VOH specification. If the measured voltage is less than the VOH limit, the test fails. This process is repeated until all pins have been verified in the high output state. This test may require running more than one preconditioning sequence to set all outputs to the logic 1 state. See Figure 6-3.

Notes: 1) VDDMIN is the worst case for this test. 2) IOH is a negative current. 3) A voltage clamp must be set.

Output Voltage Test
VOH/IOH



Apply VDDmin.
Set Voltage Clamp
Precondition output to logic 1 (output high).
Using PMU, force IOH current per specification.
Wait 1 to 5 msec (Set PMU delay).
Measure resultant voltage.
Fails VOH if measured voltage is less than +2.4V.

Figure 6-3

Datalog of: VOH/IOH serial/static test using the PMU

Pin	Force/rng		Meas/rng	Min	Max	Result
PIN1	- 5.2mA/	10mA	4.30V/ 8V	2.40V		PASS
PIN2	- 2.0mA/	10mA	2.34V/ 8V	2.40V		FAIL
PIN3	-5.2mA/	10mA	3.96V/ 8V	2.40V		PASS
PIN4	-5.2mA/	10mA	3.95V/ 8V	2.40V		PASS
PIN5	- 8.0mA/	10mA	3.85V/ 8V	2.40V		PASS
PIN6	- 8.0mA/	10mA	-0.782/ 8V	2.40V		FAIL

VOH/IOH Resistance Calculation

The VOH test measures how well the output conducts current when it is in the logic 1 state. Another way of looking at this is to consider the amount of resistance the output offers when in the logic 1 state. The device

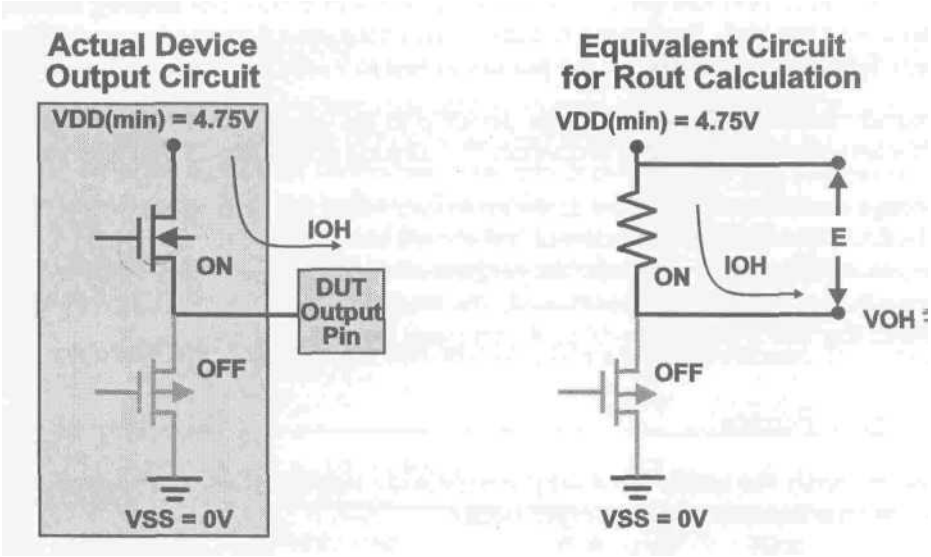
specification defines the minimum allowable voltage that the output can produce (VOH) at a specific value of current (IOH).

Using Ohm's law, the maximum allowable resistance value can be found.

Ohm's law is stated as: R = E / I
R is the resistance that is being tested for
E is the voltage across the resistance (VDD - VOH)
I is the current (IOH)

In the example shown in Figure 6-4, the maximum resistance that the output can offer and still pass the test is 452Ω A resistor having a value of 452Ωor less can be substituted for the DUT and the test will pass. Substituting a resistor for the DUT pin being tested is a good way to debug a VOH test problem.

VOH/IOH Resistance Calculation



Specification
IOH(min) = -5.2mA
VOH(min) = 2.4V

E = VDD - VOH
R = E / I
E= 4.75 - 2.4 = 2.35
R = 2.35 / 0.0052
R = 452Ω(maximum)

Figure 6-4

VOH/IOH—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the measured results. The result will be one of three possibilities:

1. The VOH voltage is correct and the test result is a pass.
2. The VOH voltage is less than the minimum limit and the test result is a fail with the output in the logic 1 state.
3. The VOH voltage is less than the minimum limit and the test result is a fail with the output in the wrong logic state (logic 0). In this case the test system is attempting to force a negative current into a device pin that is in the wrong logic state. The device pin appears as a very high resistance which causes the PMU to produce negative voltage that will go below ground. When this occurs, the lower protection diode turns on and produces a voltage of approximately -0.7 volts.

When a failure occurs, observe the measured voltage to determine the cause of the failure. Is the failure just under the VOH limit as described in item 2 above or does the output appear to be in the wrong logic state as described in item 3?

If the failing measurement is just under the VOH limit, the DUT is probably defective. The datalog above shows this type of failure on pin 2. It can be seen that the device was in the correct logic state at the time the measurement was made, therefore the preconditioning was successful. The datalog reading indicates that the output resistance was too high. Resistance within the test hardware fixture can cause the output resistance to appear too high. Substitute a resistor for the pin under test to verify test system accuracy.

The failing measurement may indicate that the device is in the wrong logic state, which would most likely be caused by an incorrect preconditioning sequence. The datalog above shows this type of failure on pin 6.

All test vector sequences should be tested in the gross functional test, including the preconditioning sequences used within the DC tests. The gross functional test should occur before the DC tests within the program flow. Therefore, the preconditioning vectors should perform correctly when used to setup a device output for a DC test. If the device is not properly preconditioned, you must resolve this issue. Once the output is set to the proper logic state, the VOH/IOH test will probably pass.

VOH/IOH—Key Points

- Purpose: to verify the ability of an output to provide current (IOH) at a specified voltage (VOH)—this test verifies the resistance of the output buffer
- DC static test uses PMU to force current and measure voltage
- Test requires outputs to be preconditioned to logic 1
- Test limits defined in device specifications
- VDDMIN is worst case test condition

VOL/IOL

VOL represents the maximum voltage (V) produced by an output (O) when the output is in the logic 0 (Low) state. IOL represents the current sinking capabilities (I) of an output (O) when the output is in the logic 0 (Low) state. The table below shows the VOL specification for the 256 x 4 Static RAM from chapter 2 XREF:

Parameter	**Description**	Test **Conditions**	**Min**	**Max**	Units
VOL	Output LOW Voltage	VDD = 4.75V, IOL = 8.0mA		0.4	V

Why Test for VOL/IOL?

The VOL/IOL test measures the resistance of an output pin when the output is in the logic 0 state. This test insures that the resistance of the output meets the design parameters and guarantees that the output will provide the specified IOL current without exceeding the VOL voltage. In other words, the device output pins must sink at least a specified minimum amount of current and stay in the correct logic state.

VOL/IOL—Serial/Static Test Method

The VOL/IOL parameters may be verified either statically or dynamically (dynamic VOL is discussed in the next chapter.). To perform a static (DC) test, the device is preconditioned to set the output(s) to the logic 0 state. The DC measurement system (PMU) is then connected to the pin under test, the IOL current is forced and the resultant voltage is measured and compared to the VOL specification. If the measured voltage is greater than the VOL limit the test fails. This process is repeated for each pin until all pins have been verified in the low output state. This test may require running more than one preconditioning sequence to set all outputs to the logic 0 state. See Figure 6-5.

Notes: 1) VDDMIN is the worst case condition for this test. 2) IOL is a positive current. 3) A voltage clamp must be set.

Datalog of: VOL/IOL serial/static test using the PMU

Pin	Force/rng	Meas/rng	Min	Max	Result
PIN1	12.0mA/ 20mA	130mV/ 8V		400mV	PASS
PIN2	12.0mA/ 20mA	421mV/ 8V		400mV	FAIL
PIN3	4.0mA/ 10mA	125mV/ 8V		400mV	PASS
PIN4	4.0mA/ 10mA	90mV/ 8V		400mV	PASS
PIN5	8.0mA/ 10mA	205mV/ 8V		400mV	PASS
PIN6	8.0mA/ 10mA	5.52V/ 8V		400mV	FAIL

VOL/IOL Resistance Calculation

The VOL test measures how well a DUT output conducts current when it is in the logic 0 state. Another way of looking at this is to consider the amount of resistance the output offers when in the logic 0 state. The device specification defines the maximum allowable voltage that the output can produce (VOL) at a specific value of current (IOL).

Using Ohm's law, the maximum allowable resistance value can be found:

Ohm's law is stated as: R = E / I

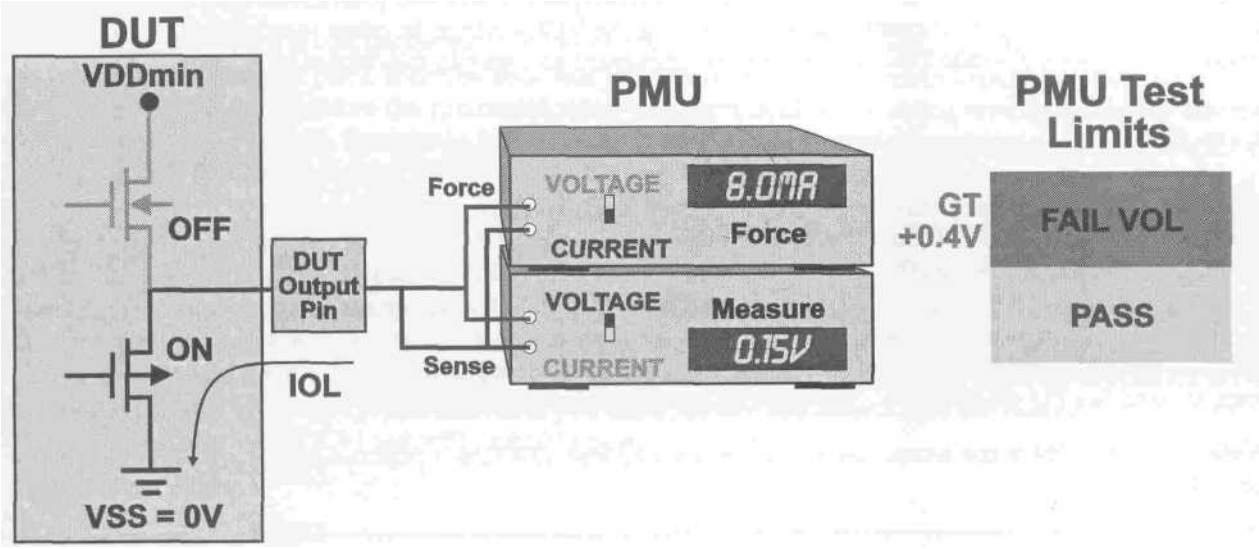
R is the resistance being tested for

E is the voltage (VOL)

I is the current (IOL)

In the example shown in Figure 6-6, the maximum resistance that the output can offer and still pass the test is 50Ω A resistor having a value of 50Ωor less could be substituted for the DUT and the test would pass. Substituting a resistor to ground for a DUT pin is a good way to debug a VOL test problem.

Output Voltage Test
VOL/IOL



Apply VDDmin.

Precondition output to logic 0 (output low).
Set Voltage Clamp
Using PMU, force IOL current per specification.
Wait 1 to 5 msec (Set PMU delay).
Measure resultant voltage.
Fails VOL if measured voltage is greater than +0.4V.

Figure 6-5

VOL/IOL—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the measured results. The result will be one of three possibilities:

1. The VOL voltage is correct and the test result is a pass.
2. The VOL voltage is over the maximum limit and the test result is a fail, but the output is in the correct logic 0 state.
3. The VOL voltage is over the maximum limit and the test result is a fail but the output is in the wrong logic state. In this case the measured value will reflect the programmed voltage clamp limit.

VOL/IOL Resistance Calculation

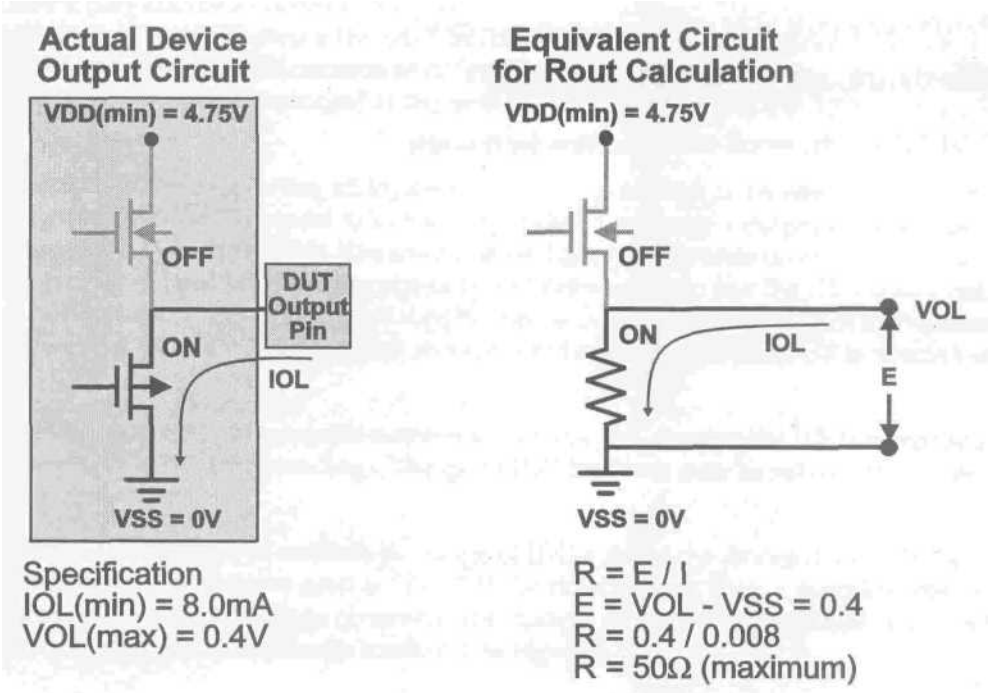


Figure 6-6

When a failure occurs, observe the measured voltage to determine the cause of the failure. Is the failure just over the VOL limit as described in item 2 above or does the output appear to be in the wrong logic state as described in item 3?

If the failing measurement is just over the VOL limit, the DUT is probably defective. The datalog above shows this type of failure on pin 2. Notice that the device is in the correct logic state when the measurement is made,

so the preconditioning was successful. The datalog reading indicates that the output resistance was too high. Resistance within the test hardware fixture can cause the output resistance to appear too high. Substituting the failing DUT pin with a resistor to ground will verify test system accuracy.

The failing measurement may indicate that the device is in the wrong logic state, which would most likely be caused by an incorrect preconditioning sequence. The above datalog shows this type of failure on pin 6.

All test vector sequences should be tested in the gross functional test, including the preconditioning sequences used within the DC tests. The gross functional test should occur before the DC tests within the program flow. Therefore, the preconditioning vectors should perform correctly when used to setup a device output for a DC test. If the device is not properly preconditioned, you must resolve this issue. Once the output is set to the proper logic state the VOL/IOL test will most likely pass.

VOL/IOL—Key Points

- Purpose: to verify the ability of an output to sink current (IOL) without exceeding the specified voltage (VOL)—this test verifies the resistance of the output buffer
- DC static test using PMU to force current and measure voltage
- Requires outputs to be preconditioned to logic 0
- Test limits defined in device specifications
- VDDMIN is the worst case condition for this test

IDD Gross Current

IDD represents the current (I) which flows drain to drain (DD) within a CMOS circuit. When testing TTL devices this test is called ICC. ICC represents the current (I) which flows collector to collector (CC) within a TTL circuit. Gross indicates that the test is made with conditions that have been relaxed from those defined in the device specification.

Why Test for Gross IDD Currents?

The purpose of the gross IDD test is to quickly determine if it is reasonable to continue to test the DUT. The gross IDD test is often performed immediately after the opens/shorts test and is the first test made after initially powering up the DUT. If the DUT is drawing excessive current, the test hardware may be damaged. When this test fails, device power is immediately shut off and the DUT is rejected.

Test Method for Gross IDD Currents

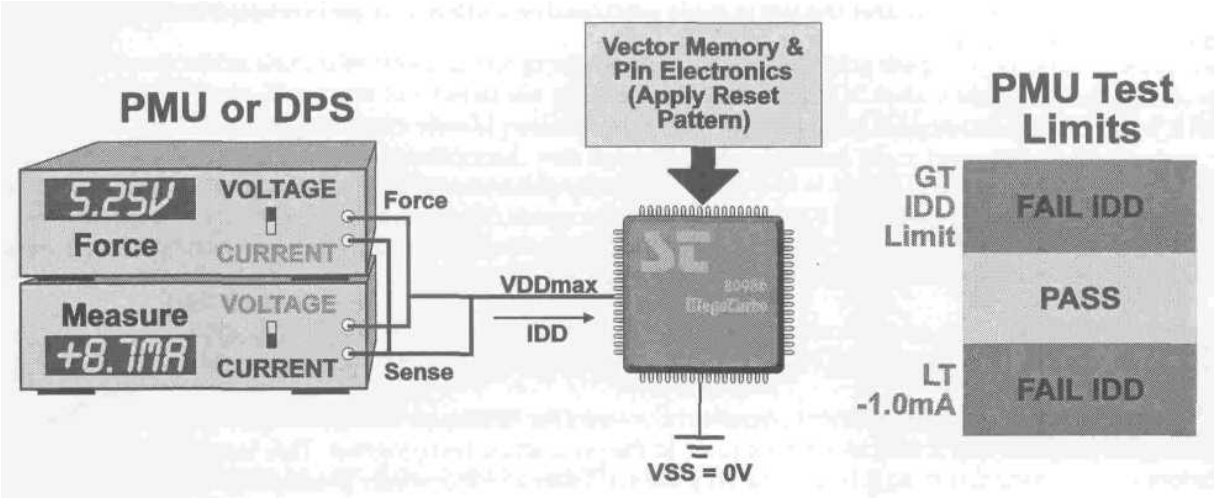
The gross IDD test is a measurement of current flow into the VDD pin. The test is often installed in the wafer probe test program and it may also be included in the production test program. This test is typically performed before any functional test, so it is not known if the DUT can be successfully preconditioned. The device must be correctly preconditioned to test for IDD currents as defined in the device specification. Since functionality (and therefore preconditioning) is not guaranteed at this point within the test flow, the IDD specification must be relaxed.

The test is normally performed by first setting all inputs low or all inputs high or by resetting the device. Typically VIL is set to zero volts and VIH is set to VDD. Outputs have no loads—output currents could increase IDD and wrongly cause the test to fail. The exact preconditioning depends on the function of the DUT and should be kept as simple as possible. The purpose of preconditioning is to put the DUT into a stable condition. Once the DUT is preconditioned the total supply current flowing into the device is measured. If the measured current exceeds the test limit the test result is considered a failure and the DUT is rejected without further testing.

To establish the gross IDD test limit, test a sample number of devices and observe the IDD current readings. Then set test limits based on the observed readings. The gross IDD test limit may be set to 2 or 3 times the IDD limit found in the device specification.

This test may also be used as a monitor to indicate the range of IDD current the device draws. The initial test specification may state that the IDD current limit is "TBD" (to be determined). If so, a sample number of devices should be tested and the IDD readings observed. The current limit for the gross IDD test should then be set two or three times higher than the average reading. See Figure 6-7.

Gross IDD Test



Using DPS or PMU, apply VDDmax.
Set Maximum Current Clamp on DPS/PMU
Set Pass/Fail Limits.
Set all inputs Low/High or Execute Reset Sequence.
Stop pattern.
Wait 5 to 10msec
Measure current flowing into VDD pin(s).
Fails IDD if measured current is outside of limits.

Figure 6-7

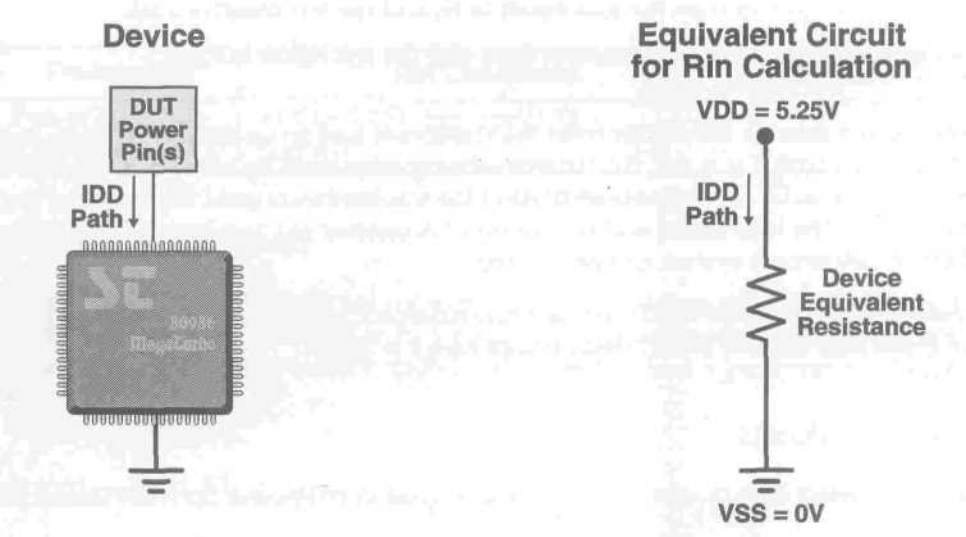
Datalog of: Gross IDD Current using the DPS

Pin	Force/rng	Meas/rng	Min	Max	Result
DPS1	5.25V/ 10V	8.7mA/ 50mA	-1mA	+45mA	PASS

Gross IDD Resistance Calculation

The gross IDD test measures the total resistance of the DUT between VDD and ground. When the VDD voltage is specified at 5.25V and the IDD limit is 45mA, Ohm's law can be used to determine the minimum allowable resistance.

Gross IDD Resistance Calculation



Specification
VDD = 5.25V
IDD max = 45mA

R = E / I
E = VDD - VSS (= 5.25 - 0.0)
R = 5.25 / 45mA
R = 117Ω

Figure 6-8

Ohm's law is stated as: R=E/I
R is the resistance being tested for
E is the voltage (VDD)
I is the current (IDD)

As seen in Figure 6-8, when a 117 ohm resistor is placed in the DUT test socket between VDD and ground, the IDD current measurement should read 45mA.

Gross IDD - Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a known good device is available (often called the standard or golden DUT), test it and observe the measured result. The result will be one of three possibilities.

1. The current is correct and the test result is a pass.
2. The current is greater than the maximum limit and the test result is a fail.
3. The current is less than the minimum limit and the test result is a fail.

When a failure occurs remove the device from the test socket and rerun the test. The result of an open socket test should be zero current; if it is not, the current is being consumed by something other than the DUT. For non-zero current with no DUT, eliminate sections of the test hardware until the source of the problem is found. (For example, remove the load board and run the test.) A resistor can also be used to replace the DUT and the accuracy of the measurement system can be verified.

Note: Zero current may not show as 0.0; for each measurement range, there is a minimum current resolution so 0.01mA may mean zero current if the measurement range is 20mA.

Gross IDD - Key Points

- Purpose: to detect high power supply currents upon initial power up (very useful at wafer test)
- DC static test using PMU or DPS
- Test requirements not found in device specifications
- Limit is relaxed vs. IDD static and dynamic tests
- Test requires simple preconditioning pattern
- Does not require precise preconditioning

IDD Static Current

IDD represents the current (I) which flows drain to drain (DD) within a CMOS circuit. When testing TTL devices this specification is called ICC. ICC represents the current (I) which flows collector to collector (CC) within a TTL circuit. Static indicates that the DUT is not active during the test. The following table shows a sample Static IDD specification:

Parameter	Description	Test Conditions	Min	Max	Units
IDD Static	Power Supply Current	VDD = 5.25V, inputs=VDD Iout = 0		+22	µA

Why Test for Static IDD Current?

The static IDD test insures that the DUT will not consume more current than the value stated in the device specification, when the DUT is preconditioned to its lowest current consumption logic state. This measurement is extremely important for battery operated devices. It is also a good way to identify processing problems with CMOS devices.

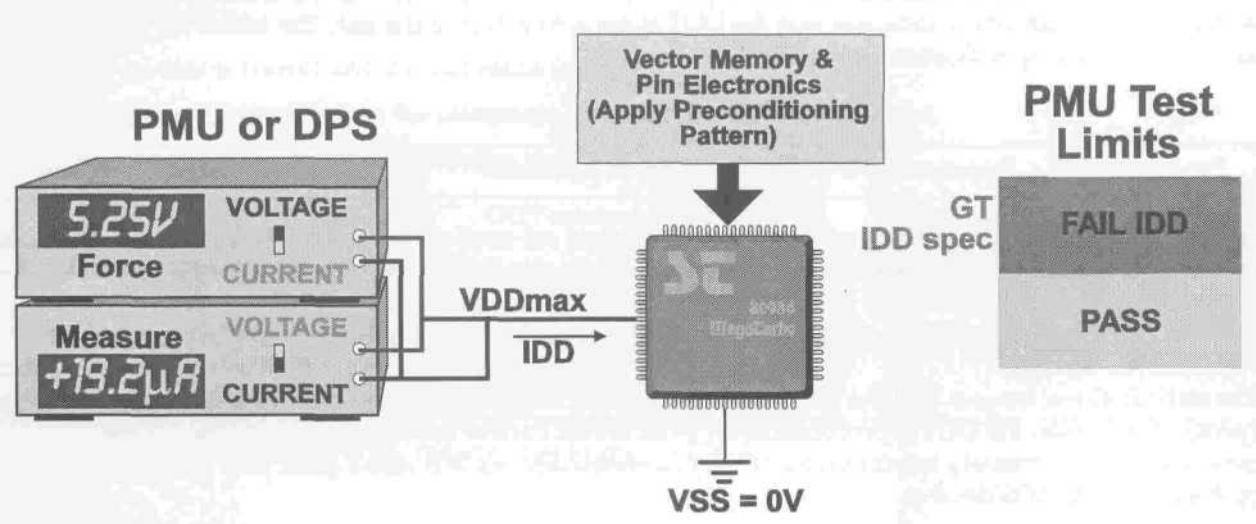
Static IDD Current—Test Method

The static IDD test measures total current flow into the VDD pin. It is performed by executing a test vector pattern that preconditions the device to a known state, typically the state that draws the least amount of IDD current. The device is then held in a static condition and the amount of current flowing into the VDD pin is measured. The measured current is then compared to the IDD static device specification. The following parameters influence the test results and must be set as defined in the device specification: VIL, VIH, VDD, vector sequence and output loading.

The design engineer should specify the exact preconditioning sequence required to achieve the lowest current readings. If this information is not available, some experimenting may be required to find the correct vector sequence. The precise preconditioning sequence is critical to achieving the correct IDD measurement.

If the expected IDD current is very small, additional delay time (settling time) may be required before making the current measurement. External by-pass capacitors on the test hardware can affect the measured results. In some cases it may be necessary to use a relay to disconnect external by-pass capacitors to make an accurate measurement.

Static IDD Test



Using DPS or PMU, apply VDDmax.
Execute Preconditioning Pattern.
Stop pattern.
Wait 1 to 5msec (Set delay).
Measure current flowing into VDD pin(s).
Fails IDD test if measured current is greater than IDD spec.

Figure 6-9

Note: VDDMAX is the worst case for this test. A device with a single power supply set to a positive voltage will always draw a positive current.

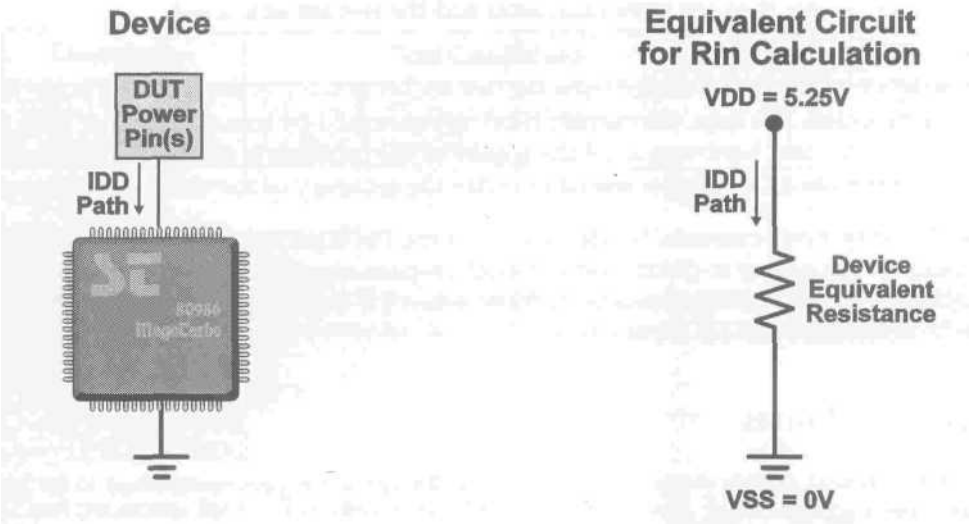
Datalog of: Static IDD Current using the PMU

Pin	Force/rng	Meas/rng	Min	Max	Result
VDD1	5.25V/10V	19.20uA/25uA		+22uA	PASS

Static IDD Resistance Calculation

The static IDD test measures the total resistance of the DUT between VDD and ground. When the VDD voltage is specified at 5.25V and the IDD limit is 22uA, Ohm's law can be used to determine the minimum allowable resistance.

Static IDD Resistance Calculation



Specification
VDD = 5.25V
IDD max = 22uA

R = E / I
E = VDD - VSS (= 5.25 - 0.0)
R = 5.25 / 22uA

R = 238,636Ω

Figure 6-10

Ohm's law is stated as: R=E / I
R is the resistance that is being tested for
E is the voltage (VDD)
I is the current (IDD)

Figure 6-10 shows a 238,636Ω resistor placed in the DUT test socket between VDD and ground to produce the IDD current measurement of 22uA.

Static IDD—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the measured results. The result will be one of three possibilities.

1. The current is correct and the test result is a pass.
2. The current is over the maximum limit and the test result is a fail.
3. The current is less than the minimum limit and the test result is a fail.

When a failure occurs remove the device from the test socket and rerun the test. The Static IDD test should pass with an open socket. If it fails, the current is being consumed by something other than the DUT so eliminate sections of the test hardware until the source of the problem is found. Put a resistor between the VDD and Ground pins as a DUT replacement to verify the accuracy of the measurement system.

When measuring very small currents, the delay time of the PMU may need to be increased. In some cases it may be necessary to use a relay to disconnect external by-pass capacitors to achieve an accurate measurement. The static IDD test should produce consistent results when the test is repeated on a single DUT. Rerun the test several times to be sure the results are stable.

Static IDD—Key Points

- Purpose: to detect power supply currents when the device is preconditioned to its lowest current consumption mode
- DC static test using PMU (or DPS) to supply VDD and measure current
- Effective method of identifying processing problems within CMOS devices
- Test limits defined in device specifications, but exact preconditioning may not be defined
- Test requires preconditioning pattern
- CMOS IDD currents affected by: input levels, input pull-up and pull-down resistors, VDD levels, vector sequence, output current loading and output capacitance loading

IDDQ

The IDDQ test measures the quiescent current under varying logic conditions and provides improved test coverage as compared to the standard Static IDD test.

To perform this test a sequence of Static IDD tests, consisting of 6 to 12 individual current measurements, is performed at unique points within the functional vector set. The intent of the test vector sequencing is to toggle on and off as many internal gates as possible while verifying the amount of IDD leakage current at each vector stopping point.

The IDDQ test may detect minor defects within the core of the circuit that could not otherwise be detected.

IDD Dynamic Current

IDD is the current (I) which flows drain to drain (DD) in a CMOS circuit. When testing TTL devices this specification is called ICC. ICC is the current which flows collector to collector (CC) within a TTL circuit. Dynamic indicates that the DUT is active (being clocked or with its internal devices changing logic states) during the test. The following table shows a sample IDD specification:

Parameter	Description	Test Conditions	Min	Max	Units
IDD Dynamic	Power Supply Current	VDD = 5.25V (commercial) f = fMAX(66MHz)		18	mA

Why Test for Dynamic IDD current?

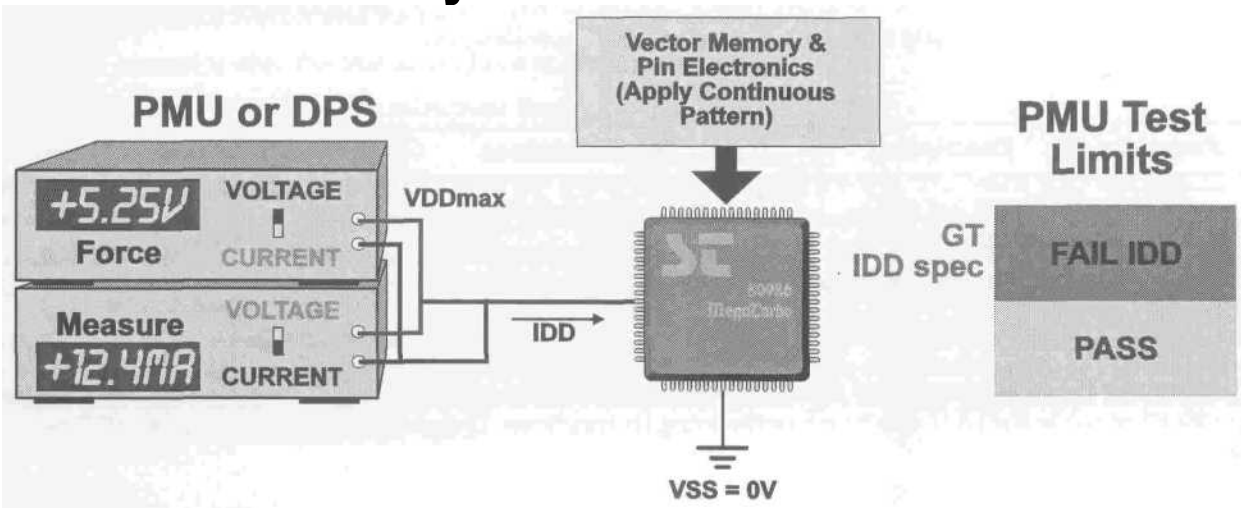
The dynamic IDD test insures that the DUT will not consume more current than the value stated in the device specification while the DUT is actively performing its functions. Dynamic IDD is the operating IDD of the device.

Dynamic IDD—Test Method

The dynamic IDD test measures the total current flow into the VDD pin(s). It is made by the PMU or DPS while executing a continuous test vector pattern, normally at the maximum operating frequency of the DUT. The resultant current measurement is then compared to the IDD dynamic current given in the device specification. The following parameters will influence the test results and therefore must be set as defined in the device specification: VIL, VIH, VDD, test frequency, vector sequence and output loading.

Some test systems provide the ability to measure current flow within the DPS, but the accuracy of the test hardware may not produce reliable readings for low currents. If the DPS current measurement accuracy is not adequate then the PMU must be used. This requires a longer test time but the PMU gives more accurate results.

Dynamic IDD Test



Using DPS or PMU, apply VDDmax.
Execute continuous pattern.
Wait 5 to 10msec (Set delay).
Measure current into VDD pin(s) while device is actively executing pattern.
Fails IDD if measured current is greater than IDD spec.
Stop the pattern.

Figure 6-11

The design engineer should specify the exact preconditioning sequence used to exercise the device during the dynamic IDD test. If this information is not available some experimenting may be required to find the correct vector sequence. External by-pass capacitors on the test hardware may also affect the measured results. See Figure 6-11.

Notes: 1) VDDMAX is the worst case for this test. 2) A device with a single power supply set to a positive voltage will always draw a positive current.

Datalog of: Dynamic IDD Current using the DPS

Pin	Force/rng	Meas/rng	Min	Max	Result
VDD1	5.25V/ 10V	12.4mA/ 25mA		+18mA	PASS

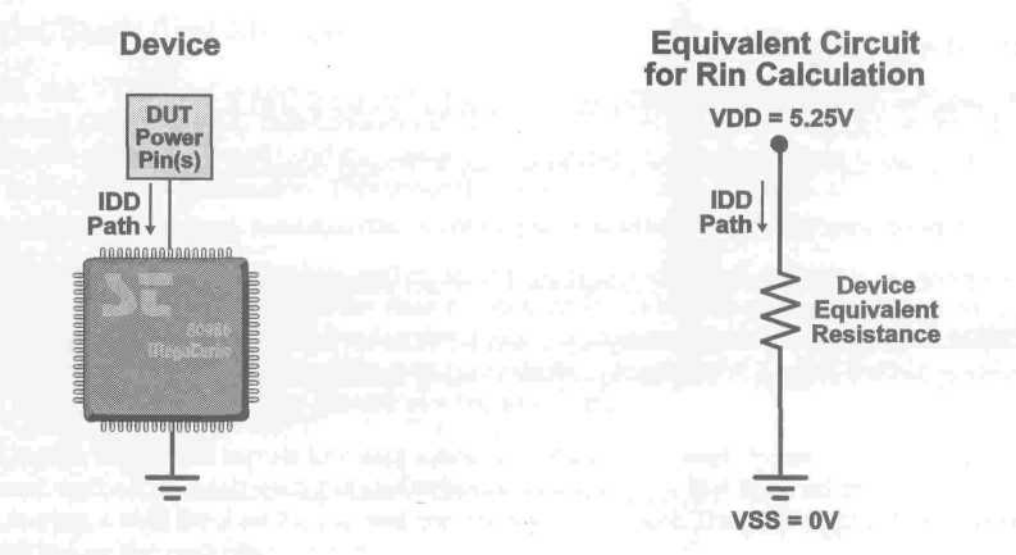
Dynamic IDD Resistance Calculation

The dynamic IDD test measures the total resistance of the DUT between VDD and ground. When the VDD voltage is specified at 5.25V and the IDD limit is 18mA Ohm's law can be used to determine the minimum allowable resistance.

Ohm's law is stated as: R = E / I
R is the resistance that is being tested for
E is the voltage (VDD)
I is the current (IDD)

When a 292Ω resistor is placed in the DUT test socket between VDD and ground the IDD current measurement should read 18mA.

Dynamic IDD Resistance Calculation



Specification
VDD = 5.25
V IDD max = 18mA

R = E / I
E = VDD - VSS (= 5.25 - 0.0)
R = 5.25/18mA
R = 292 Ω

Figure 6-12

Dynamic IDD—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the results. The result will be one of three possibilities:

1. The current is correct and the test result is a pass.
2. The current is over the maximum limit and the test result is a fail.
3. The current is less than the minimum limit and the test result is a fail.

When a failure occurs, remove the device from the test socket and rerun the test. As with the Gross and Static IDD tests, an open socket test should pass with no current. If it fails, the current is being consumed by something other than the DUT, so eliminate sections of the test hardware until the source of the problem is found. A resistor can also be used to simulate the DUT by placing it between the VDD and Ground pins to check the accuracy of the measurement system.

When measuring dynamic IDD currents, the delay time of the PMU may need to be adjusted—some experimenting may be required. Sometimes it is necessary to use a relay to disconnect external by-pass capacitors to get an accurate measurement. The dynamic IDD test should produce consistent results when the test is repeated. Rerun the test several times with the same DUT and make sure the test produces stable results.

Dynamic IDD—Key Points

- Purpose: to detect power supply currents when the device is active
- DC dynamic test using PMU (or DPS) to supply VDD and measure current
- Test requires preconditioning pattern and pattern which executes during measurement
- Test limits defined in device specifications, but exact preconditioning may not be defined
- CMOS IDD currents are affected by input levels, VDD levels, output current (resistive and capacitive) loading, vector sequence, output capacitive loading and test frequency.

Input Currents (IIL/IIH)

IIL is the current (I) in an input (I) when it is forced low(L). IIH is the current (I) in an input (I) when it is forced high (H). A specification limit for IIL/IIH. is shown below for the 256 x 4 Static RAM device:

Parameter	Description	Test Conditions	Min	Max	Units
IIL, IIH	Input Load Current	Vss ≦ Vin ≦ VDD = 5.25	-10	+10	µA

Why Test for IIL/IIH?

The IIL test measures the resistance from an input pin to VDD. The IIH test measures resistance from an input pin to VSS. This test insures that the resistance of the input meets the design parameters and guarantees that the input will not draw more than the specified IIL/IIH current. It is also a good way to identify processing problems in CMOS devices. There are several methods used to perform the IIL/IIH input current test as described below.

IIL/IIH—Serial/Static Test Method

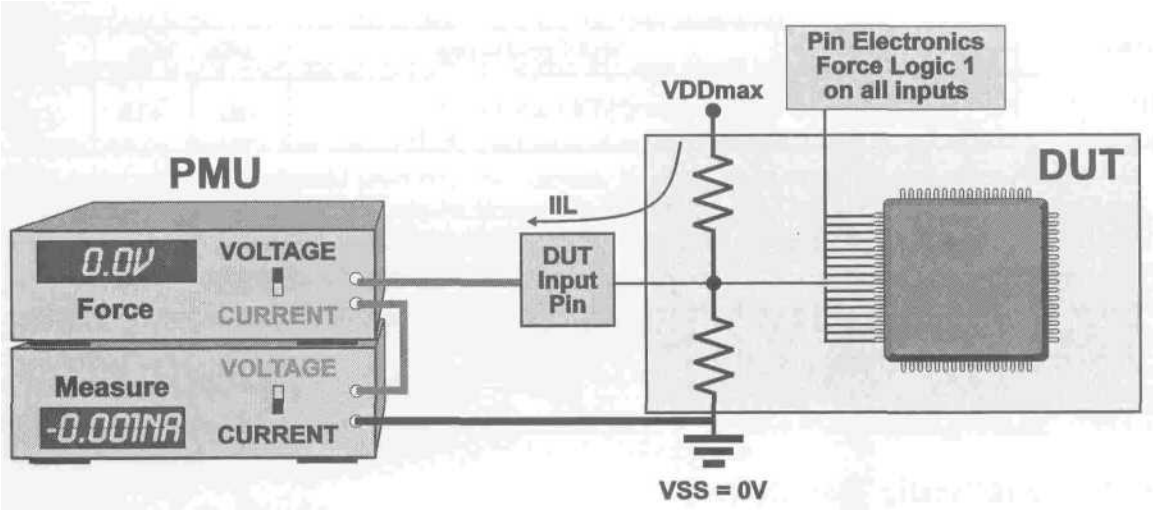
To perform the IIL test, VDDMAX is applied and all input pins are preconditioned to logic 1 using the functional pin drivers (VIH). The DC measurement system (PMU) then forces each input low (pin driver is disconnected from the pin being tested) and the resulting current is measured and compared to the IIL current limit as defined in the device specification. This process is repeated on each pin until all inputs have been tested.

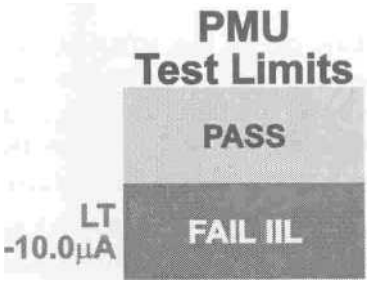
To perform the EH test, VDDMAX is applied and all inputs are preconditioned to logic 0 using the functional pin drivers (VIL). The DC measurement system then forces each input high and the resulting current is measured and compared to the EH current limit as defined in the device specification. This process is repeated on each pin until all inputs have been tested. Be sure to verify the values for VDD, Vin (voltage applied) and IIL/IIH (current limits). This test requires the use of a current clamp.

Note: Setting all inputs high or all inputs low may cause a problem with some types of DUTs. An alternative serial measurement method is to test each pin individually, first forcing a low level on the pin and measuring the current then forcing a high level on the pin and measuring the current. The tested pin is then returned to its original logic state before the next pin is tested.

The advantage of the Serial test is the ability to measure the individual current flow of each pin, and also to test for pin to pin leakage. Since the pin under test receives a different voltage from all other inputs, any leakage path between inputs will be found. The disadvantage is the test time required to measure each pin individually. See Figure 6-13 and Figure 6-14.

Input Leakage Low Test (ML)
Serial Method





Apply VDDmax.
Precondition all inputs to logic 1 with pin drivers.
Using PMU, force individual inputs to VSS.
Wait 1 to 5 msec (Set PMU delay).
Measure resultant current.
Fails IIL if measured current is less than -10.0 uA.

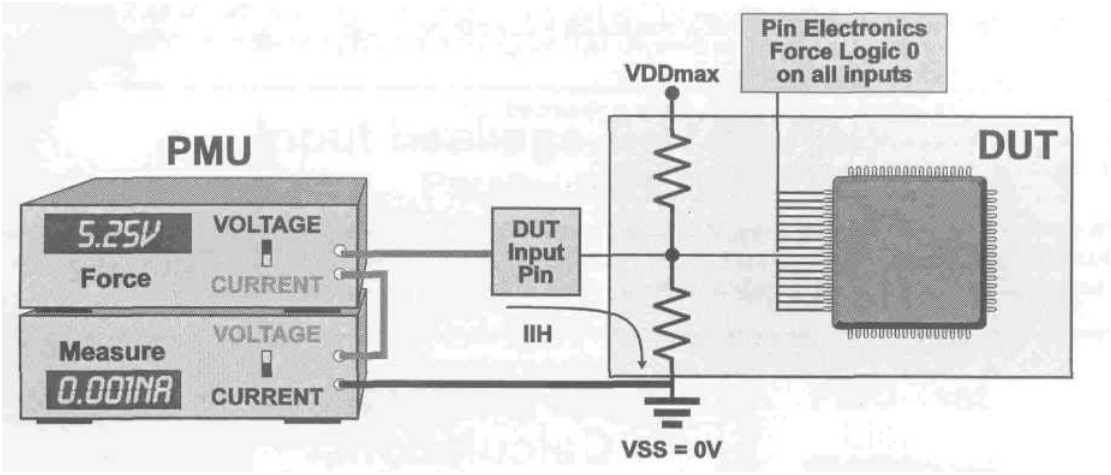
Figure 6-13

Special Note

Setting all inputs low or all inputs high at the same time may result in an invalid condition to the device under test—check the functional truth table before implementing. It is also important to note that bi-directional pins may be inadvertently turned on during the IIL/IIH test. If these pins are being driven by the tester, high IDD currents may cause the DUT's internal supply voltage to droop below the input testing voltage, causing the input protection device to draw current.
If the device is CMOS and the bi-directional pins are left floating, high currents may also occur. The solution to this problem is to place an output load on the bi-directional pins to pull them to a solid 1 or 0 level. If the pin is inadvertently turned on, the current is limited by the load.

Input Leakage High Test (IIH)

Serial Method



Apply VDDmax.
Precondition all inputs to logic 0 with pin drivers.
Using PMU, force individual inputs to VDDmax.
Wait 1 to 5 msec (Set PMU delay).
Measure resultant current.
Fails IIH if measured current is greater than +10.0uA.

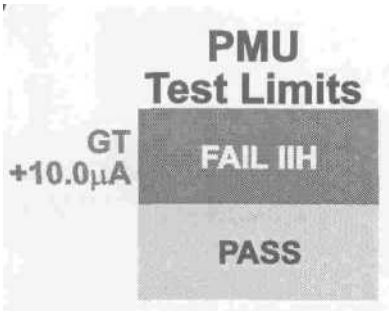


Figure 6-14

Note: The IIL/IIH input current test is normally performed on pure input pins only. Read the device specification carefully to determine which pins to test. Typically when testing CMOS devices, VDD is set to VDDMAX (worst case), and the inputs are forced to zero and VDDMAX volts.

Datalog of: IIL/IIH serial/static test using the PMU

Pin	Force/rng	Meas/rng	Min	Max	Result
PIN1	5.250V/ 8V	I.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN1	0.000V/ 8V	0.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN2	5.250V/ 8V	20.40uA/ 20uA	-I0.00uA	I0.00uA	FAIL
PIN2	0.000V/ 8V	0.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN3	5.250V/ 8V	I.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN3	0.000V/ 8V	-I.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN4	5.250V/ 8V	I.00nA/ 20uA	-I0.00uA	I0.00uA	PASS
PIN4	0.000V/ 8V	-18.60uA/ 20uA	-I0.00uA	I0.00uA	FAIL

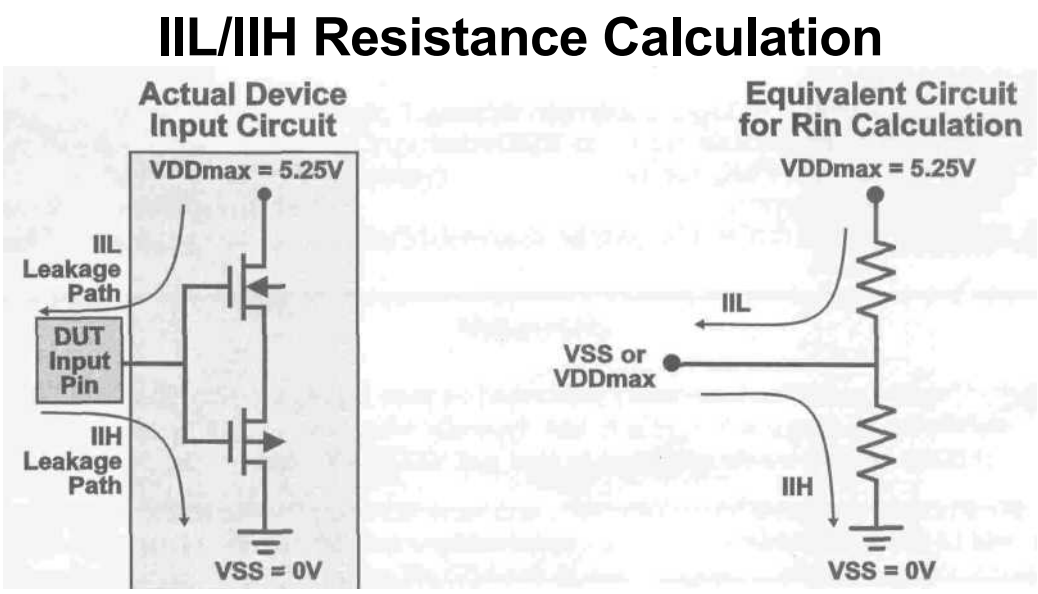
IIL/IIH Resistance Calculation

The input leakage test measures the resistance from the input to VSS when VDD is applied, and from the input to VDD when VSS is applied. By applying a voltage to the input and measuring the resulting current, the input resistance can be determined using Ohm's law. The device specification defines the maximum allowable current that can flow from the input (IIL/IIH) when the test voltage (Vin) is applied to the input pin.

Using Ohm's law the minimum allowable resistance value can be found:

Ohm's law is stated as: R = E / I
R is the resistance that is being tested for
E is the voltage across the resistance (VDD/GND)
I is the maximum current (IIL/IIH)

In the example shown in Figure 6-15, the minimum resistance that the input can offer and still pass the test is 525KΩ If the resistance of the input is less than 525KΩ the current flow would exceed the IIL/IIH limit and the test would fail. The typical resistance of a CMOS input is much greater than 525KΩ



Specification Vin = VSS (for IIL test)
Vin = VDDmax (for IIH test)
IIL= 10uA

R = E / I
E = VDDmax - VSS (for IIL and IIH tests)
R = 5.25/10uA
R = 525,000Ω (minimum)

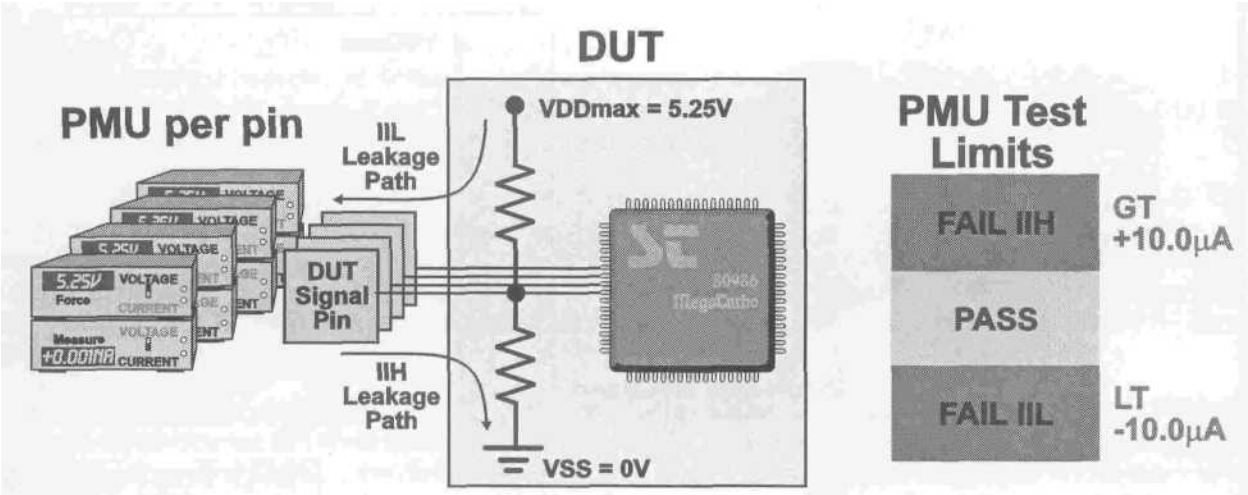
Figure 6-15

IIL/IIH—Parallel Test Method

Some test systems have the ability to perform parallel leakage measurements. Parallel Measurement means that all measurements are made simultaneously, but on an individual basis. The parallel leakage test is performed by using PMU per pin circuitry located on the pin electronics cards. All inputs are forced to a logic 1 and the current flow of each pin is measured in parallel (simultaneously). All inputs are then forced to a logic 0 and the current flow of each pin is again measured in parallel. The results of the current measurements are compared to the limits set in the test program and a pass/fail decision is made.

Input Leakage Test (IIL/IIH)

Parallel Method



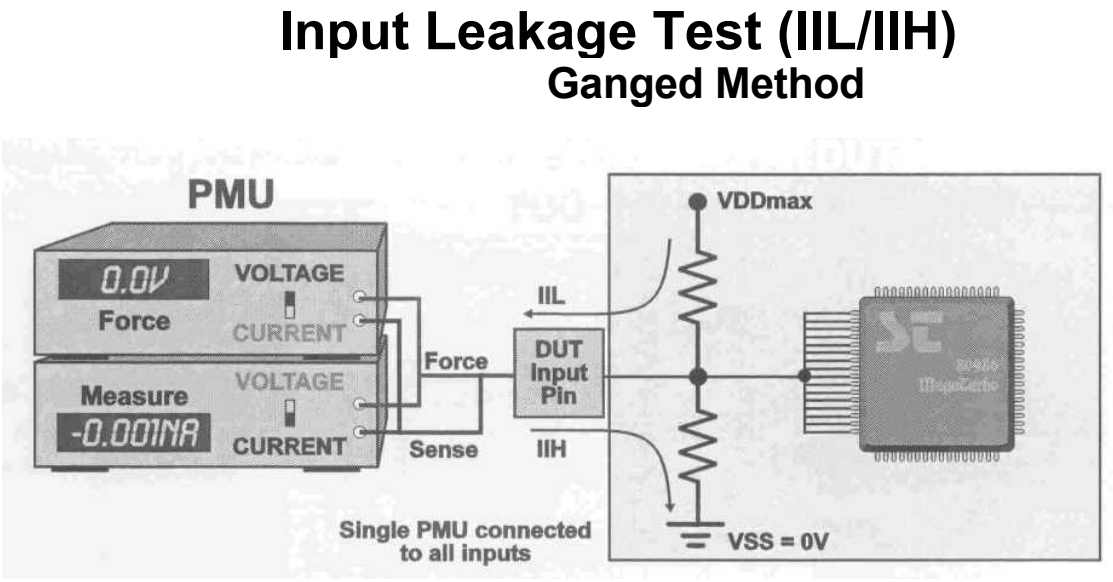
Apply VDDmax.
Using PMU per pin, force each pin to VDDmax (IIH test).
Wait 1 to 5 msec (Set PMU delay).
Measure resultant current.
Fails if measured current (per pin) is outside limits.
Repeat test forcing each pin to VSS (IIL test).

Figure 6-16

The advantage of this technique is that the IIL/IIH test can be performed quickly and yet the individual current flow of each pin is measured. The disadvantage is that pin to pin leakage between inputs is more difficult to detect because all inputs are held at the same voltage level when the current measurements are made. This method also requires the test system to have PMU per pin hardware.

IIL/IIH—Ganged Test Method

Some test systems have the ability to perform ganged leakage measurements. Ganged Measurement means the measurement is made by connecting a single PMU to all inputs at one time and measuring the total current flow. All inputs are forced to logic 1 and the total current flow is measured. All inputs are then forced to logic 0 and the total current flow is again measured. The results of the current measurements are compared to the limits set in the test program and a pass/fail decision is made.



Apply VDDmax.

Using PMU, force all input pins to VDDmax. (IIH)
Wait 1 to 5 msec (Set PMU delay).
Measure resultant current.
Fails IIH if measured current is outside limits.
Repeat test forcing all input pins to VSS. (IIL)
If test fails, retest using serial method.

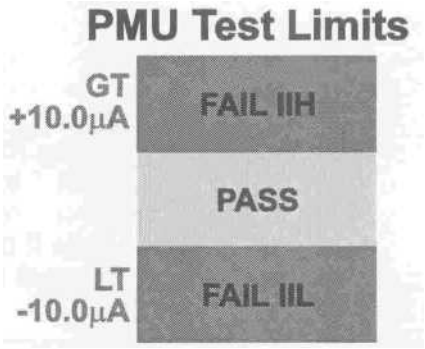


Figure 6-17

The current limit for the ganged test is set to the value found in the device specification for an individual pin. In the event that the measured current exceeds the limit, the program must then perform the test again using the serial measurement technique as defined previously. This ganged test works well when testing CMOS inputs with very high impedance. These inputs typically measure zero current, so the sum of the total current flow is zero. Ganged leakage tests cannot be performed on resistive inputs or inputs with pull-up or pull-down inputs because the sum of leakage current for all pins will be greater than the specification limit for an individual pin.

The advantage of this technique is that the IIL/IIH test can be performed quickly and does not require the PMU per pin circuitry. The disadvantages are that this method can only be used when testing inputs with very high impedance, the individual current measurement for each pin is unknown, and the test must be executed again using the serial test method each time a failure occurs.

IIL/IIH—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the result. The result will be one of three possibilities.

1. The measured current is within the upper and lower limits and the test result is a pass.
2. The measured current is outside of one of the limits and the test result is a fail, but the current is not excessive (it is within the measurement range).
3. The measured current is outside of one of the limits and the test result is a fail, but the current is excessive (it is not within the measurement range).

To debug this test, first remove the device from the test socket and rerun the test. The result of an open socket test should be no current (a pass). If not, the current is being consumed by something other than the DUT so eliminate sections of the test hardware until the source of the problem is found.

The datalog above shows that the measurement on pin 4 is just outside the limit (as described in item 2). This failure is most likely due to a defective device and may be caused by variations in the fabrication process or perhaps a pin damaged by static electricity. It can be seen from the datalog that the path to VDD is causing the leakage—when ground is applied to the input pin, current flows from VDD through the device and into the PMU, resulting in a negative current. If necessary, verify the accuracy of the measurement system by replacing the DUT with a resistor.

The datalog above shows that the measured current on pin 2 is quite excessive as described in item 3. Note that this current represents the full scale reading of the PMU in its 20uA range. This type of failure is most likely caused by a serious defect within the device. It can be seen from the datalog that the path to VSS is causing the leakage. When VDD is applied to the input pin, a (positive) current flows from the PMU through the device and into VSS.

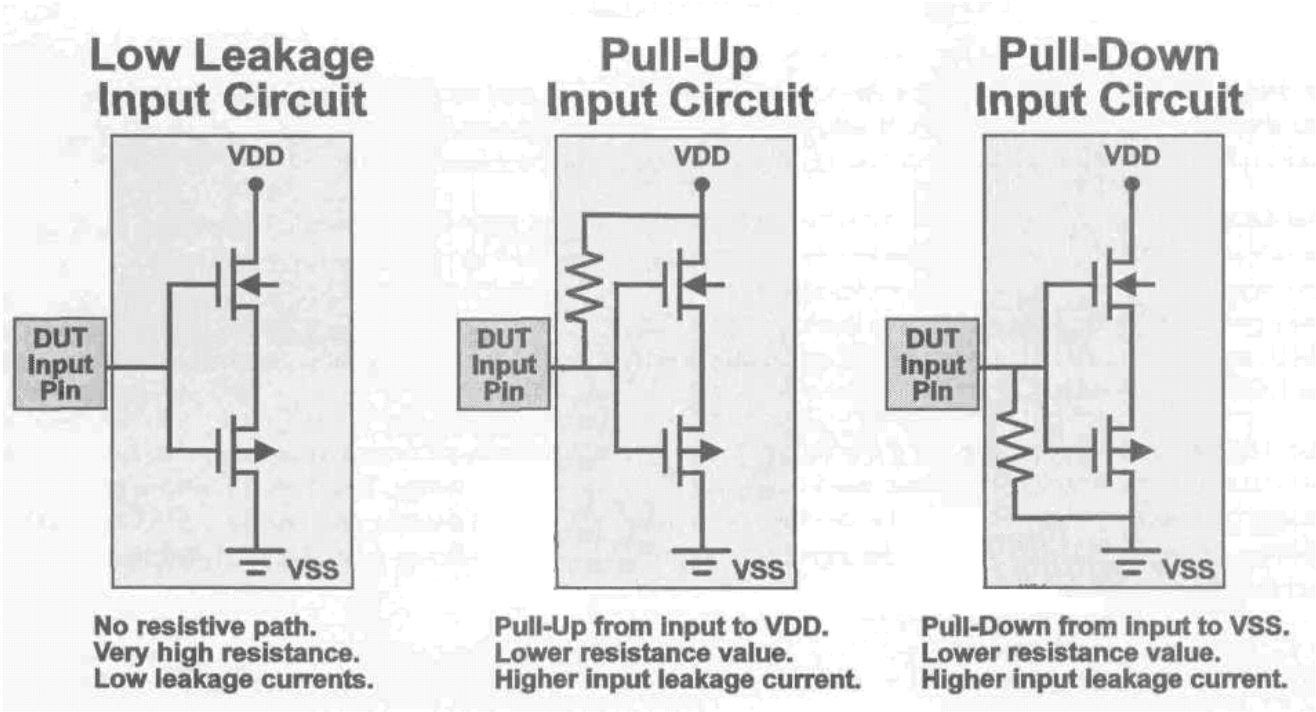
IIL/IIH—Key Points

- Purpose: to verify that the input buffers offer a high resistance when applying 0V and VDD
- DC test uses PMU to force voltage and measure current
- Serial test method is very slow—use alternative method if possible
- Test limits defined in device specifications
- VDDMAX is worst case test condition

Resistive Inputs—Pull-ups and Pull-downs

Certain types of inputs may have active pull-ups, pull-downs or resistive paths associated with them. The device specification may define a current range, for example 80uA to 120uA. In this example each input may be expected to draw approximately 100uA when tested under the conditions defined in the device specification. Since each pin draws a specific amount of current, each pin must be tested individually; ganged testing will not work. Parallel per pin testing will work fine and will result in a high speed test. Resistive inputs can also affect the IDD current of a device, depending on the voltage level set for each input.

CMOS Input Structures



Resistive Inputs—Key Points

- Purpose: to verify that the input buffers were manufactured with the correct resistance paths to either VDD or ground
- Ganged testing will not work
- Dual test limits are often used to test for a range of acceptable resistance

Output Fanout

Fanout is defined as the ability of a single device output pin to drive (or control) multiple device input pins, based on the voltage and current parameters.

We have looked at input leakage and output drive specifications independently. Now we can see how these specifications are used by application design engineers. Figure 6-19 shows the relationship between the DC specifications for outputs and inputs. In most semiconductor applications, the various devices communicate with each other through direct interconnections. This means that the output of one device will be connected to one or more inputs of other devices.

The application design engineer who uses the various semiconductors within a system will need to know the voltage and current requirement of each input as well as the voltage and current drive capability of each output. This information is defined in the device specification and the test program must therefore provide the proper test conditions to guarantee that the device meets the published specifications. An example specification is shown.

Parameter	Description	Test Conditions	Min	Max	Units
VOH	Output HIGH Voltage	VCC = 4.75V, IOH = -2.6mA	2.4		V
VOL	Output LOW Voltage	VCC = 4.75V, IOL = 24.0mA		0.4	V
IIL	Input Low Load Current	Vin = 0.4V	-800		uA
IIH	Input High Load Current	Vin = 2.4V		150	uA

Note: Fanout is different for TTL and CMOS. Because most CMOS devices have high impedance input structures, DC fanout for CMOS is virtually unlimited. In other words, if you can wait long enough, one CMOS output can drive as many CMOS inputs as you like. A CMOS input looks like a capacitor, and the more device inputs you connect together, the higher the capacitance. The output which drives this capacitor must charge it up to VIH and down to VIL when making logic 0 to 1 and 1 to 0 transitions. In a similar way, the device outputs must drive the test system inputs which also have capacitance.

Device Fanout Capability

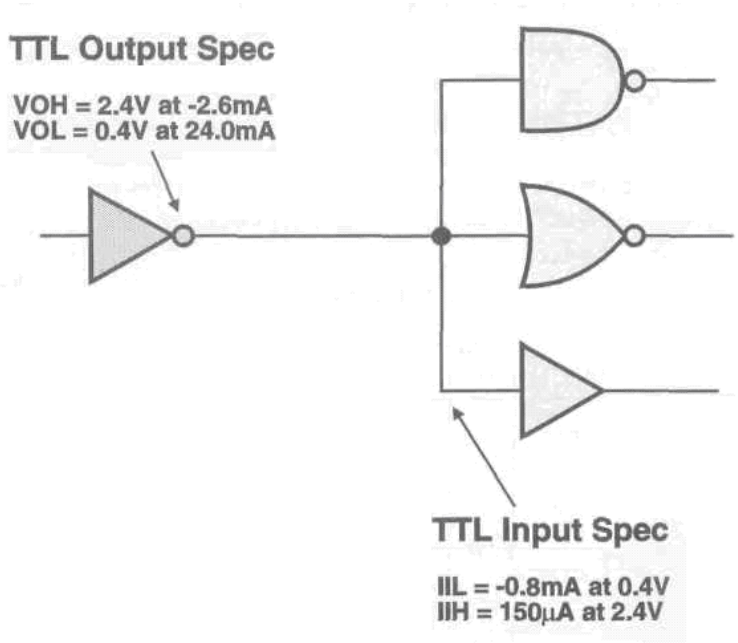


Figure 6-18

Look at the example shown in Figure 6-19. Given the specifications defined, how many inputs can one output drive low, how many inputs can one output drive high? What is the maximum number of inputs that one output can be connected to and still guarantee correct operation?

Answer: One output can drive 17 inputs high and 30 inputs low, therefore the output high drive capability will limit the number of input connections to a total of 17.

High Impedance Currents (IOZL/IOZH)

IOZL is the current (I) from the output (O) when the output is in the high impedance state (Z) and a low voltage (L) is applied to the output. IOZH is the current (I) from the output (O) when the output is in the high impedance state (Z) and a high voltage (H) is applied to the output.

Why Test for IOZ?

The IOZ high impedance leakage tests are made to insure that bi-directional and high impedance outputs are capable of achieving a high impedance or off state. The IOZL test measures the resistance from an output pin to VDD when the output is in the high impedance state. The IOZH test measures the resistance from an output pin to ground when the output is in the high impedance state. This test insures that the resistance of the output, when it is turned off, meets the design parameters and guarantees that the output will not draw more than the specified IOZL/IOZH current. It is also a good way to find process problems in CMOS devices. A sample specification for IOZ is shown below:

Parameter	Description	Test Conditions	Min	Max	Units
IOZ	Output Current High-Z	VSS $\leq$ Vout $\leq$ VDD = 5.25V Output Disabled	-2.0	+2.0	uA

IOZL/IOZH—Serial/Static Test Method

To perform the IOZ test, VDD is applied and a pattern is executed which preconditions the device pins to their high impedance state. The DC measurement system (PMU) forces both high and low voltage onto each pin (one at a time) and the resulting current is measured. The measured current is then compared to the IOZ current limit in the device specification. This process is repeated on each pin individually until all high-Z pins have been tested. Be sure to verify the values for VDD, VOZ (voltage applied to the output) and the IOZ current limits. This test requires the use of a current clamp.

The advantage of the Serial test is the ability to measure the individual current flow of each pin. The disadvantage is, as usual, the test time required to measure each pin individually. See Figure 6-20.

IOZL/IOZH—Parallel Test Method

Some test systems have the ability to perform parallel leakage measurements. Parallel Measurement means that all measurements are made simultaneously, but on an individual basis. The parallel leakage test is performed by using PMU per pin circuitry located on the pin electronics cards. A pattern is executed which preconditions the device pins to their high impedance state. Each PMU forces a logic 0 and the current flow of each pin is measured at the same time. A logic 1 is then forced and the current flow of each pin is again measured in parallel. The results of the current measurements are compared to the limits set in the test program and a pass/fail decision is made.

The advantage of this technique is that the IOZ test can be performed quickly and yet the individual current flow of each pin is measured. This method requires the test system to have PMU per pin test hardware.

Parametric Tristate Leakage Test (IOZ)

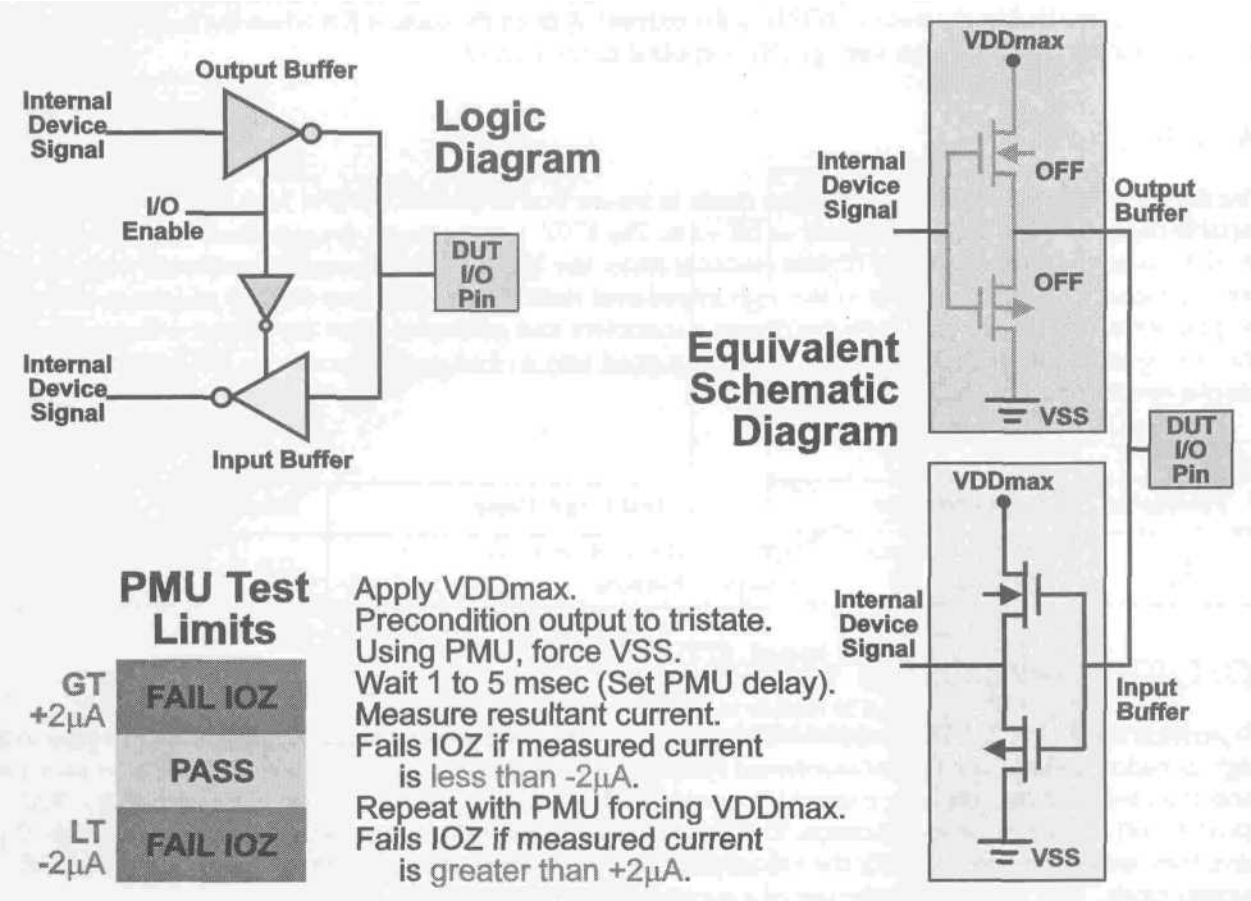


Figure 6-19

IOZL/IOZH—Ganged Measurements

Ganged measurements are not performed on outputs.

Note: The IOZL/IOZH high impedance current test is normally performed on outputs that have the ability to turn off (three-state and bi-directional pins). Read the device specification carefully to determine which pins to test. Typically when testing CMOS devices, VDD is set to VDDMAX (worst case) and the outputs are forced to zero and VDDMAX volts.

IOZ Resistance Calculation

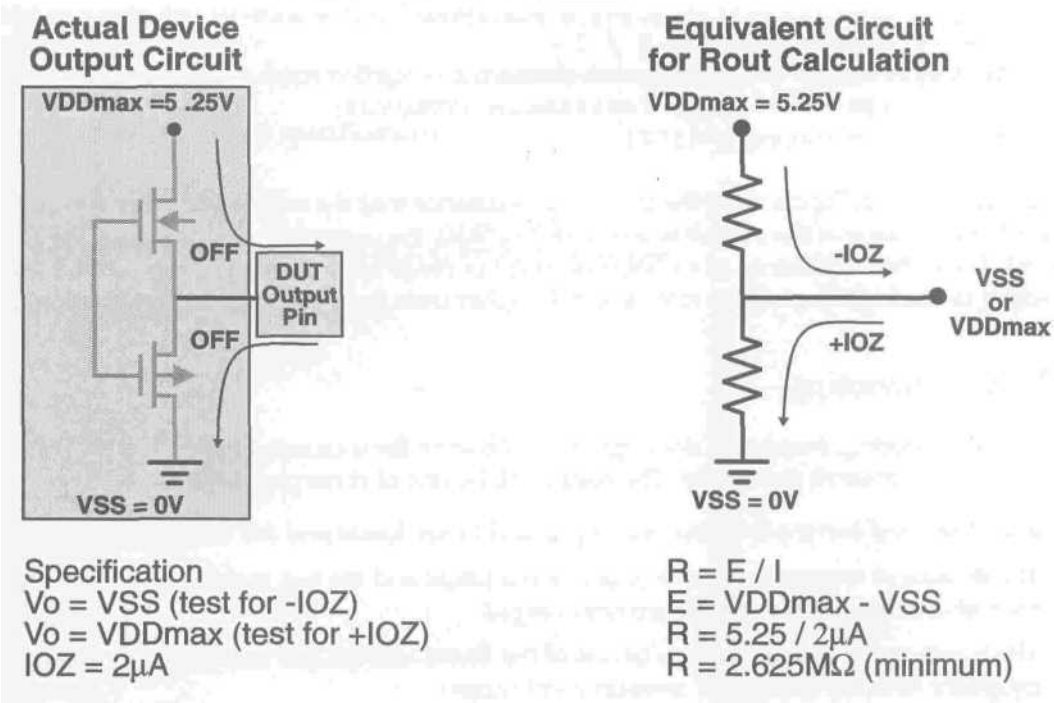


Figure 6-20

Datalog of: IOZL/IOZH serial/static test using the PMU						
Pin	Force /	rng	Meas/ rng	Min	Max	Result
PIN1	5.250V/	8V	I.0nA/ I0uA	-2.00uA	2.00uA	PASS
PIN1	0.000V/	8V	0.0A/ I0uA	-2.00uA	2.00uA	PASS
PIN2	5.250V/	8V	10.2uA/ I0uA	-2.00uA	2.00uA	FAIL
PIN2	0.000V/	8V	0.0nA/ I0uA	-2.00uA	2.00uA	PASS
PIN3	5.250V/	8V	I.0nA/ I0uA	-2.00uA	2.00uA	PASS
PIN3	0.000V/	8V	-I.0nA/ I0uA	-2.00uA	2.00uA	PASS
PIN4	5.250V/	8V	I.0nA/ I0uA	-2.00uA	2.00uA	PASS
PIN4	0.000V/	8V	-78uA/ I0uA	-2.00uA	2.00uA	FAIL

IOZ Resistance Calculation

The IOZ test measures the resistance of the output when it is turned off. By applying a voltage to the output and measuring the resulting current, the output resistance can be determined using Ohm's law. The device specification defines the maximum allowable current that can flow from the output (IOZ) when the test voltage (Vout) is applied to the output.

Using Ohm's law, the minimum allowable resistance value can be found.

Ohm's law is stated as: R = E / I
R is the resistance being tested for
E is the voltage across the resistance (VDD/VSS)
I is the maximum current (IOZ)

In the example shown in Figure 6-21, the minimum resistance that the output can offer and still pass the test is 2.625MΩ If the resistance of the output is less than 2.625MΩ the current flow will exceed the IOZ limit and the test will fail. The typical resistance of a CMOS output can range from 20MΩ to 50MΩ, so the resistance of a CMOS output in the high impedance state is much higher than the value required by the device specification.

IOZ - Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the results. The result will be one of three possibilities:

1. The measured current is within the upper and lower limits and the test result is a pass.
2. The measured current is outside of one of the limits and the test result is a fail, but the current is not excessive (it is within the measurement range).
3. The measured current is outside of one of the limits and the test result is a fail, but the current is excessive (it is not within the measurement range).

To debug this test, first remove the device from the test socket and rerun the test. The result of an open socket test should be a pass with zero current. If it fails, the current is being consumed by something other than the DUT so eliminate sections of the test hardware until the source of the problem is found.

The datalog above shows that the measurement on pin 4 is just outside the limit (as described in item 2). This failure is most likely due to a defective device and may be caused by variations in the fabrication process. It can be seen from the datalog that the path to VDD is causing the leakage. When ground is applied to the input pin, current flows from VDD through the device and into the PMU, resulting in a negative current. If necessary, verify the accuracy of the measurement system by replacing the DUT pin with a resistor.

The datalog above shows that the measured current on pin 2 is quite excessive as described in item 3. This type of failure is caused by a serious defect within the device or it may be caused by incorrect preconditioning or maybe a paper clip on the load board. If the output is not properly preconditioned, it may be driving a logic 0 or a logic 1 instead of entering a high impedance state as required for this test.

All test vector sequences should be tested in the gross functional test, including the preconditioning sequence used for the IOZ test. The gross functional test should occur before the IOZ test within the program flow. Therefore, the preconditioning vectors should perform correctly when used to setup a device output for a DC test. If the device is not properly preconditioned, you must resolve this issue. Once the output is set to the proper logic state, the VOH/IOH test will most likely pass.

The datalog above shows excessive current flow when 5.25V is applied to pin 2—this indicates that the device output may actually be driving a logic 0. When VDD is applied to the pin, current flows from the PMU

through the device and into ground, resulting in a positive current. To locate the source of this problem, observe the output after preconditioning but before the PMU is connected. It will be possible to determine if the output is actually driving a valid logic level. If it is not driving a valid logic 0 or a logic 1 then the output is most likely defective, thus the failure is not caused by preconditioning.

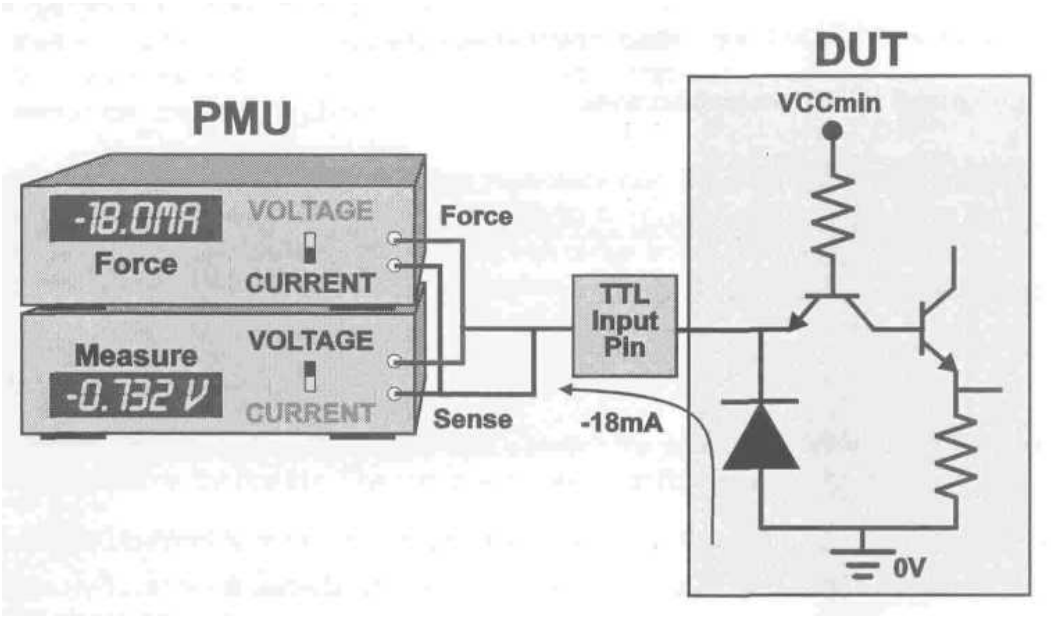
IOZ - Key Points

- Purpose: to verify that tri-state output buffers offer an extremely high resistance when in the off state
- DC static test uses PMU to force voltage and measure current
- Test limits defined in device specifications
- Test requires preconditioning
- Performed only on three-state outputs and bi-directional pins
- VDDMAX is worst case test condition

Input Clamp (VI)

VI represents the voltage (V) measured on the input (I) when a negative current is forced from the input of a TTL (not CMOS) device.

Input Clamp Diode Test



Test applies only to TTL type devices.
Apply VCCmin.
Set Voltage Clamp
Using PMU, force -18mA (input pins only).
Measure resultant voltage.
Fails test if measured voltage is less than -1.5V.

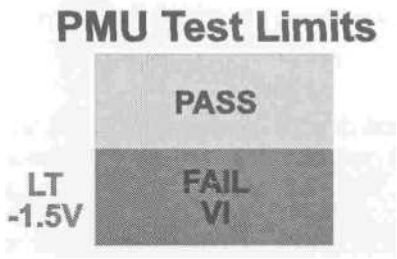


Figure 6-21

A sample specification for the VI parameter is shown below:

Parameter	Description	Test Conditions	Min	Max	Units
VI	Input Clamp Voltage	VCC = Min Iin = -18mA		+1.5	V

Why Test for VI?

The input clamp test verifies the integrity of the input clamp diode connected between the emitter of the input transistor and ground. This test is only performed on TTL circuits.

VI—Serial/Static Test Method

A negative current, in the range of -15mA to -20mA, is forced by the PMU. The resultant voltage is measured and compared to the device specification (often set at -1.5V). This test is performed serially, one pin at a time, using a single PMU.

VI Resistance Calculation

The VI test measures the resistance of the input structure to ground. By applying a negative current to the input, the diode to ground is forward biased and the voltage drop across the diode can be measured.

Using Ohm's law, the maximum allowable resistance value can be found:

Ohm's Law is stated as: R = E / I
R is the resistance being tested for
E is the voltage across the resistance (Vclamp)
I is the forced current (I)

In the example shown in Figure 6-22, the maximum resistance that the input diode can offer and still pass the test is 83.3ΩIf the resistance of the output is greater than 83.3Ω the measured voltage will exceed the VI limit and the test will fail.

VI—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the result. It will be one of three possibilities.

1. The measured voltage is within the lower limit and the test result is a pass.
2. The measured voltage is outside of the limit and the test result is a fail, but the voltage is not excessive (it is within the measurement range).
3. The measured voltage is outside of the limit and the test result is a fail, but the voltage is excessive (it is not within the measurement range or it has been clamped by the PMU).

Datalog of: VI serial/static test using the PMU

Pin	Force/rng		Meas/rng		Min	Max	Result
PIN1	-18.0mA/	20mA	-0.843/	2V	-1.50V		PASS
PIN2	-18.0mA/	20mA	-1.641/	2V	-1.50V		FAIL
PIN3	-18.0mA/	20mA	-2.048/	2V	-1.50V		FAIL
PIN4	-18.0mA/	20mA	-0.876/	2V	-1.50V		PASS

The datalog above shows that the measurement on pin 2 is just outside the limit (as described in item 2). This failure is most likely due to a defective device and may be caused by variations in the fabrication process. If necessary verify the accuracy of the measurement system by replacing the DUT with a resistor from the input pin to ground.

The datalog above shows that the measured current on pin 3 is quite excessive as described in item 3. This type of failure is caused by a serious defect within the device. Notice that the measured voltage is -2.048V, which is a full scale reading on the 2V measurement range. The PMU is thus clamped at its maximum voltage for the range and does not give the actual voltage produced by the DUT.

VI - Key Points

- Purpose: to verify the integrity of the input structure
- DC static test using PMU to force current and measure voltage
- Test limits defined in device specifications
- Performed on TTL devices only
- Performed on one pin at a time

Output Short Circuit Current (IOS)

IOS represents the current (I) produced by the output (O) when a shorted condition (S) is applied to the output. A sample specification for IOS is shown below:

Parameter	Description	Test Conditions	Min	Max	Units
IOS	Output Short Circuit Current	Vout = 0V VDD = 5.25V *Short only 1 output at a time for no longer than 1 second	-85	-30	mA

Why Test for IOS?

The IOS test measures the resistance of an output pin when the output is in the logic 1 state and zero volts is applied to the output. This test insures that the resistance of the output meets the design parameters when subjected to worst case loading conditions and guarantees that an output, when shorted, is capable of sourcing a predefined amount of current. This current also represents the maximum amount of current which can be supplied by the DUT pin to charge a capacitive load and can be used to calculate an output's rise time.

Output Short Circuit—Serial/Static Test Method

The output short circuit test is done by first preconditioning the device so the output under test is driving a logic 1. Zero volts is then applied to an individual output by the DC measurement system (PMU) and the resultant current is measured and compared to the device specification (often a double ended limit). This process is repeated until all outputs have been tested. The device specification may indicate the maximum time an output may be shorted to prevent overheating of the device. Check any "Notes" or "Maximum Ratings" sections of the device specification for details. See Figure 6-23.

Avoid Hot Switching

The IOS test requires careful program planning to avoid hot switching. The device output is first preconditioned to a logic 1 state. The voltage produced by the DUT output will be between VOH and VDD. The device specification states that the output must be shorted to zero volts. If the PMU is programmed to force zero volts then connected to the device output, high current will flow as soon as the PMU connects. The hot switching problem comes about because the PMU is forcing zero volts and the DUT output is supplying VOH.

To correctly perform the test, the PMU must first be programmed to force zero current, putting the PMU into the voltage measure mode. It is then connected to the DUT output and the DUT's VOH voltage is measured and saved. The PMU is then disconnected from the DUT and programmed to force the VOH voltage that was just measured. Now the PMU and the DUT output are both at the same voltage and can be safely connected together (hot switching is avoided). Next the PMU is programmed to force zero volts, the IOS current is measured and compared to the test limits. The PMU is again programmed to force the measured VOH voltage and disconnected. This must be done for each DUT pin that requires the IOS test.

The general rule for avoiding hot switching is to have both sides of a relay at the same voltage before opening or closing the relay.

Output Short Circuit Current (IOS)

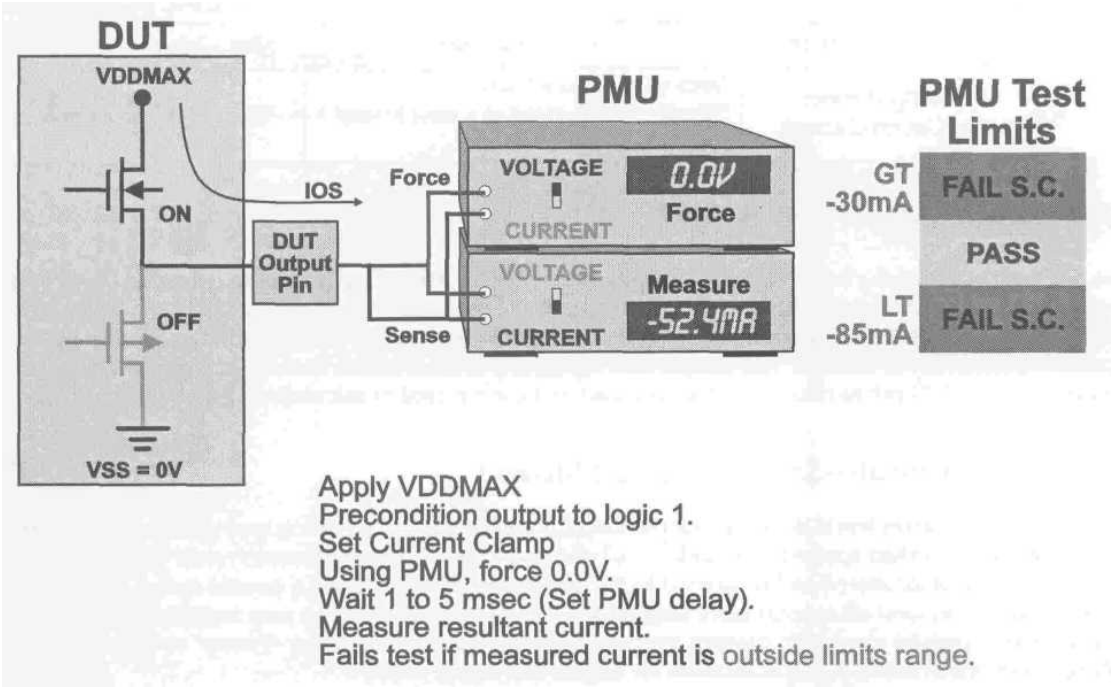


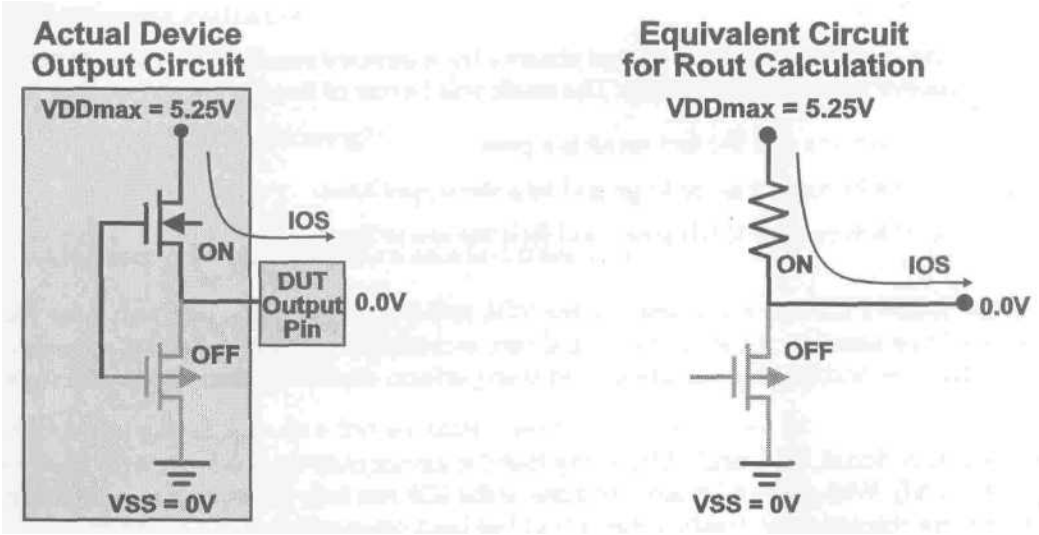
Figure 6-22

Note: The IOS test is normally performed with VDD set to VDDMAX volts.

Datalog of: IOS serial/static test using the PM

Pin	Force/rng		Meas/rng		Min	Max	Result
PIN1	0.000V/	8V	-52.4mA/	100mA	-85.0mA	-30.0mA	PASS
PIN2	0.000V/	8V	-28.5mA/	100mA	-85.0mA	-30.0mA	FAIL
PIN3	0.000V/	8V	-61.6mA/	100mA	-85.0mA	-30.0mA	PASS
PIN4	0.000V/	8V	-92.3mA/	100mA	-85.0mA	-30.0mA	FAIL
PIN5	0.000V/	8V	-0.0 0mA/	100mA	-85.0mA	-30.0mA	FAIL

IOS Resistance Calculation



Specification	Case 1	Case 2:
Vo =0.0 V	Maximum Resistance	Minimum Resistance
IOS = at least 30mA	R=E / I	R=E / I
no more than 85mA	R=5.25/30mA	R=5.25/85mA
	R=175W maximum	R=61.7W minimum

Figure 6-23

IOS Resistance Calculation

The IOS test measures the resistance of an output when a shorted condition is applied to that output. By applying zero volts to the output and measuring the resulting current, the output resistance can be determined using Ohm's law. The device specification defines a range of allowable current that can flow from the output (IOS) when the test voltage (Vout) is applied to the output.

Using Ohm's law, the minimum allowable resistance value can be found.

Ohm's Law is stated as: R = E / I
R is the resistance being tested for
E is the voltage across the resistance (VDD/GND)
I is the maximum current (IOSmin / IOSmax)

In the example shown in Figure 6-24, the minimum resistance that the output can offer and still pass the test is 61.7Q. If the resistance of the output is less than 61.7Ω the current flow exceeds the IOS limit and the test fails. The maximum resistance that the output can offer and still pass the test is 175ΩIf the resistance of the output is greater than 175Ωthe current flow is less than the IOS limit and the test fails.

IOS—Trouble Shooting

To begin trouble shooting, enable the datalogger and observe the measured result. If a DUT standard is available, test it and observe the measured results. The result will be one of three possibilities.

1. The IOS current is correct and the test result is a pass.
2. The magnitude of IOS current is too large and fails the upper limit.
3. The magnitude of IOS current is too small and fails the lower limit.

Generally the IOS test follows the functional test and the VOL and VOH tests in the program flow. All test vector sequences should be tested in the gross functional test, including the preconditioning sequences used within the DC tests. The preconditioning vectors should then perform correctly when used to set up a device output for a DC test.

Knowing the device is functional, VOL and VOH verify that the device outputs work correctly under normal current loading (IOL/IOH). With these tests already done, if the IOS test fails the output is either defective (outside of the design specification) or possibly the output has been preconditioned incorrectly.

When a failure occurs, observe the measured current to determine the cause of the failure. If the test fails the upper limit, the output resistance is too high so not enough current flows. This type of failure can be seen in the datalog on pin 2. Resistance within the test hardware fixture can cause the output resistance to appear too high. A resistor can be used to verify the test system accuracy.

If the test fails the lower limit, the output resistance is too low so too much current flows. This type of failure can be seen in the datalog on pin 4.

An IOS of zero (or near 0) may indicate that the device output is in the wrong logic state; this can be seen in the datalog on pin 5. With the output at logic 0, when the PMU applies zero volts to the output no current flows because the PMU and the output voltage are both at the same potential. This type of failure is usually caused by an incorrect preconditioning sequence. If the device is not properly preconditioned, you must resolve this issue. Once the output is set to the proper logic state the test will most likely pass.

IOS—Key Points

- Purpose: to verify the integrity of the output during maximum current flow
- DC static test uses PMU to force voltage and measure current
- Test limits defined in device specifications
- Dual test limits are often used to test for a range of current
- This test is performed on only one output at a time

DC Test Review

1. The program flow is most important to:
   a) Characterization testing
   b) Engineering evaluation
   c) Production testing

2. What is the purpose of binning?

3. What are some of the items shown on a test summary?

4. What is the information in a test summary used for and who uses it?

5. What is the meaning of "TBD" in a test specification?

6. The current flow for a static IDD test is typically:
   a) A positive current
   b) A negative current
   c) A or B
   d) None of the above

7. In most cases, a dynamic current test will result in:
   a) Less current than a static IDD test
   b) More current than a static IDD test
   c) The same amount of current as the static IDD test
   d) None of the above

8. To perform a static IOL test, the outputs are first preconditioned to the logic 1 state:
   a) True
   b) False

9. When performing an IOH measurement, the current is:
   a) A positive current
   b) A negative current
   c) A or B
   d) None of the above

10. When performing a VOL/IOL measurement, the specified current is 4mA and the specified VOL is 0.4V. What is the maximum value of resistance that the output may contain and still pass the test?
   a) 50Ω
   b) 100Ω
   c) It depends on the value of VDD
   d) None of the above

11. The input leakage test (EL/EH) verifies:
   a) Input capacitance
   b) Input threshold levels
   c) Input resistance
   d) None of the above

12. When performing the IIL/IIH test, what is the advantage of making a parallel measurement?

13. Which test method should not be used when performing the IIL/IIH test on resistive inputs (inputs with pull-ups or pull-downs)?
   a) Serial method
   b) Ganged method

14. The IOZ (high impedance current) test is performed on (circle all correct answers):
   a) Pure input pins
   b) Pure output pins
   c) Three-state output pins
   d) Bi-directional pins
   e) Pins defined under the IOZ test in the device specification

15. Before the IOZ test is performed, the pins to be tested must first be preconditioned to:
    a) A logic 1 state
    b) A logic 0 state
    c) An off state (High Z state)
    d) A bi-directional state

16. To perform the VI (input damp) test:
    a) A negative voltage is forced
    b) A positive voltage is forced
    c) A negative current is measured
    d) None of the above

17. What preconditioning is required to make the IOS (output short circuit) test?

Chapter 7 Verifying Functional Parameters

Objectives:

This section explains:

- An overview of functional testing
- Introduction to Faults and DFT
- What is required to execute a functional test
- Developing input/output signals
- Various functional test methods

Introduction to Functional Testing

There are two separate phases associated with functional testing: first *Functional Verification and characterization*, second *Manufacturing Test*. New chip designs require functional verification to insure that the circuit correctly performs all of its intended functions, this phase of testing is generally referred to as *Functional Verification*. During this phase all logic paths and logic functions are tested and characterized to insure that the device will work properly in its intended application. Vector patterns are used to provide instructions to the device input pins and to predict the states of the output pins. The vector patterns for the first phase of functional testing are based upon how the circuit will actually be used. These patterns are call *Application Vectors*, or you may hear them referred to as *Parallel Vectors*. The term parallel is used
because data is applied to all primary inputs, and all primary outputs are evaluated within each cycle, in parallel.
Upon successful completion of the first phase, the new circuit moves from an engineering environment to a manufacturing environment where screening for manufacturing defects becomes the main focus of test.
In order to simplify test vector generation and improve functional test quality Scan test methodologies have been introduced (explained in more detail later in this chapter).
Scan logic design moves data serially in and out of the device through dedicated test pins. Scan enables a complex circuit to be tested as individual logic blocks which can greatly increase the thoroughness of the test.
The Scan approach differs from the application (parallel) approach in a very fundamental way. An application vector test is designed to verify that the circuit is functional, the circuit is controlled via the primary inputs and correct performance is verified by observing the primary outputs.
A Scan vector test is designed to detect defects within the circuit, it is a defect oriented test developed to prove that the circuit was manufactured defect free. The term *Structural Test* is sometimes used in place of Scan test, particularly when Scan vector patterns are highly sensitized to detect manufacturing defects.
During manufacturing test, it is common to use a combination of application vectors and Scan vectors. The application vectors prove that the circuit functions correctly, the Scan vectors prove that the circuit is defect free.

Detecting Faults

The term Fault Coverage (or fault grading) is used to describe how effective a vector pattern is at detecting and screening out devices that contain *Faults*. Recall from chapter 3, a fault is a flaw that causes the circuit to malfunction. The definition of fault coverage is defined as:

$$\text{Fault Coverage} = \frac{\text{Number of Deteced Faults}}{\text{Total Faults - Undetecable Faults}}$$

It is desirable to achieve the highest fault coverage possible, but it can be quite difficult (or impossible) to achieve 100% fault coverage on complex circuits. A common method of analyzing how effective a vector pattern is at detecting faults involves injecting “Stuck-At-Faults” (SAF) into the circuit model and then observing if the fault can be detected. Here is a quick look at how it works.

The circuit designer, using a simulator, creates a software model of the circuit, simulators are part of the Electronic Design Automation (EDA) toolset. A vector pattern is then created which supplies input instructions to the circuit with the goal of exercising all of the chip logic. The simulator then processes the input vector data through the chip model and captures the output response of the model. The captured output data represents the performance of a good device, this data can now be used to compare against a defective device.

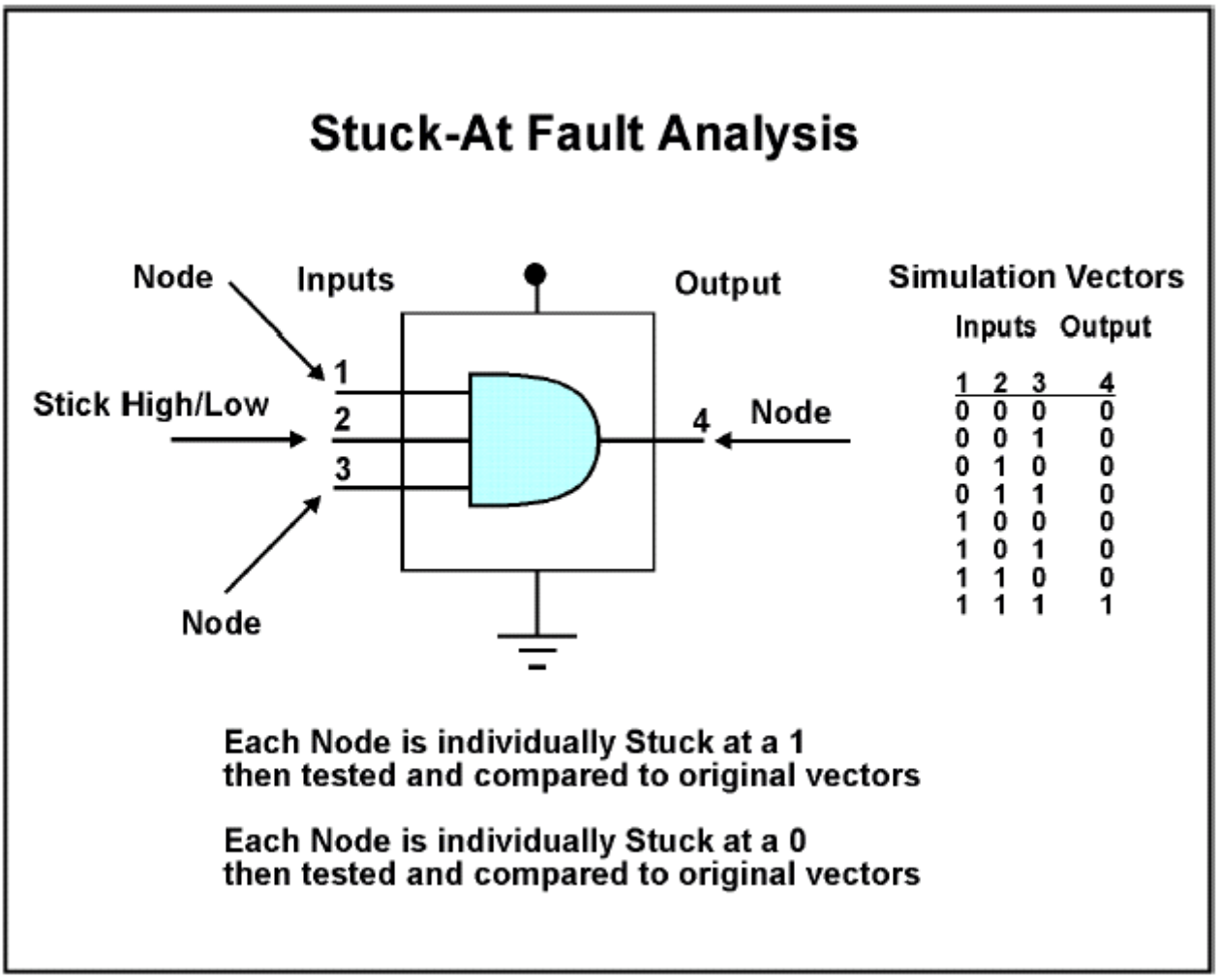


Figure 7-1

Stuck-At Fault Analysis

Figure 7-1 shows a simple AND gate, for this circuit the design engineer will force a stuck-at condition on each individual node of the chip, both externally and internally. A stuck-at condition models either a short to VDD or a short to ground. With the stuck-at condition placed in the model the input vectors are again processed and the output response is compared to the original (good) response. If the output response differs then the fault has been detected, if the response remains unchanged the fault was not detected. This is a simple approach to fault grading and only considers stuck-at 1 or stuck-at 0 faults, many other types of faults may exist, such as opens, bridging, resistive shorts, et cetera. Approaching 100 percent fault coverage can be very difficult to obtain for complex circuits, consideration must be given to simulation time, vector data size and of course cost. The cost of test and the cost associated with shipping defective devices from the factory can be quite high, it makes good business sense to attempt to lower the cost of test and improve fault coverage, with this in mind let's look at the concept of Design For Test (DFT).

What is DFT?

Design for Test (DFT) is defined as logic which is added to a circuit design to improve its testability.
Testability enhancements are made to improve test quality, to improve test generation productivity and to reduce test manufacturing cost. Test logic is also added to a design to help reduce simulation time and Design For Debuggability (DFD). When first silicon arrives, validating the design for the intended functional performance is a major undertaking and the speed at which this task is accomplished is critical to the success of a project. Inserting DFT test logic to help speed the validation and debug process has become a common industry practice.

Ad-Hoc and Structured DFT

Ad-Hoc techniques (non-structured DFT) are device specific, such as adding control and observation test points to a circuit. As an example, logic can be added to shorten the test of a long counter. A 30 bit counter will take 2 to the 30th clock cycles to fully test. At a frequency of 100 MHz, this would take over 10 seconds of ATE test time. Adding test logic to partition and test the 30 bit counter as three 10 bit counters would provide a test socket time of 30 micro-seconds (10 micro-seconds per counter). This will also reduce the simulation time.

Structured DFT techniques are standardized which makes effective use of Electronic Design Automation (EDA) tools for DFT logic insertion, test pattern generation, and verification.

For many years, structured DFT techniques were often rejected due to the negative impact on die size and performance (mainly speed). Structured DFT is now quite common due to the continued increase in design gate count making the task of generating high fault coverage functional vectors overwhelming. Also, advances in semiconductor process technology such as more metal layers and deep sub-micron gate widths have reduced the die area and lessened the performance penalty of adding DFT circuitry.

Additionally, Integrated Circuit (IC) users have increased their requirements for high test coverage, which has also helped speed the acceptance and implementation of DFT.

Types of Structured DFT

There are three common structured DFT methods currently used in chip designs. These are:

- Scan
- JTAG
- BIST

Scan

EDA companies have developed Algorithmic Test Pattern Generator (ATPG) tools, which generate test vectors for stuck at faults. However, sequential (clocked or gated) logic has a significant negative impact on the performance of ATPG tools, this is due to the fact that the state of clocked logic must be initialized to a know condition and then controlled during testing. Complex sequential logic circuits can require extensive conditioning which is both time consuming and expensive in terms of simulation and test time. Scan design is a technique that effectively converts a sequential logic (clocked) design to a combinatorial logic (non-clocked) design allowing ATPG tools to produce high fault coverage test vectors effectively.

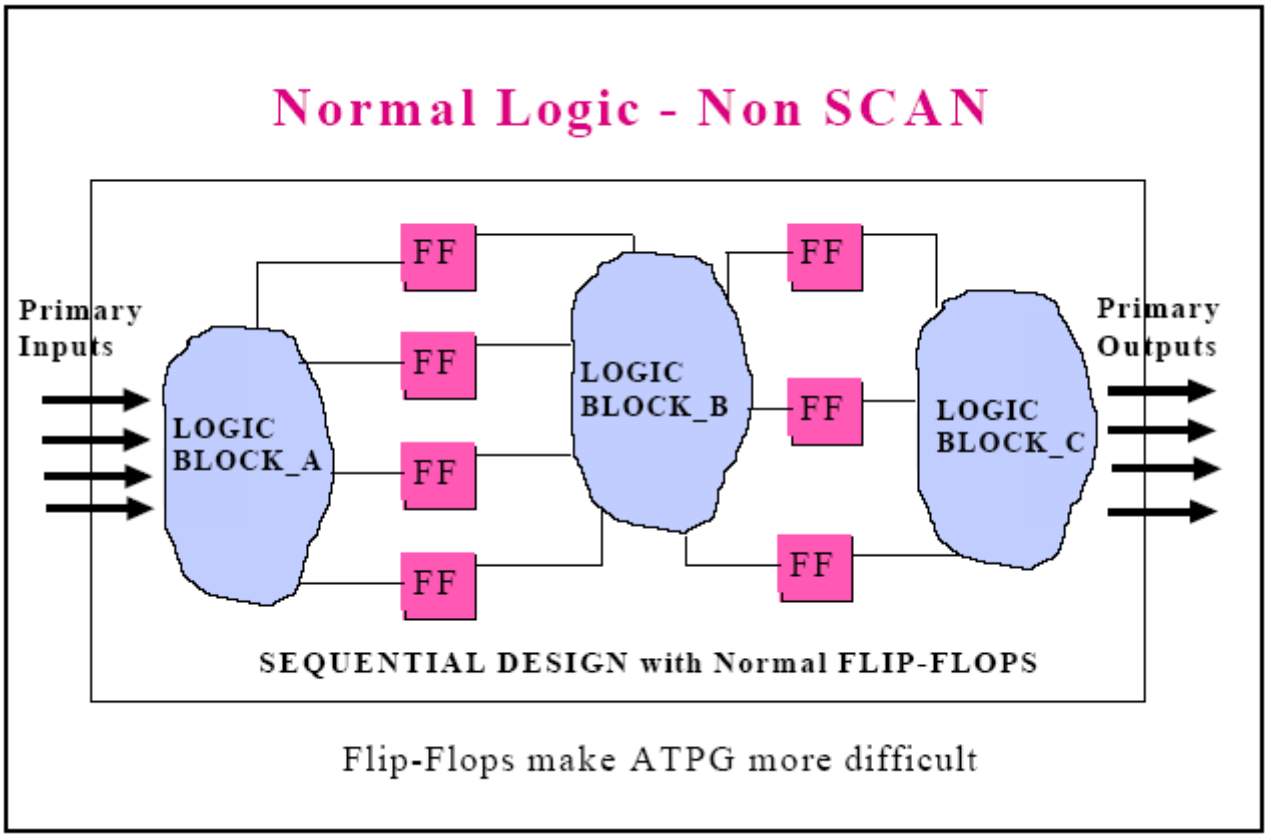


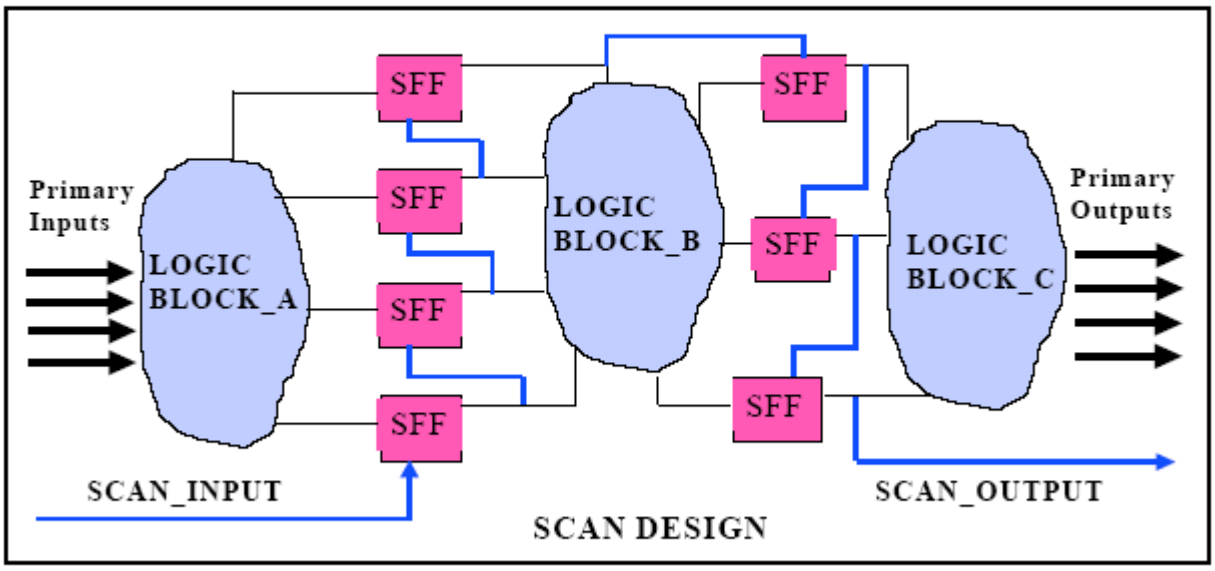
Figure 7-2

Non-Scan Logic

Figure 7-2 illustrates a typical circuit containing several blocks of logic, each with its own function, and the associated sequential logic shown as “FF” (flip-flops). Ideally each logic block will be exhaustively tested for faults. From the diagram you can see that the inputs and outputs of Logic Block B are buried in the core of the circuit and not directly accessible. Input data must pass through Logic Block A and the flip-flops before reaching Block B, the outputs of Block B must pass through flip-flops and Block C before being observed on the primary outputs. This design makes it difficult to achieve high fault coverage with a limited number of clock cycles, additionally any problem along the data path may cause the output to fail, leaving very limited debug capability.

Consider what is required to generate a high fault coverage test on each logic block. The inputs to Block A can be directly accessed through the primary inputs of the circuit, but the output response of Block A must propagate through Block B and C before the output data can be observed. The logic of Block B can be neither accessed or observed directly and the inputs to Block C can not be directly accessed. This circuit would be much easier to test if each logic block could be independently assessed, let’s see how that can be accomplished with Scan logic.

SCAN Implementation



Map normal flops to Scan flops and connect into a large shift register

Figure 7-3

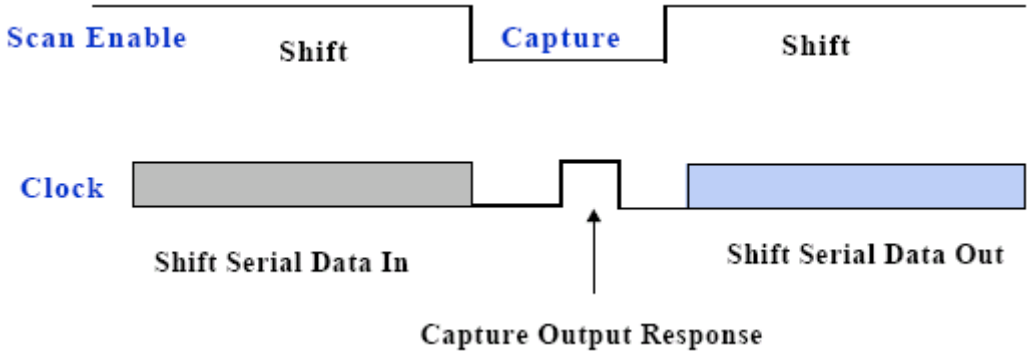
Scan Logic

Scan design is a method which connects all state elements (flip-flops and latches) of a design into one or multiple shift registers which are referred to as *Scan Chains*. Once these Scan chains are configured, they allow both controllability and observability over the entire circuit, assuming a full Scan design. For ATPG purposes, these state elements are considered virtual inputs and outputs.

Scan allows a complex circuit to be partitioned into individual core logic functions, each logic function can be tested separately and all sequential logic elements can be directly controlled. Input data is shifted serially into the device and then applied in parallel to all inputs of the logic block. The outputs of the logic block are captured in parallel, then shifted out serially for evaluation. This method of testing greatly reduces the number of data cycles required to condition and test the logic and improves the diagnostic capabilities of the test.

Two terms used to describe Scan logic are *Full Scan* and *Partial Scan*. Full Scan provides full controllability and observability of the circuit and generally results in very high fault coverage. Full Scan may add as much as a 20 percent increase to the die area. Partial Scan is a trade-off between controllability, observability, fault coverage and increased die size.

SCAN Timing Sequence



Number of Clocks during shift is generally equal to the number of elements in Scan chain minus 1

Figure 7-4

Scan Timing

Figure 7-4 shows the basic sequence of a Scan test. First serial input data is shifted into the device and loaded into the circuit under test. The output response is then captured in parallel and shifted out serially for comparison by the test system.

There are three types of Scan design techniques used, the most popular technique is called *Muxscan*. Muxscan is simply adding a mux in front of a flip-flop or latch to allow a serial data path into the state element. The serial mode is controlled by a test signal, typically referred to as *Scan Enable* or *Scan Select*.

Level Sensitive Scan Design (LSSD) is a technique that was invented (and still used) by IBM. This technique employs two latches and two clocks. LSSD is more clock skew tolerant, but has a higher die overhead penalty.

The third Scan technique is *Dual-Clock* which shares some similarities with LSSD.

Note: For additional information regarding Scan test please reference the Chapter 16 “Principles of Scan Testing”.

BIST Concept

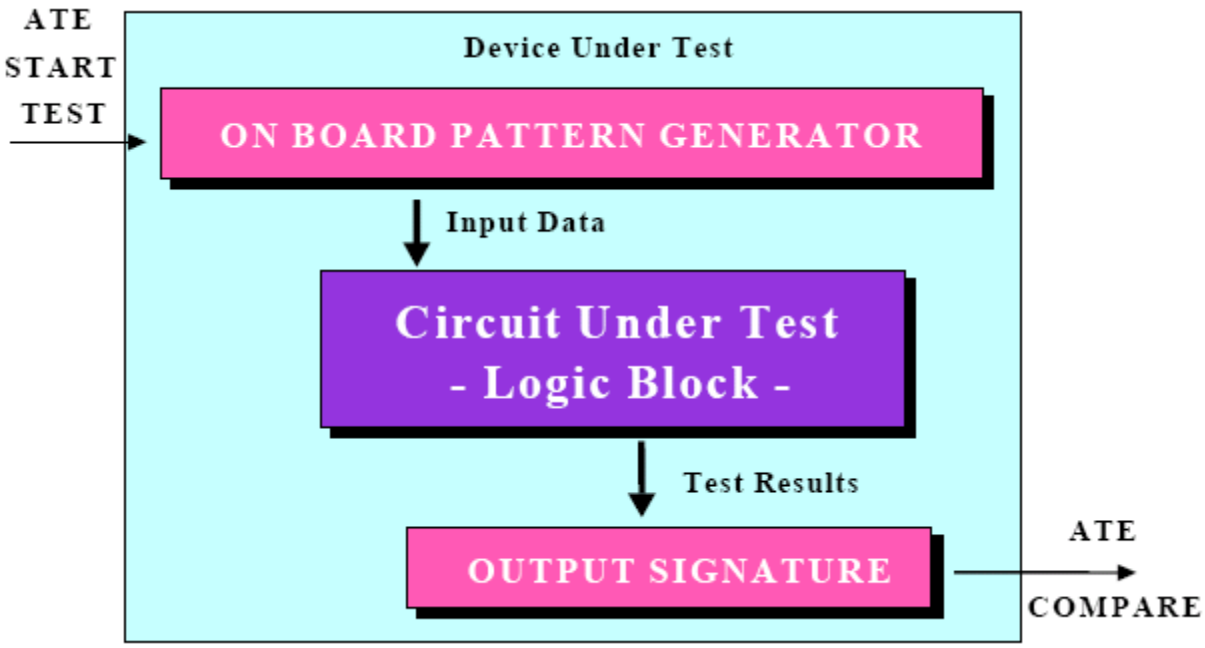


Figure 7-5

BIST

Built-In Self Test (BIST) is test logic contained within the circuit which generates test stimulus and captures device test response data for testing all or part of a circuit design. Memory Bist is quite common and Logic Bist is beginning to get more attention. Bist may require long test times but it generally reduces the ATE vector memory requirements because the built-in pattern generator provides the input data to the circuit under test, not the ATE test system.

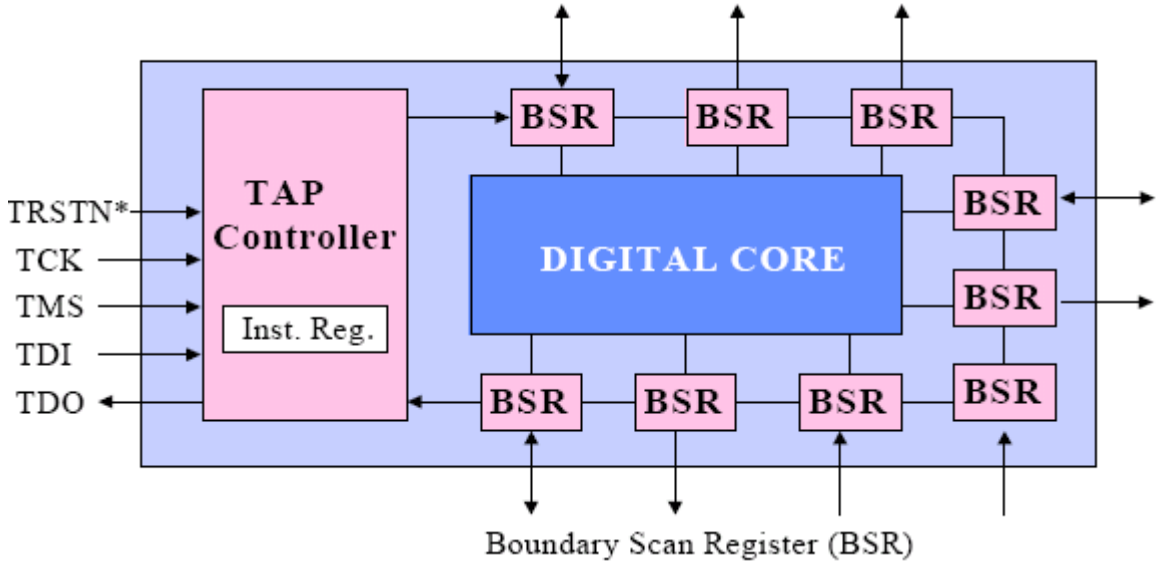
The BIST logic is normally comprised of a pattern generator and a response compactor. A pattern generator such as a counter or Linear Feedback Shift Register (LFSR) can be used to provide the stimulus to a Circuit-Under-Test (CUT). A Multiple-Input-Shift-Register (MISR) can be used to capture the circuit response and then a compressed signature can be shifted out for ATE comparison on one of the device pins. The output data, referred to as the test result *signature*, is generally a sequence of 1's and 0's. When a failure occurs the failing bit(s) within the signature may in some cases be used to diagnose the failure, providing some amount of debuggability and analysis.

The ATE system must provide the proper stimulus to the device to begin the BIST test sequence, upon completion of the BIST test the results are shifted out of the device and compared to the expected response which is stored in the ATE vector data.

A BIST test may run at a higher frequency than the test system, making it possible to test beyond the speed limitations of the ATE system.

BIST is well suited for Embedded Memories, Micro Processors and Micro Controllers. For circuits that incorporate proprietary core components, BIST enable testing the core components without the need to reveal their structure.

JTAG ARCHITECTURE



* TRSTN is optional JTAG Reset pin

Figure 7-6

JTAG

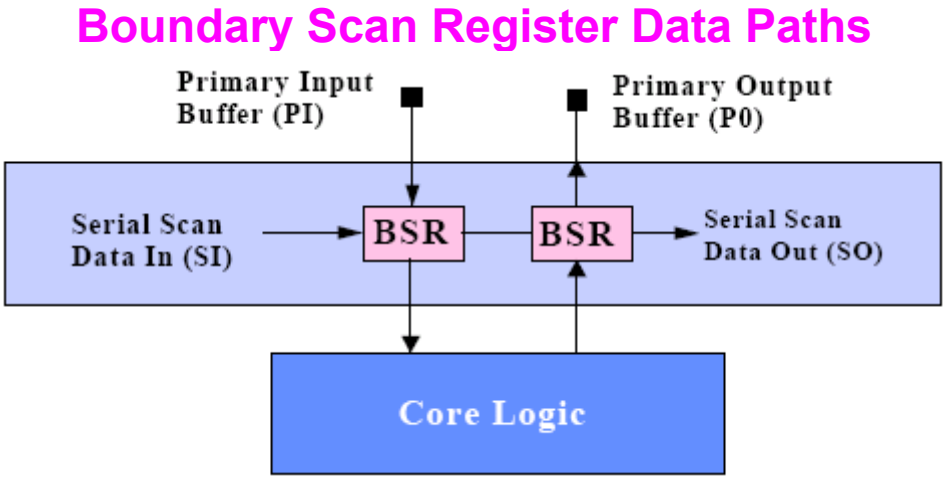
The *Joint Test Action Group* (JTAG) was formed in the mid 1980s to develop an IEEE standard for semiconductors that would allow easier device testing at the board level. In 1990, the IEEE 1149.1 standard was approved. Although this standard was originally developed for board level device testing, it is primarily used for board component interconnect testing (chip to chip connections). It is normally assumed that board components are tested adequately by the manufacturer and that the real need is to ensure that a board is fabricated and assembled correctly. Board level testing also insures that the correct parts are in the correct locations with the correct pin orientation and that no shorts or opens exist on the pin solder connections.

The architecture of the JTAG 1149.1 standard consists of a boundary Scan chain on the device IO ports, a TAP Controller and an Instruction Register, all of which are controlled via a Test Access Port (TAP) consisting of four mandatory and one optional TRSTN (JTAG reset) pin.

The TAP Controller consist of a state machine controlled by the value on the JTAG TMS pin. Various instructions support the different operations required to perform the board level test functions. JTAG instructions can also be encoded to test the device on an ATE system such as BIST and Scan. As previously mentioned, TMS is the Test Mode Select which controls the state machine, TCK is the JTAG clock, TDI is the serial data input, and TDO is the serial data output pin.

It turns out the JTAG boundary Scan logic can be used to simplify testing of the DC specifications such as VIL,VIH, VOL, VOH, and IOZ. Without the use of JTAG, testing these parameters must involve sequencing data through the core logic from the primary device inputs to the primary device outputs. JTAG enables direct control and observation to the IO

buffers without the need of involving the core logic, it is a fast way to set all output high for the VOH test or all outputs low for the VOL test. The behavior of the input buffers can also be directly observed for the VIL and VIH tests.



In normal operation data passes transparently from PI to the core logic and from the core to output PO, Scan paths are not involved.
In Scan mode data is shifted serially from SI to SO and can be parallel transferred to and from the core.
Data can be captured from PI and shifted serially, serial data can also be routed to output PO.

Figure 7-7

Figure 7-7 shows a closer look at the workings of the Boundary Scan Register. From this example it can been seen that a boundary scan register chain adds control and observability to the I/O buffers and the core logic.

DFT and ATE Considerations

DFT methodologies help automate test program development and tremendous aid for debugging new circuit designs with first silicon. Below are listed some of the key issues for DFT and ATE consideration.

- Scan, BIST and JTAG tester timing is generally very simple. Timing signals are only needed on clock pins.
- ATPG Scan vectors can easily consume functional Scan vector memory. ATE systems generally have a dedicated Scan memory with a maximum Scan chain length. Make certain that the test system has adequate resources to store all of the Scan vector data.
- Implementing multiple Scan chains and testing the chains in parallel will reduce the test time.
- BIST requires very little ATE vector memory. It does however require many clock cycles, this can be addressed using the ATE's "repeat" vector opcode, therefore minimizing vector memory requirements.
- JTAG can be used for testing the VIL,VIH, VOL, VOH, and IOZ DC specifications.
- DFT logic improves fault coverage and simplifies vector generation.
- DFT logic improves observability and debuggability.
- DFT logic adds to the total die area and may result in lower yields.

Basic Terms

Functional testing introduces a few new terms: (a complete list of terminology is located in the Glossary.)
Test Patterns A representation of the states of inputs and outputs for the various logical functions that the device is designed to perform. Input data is supplied to the DUT by the test system. Output pattern data is compared against the response from the output pins of the DUT. During a functional test the test patterns are executed or applied to the DUT. In the event that the expected output data does not match the output data from the DUT a functional failure occurs. Test patterns are also called test vectors or truth tables. Test vectors are often represented as a sequence of ones and zeros or other characters which represent logical levels.
Signal Format A means of describing the wave shape of an input signal supplied by the pin electronics driver circuitry. Example: NRZ/DNRZ/RZ/RO/SBC.
Output Strobe An output *Strobe* is the timing marker within the test system that is used as the timing reference for output signal evaluation. Many test systems offer an individual strobe marker on each tester channel, this allows each output signal to be evaluated independently.
Output Sample Time The point in time at which the output signal of a DUT is evaluated during a functional test. The comparator circuitry in the Pin Electronics qualifies the DUT output voltage to a pre–defined logic 1 (VOH) or logic 0 (VOL) reference level. The test system then makes a pass/fail decision at a specific point in time, which is defined by the strobe placement timing. Output sampling is also called output strobing.
Output Mask A method of enabling or disabling an output comparison for a tester channel during a functional test. This can be performed on a pin-by-pin basis for each test cycle.

Functional Testing

Functional testing verifies that the DUT will correctly perform its intended logical functions. To accomplish this, test vectors or truth tables must be created which can detect faults within the DUT. The test vectors, combined with the test timing, make up the heart of the functional test.
All aspects of the DUT's performance must be considered when developing the functional test. The exact values for the following items must be carefully examined.
VDD Min/Max — DUT power levels
VIL/VIH — input levels
VOL/VOH — output levels
IOL/IOH — output current loading
VREF — IOL/IOH switching point
Test Frequency — cycle time used for test
Input signal timings — clocks / setups / holds / controls
Input signal formats — wave shapes of input signals
Output timings — when will outputs be sampled within cycle
Vector sequencing — start / stop points within a vector file
It can be seen from the above list that the majority of the test system's resources must be used during a functional test.
All functional tests consist of two distinct components, the test vector file and the instructions contained within the main test

program. The test vector file represents the input and output logic states needed to test the DUT. The test program contains the information needed to control the test hardware in a manner that will create all the necessary voltages, wave forms and timings.

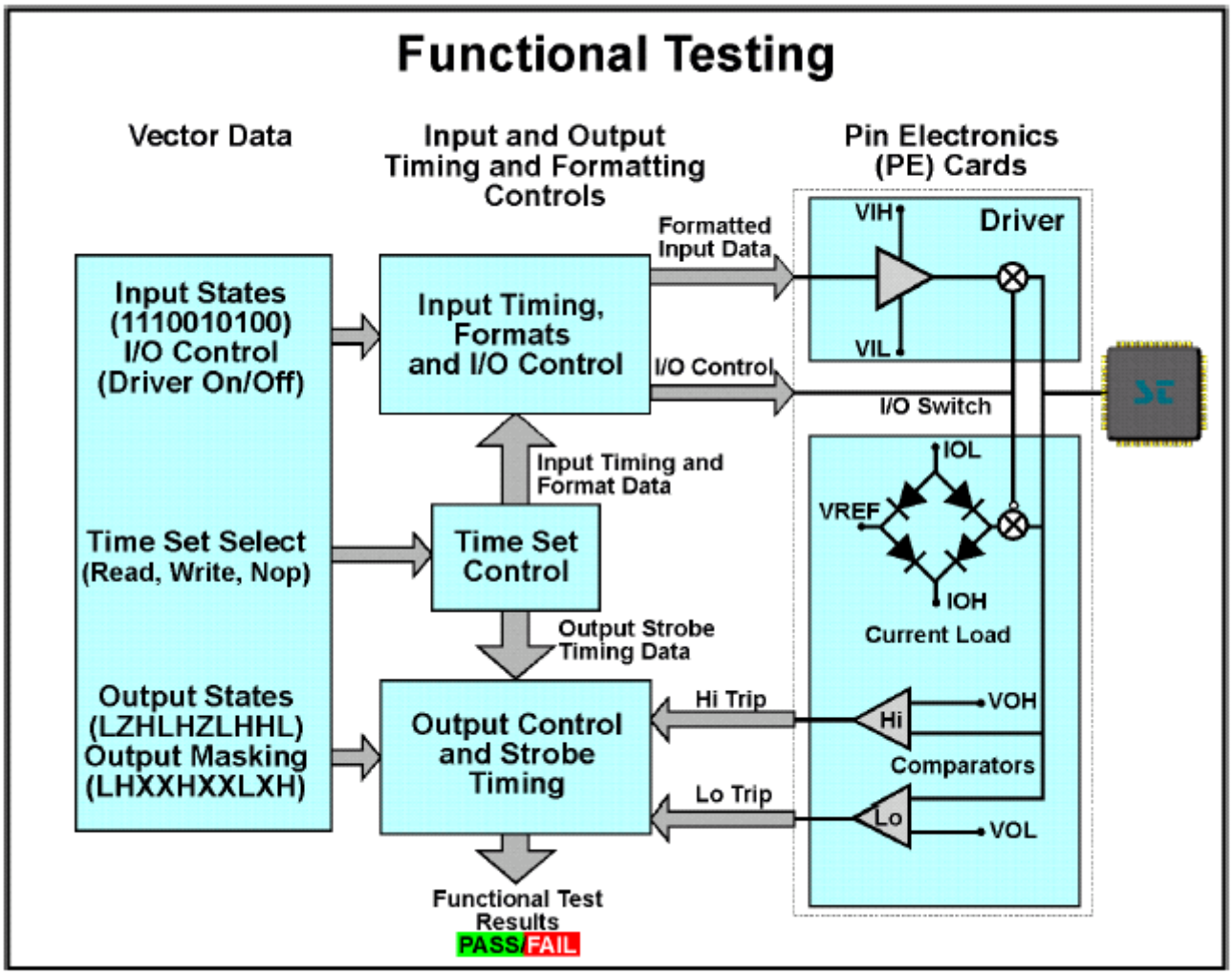


Figure 7-8

As the functional test executes, the test system supplies input data to the DUT and monitors the DUT outputs on a cycle by cycle, pin by pin basis. If any output pin fails to meet the expected logic state, voltage or timing, the result of the functional test is a failure.

The Test Cycle

The test cycle, also called the test period, is based on the operating frequency of the device and defines the time duration of one test vector. The test cycle time can be determined by the formula:

Cycle = 1/frequency.

The start of each new cycle is called time zero or T0. When developing timing for a functional test, the first step is to determine the test cycle timing.

Input Data

Input data is created by combining:

- Test vector data (instructions or stimuli to the DUT)
- Input signal timing (signal transition points)
- Input signal formats (wave shapes)
- Input levels (VIL/VIH)
- Time set selections (if more than one time set is used)

Input data in its simplest form consists of a logic 0 or logic 1 level stored as test vector data. The voltage levels which represent a logic 0 or 1 are produced at the test head by the VIL/VIH reference voltages.

Many input signals require more complex data containing unique formats (wave shapes) and timings (edge placements). This information is contained in the main test program and is controlled through the format and timing statements of the test language.

Some test systems have shared resources which means that only a limited number of input timings, formats and levels can be supplied by the test hardware at one time. A test system with a tester per pin architecture makes programming much easier because each pin can be programmed with unique timings, formats and levels.

Input Signal Formats

Signal formatting is very important—when properly used they guarantee that all AC parameters are tested to specification. Signal formats, when combined with vector data, edge placements and input levels, define the wave shape of input signals to the DUT. Figure 7–9 is a brief description of each illustrated signal format. Become familiar with all of them.

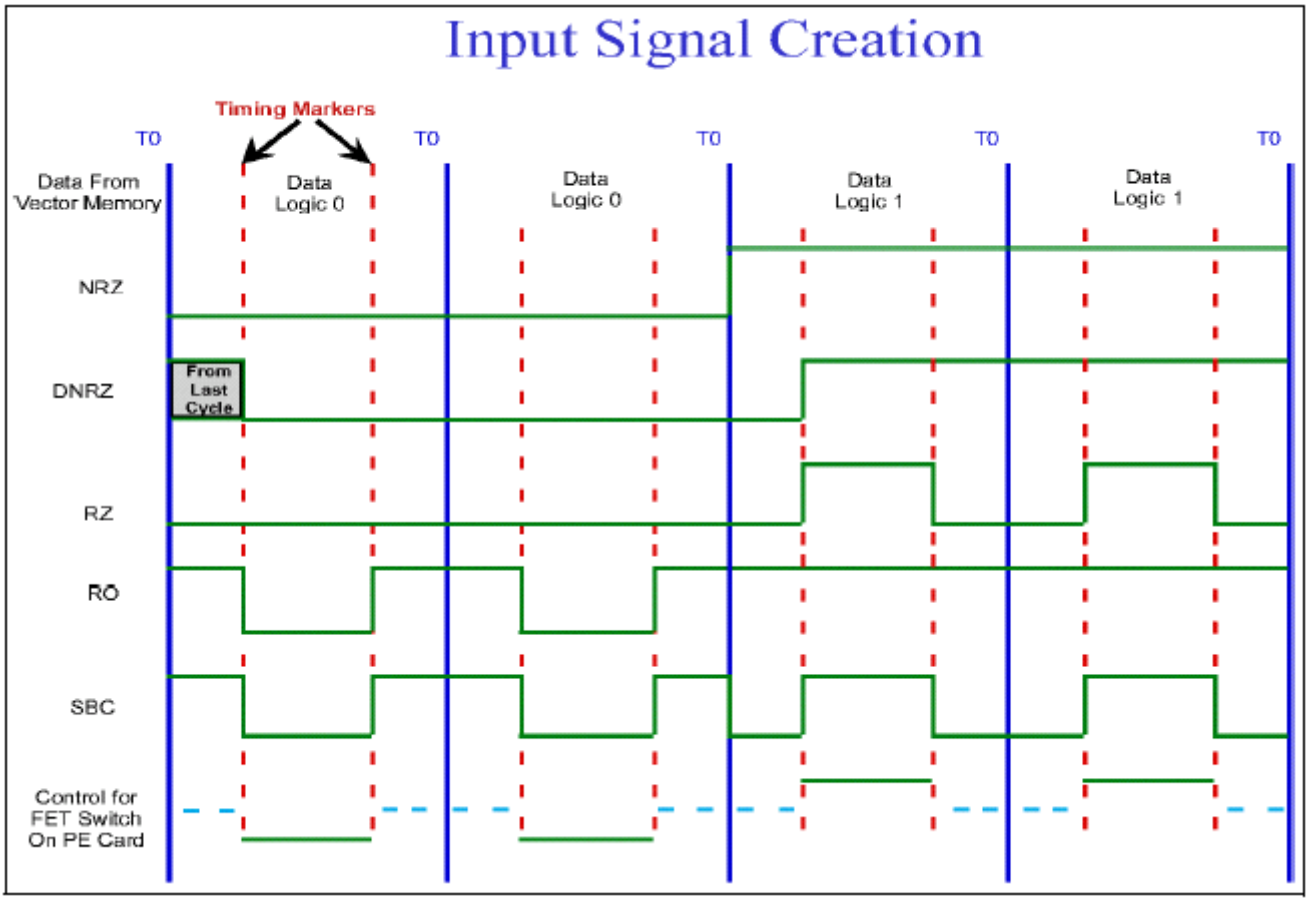


Figure 7-9

NRZ Non Return to Zero represents the actual data stored in vector memory and contains no edge timing. NRZ data changes only at the beginning of each cycle (T0).The signal stays high or low for the entire cycle.

DNRZ Delayed Non Return to Zero represents the data stored in vector memory, but the point within the cycle where the data makes a transition is defined to be a value other than the start of the cycle (T0). DNRZ data will change after a pre–defined delay period only if the vector data has changed between the current cycle and the previous cycle.

RZ Return to Zero provides a positive pulse when vector data is logic 1 and no pulse when vector data is logic 0 (the signal remains at logic 0). RZ signals have a leading (rising) edge and a trailing (falling) edge. This signal format can provide a positive clock when all vector data for the pin is logic 1. Active high signals such as CS2, as defined in the Ram data sheet, need RZ format.

RO Return to One provides a negative pulse when vector data is logic 0 and no pulse when vector data is logic 1 (the signal remains at logic 1). RO signals have a leading (falling) edge and a trailing (rising) edge. This signal format can provide a negative clock when all vector data for the pin is logic 0. Active low signals such as output enable (OE/), as defined in the Ram data sheet, use RO format.

SBC Surround By Compliment can provide three edge transitions per cycle. This signal format creates a complex signal based on the vector data. It inverts the data at the start of the cycle (T0), waits a predefined delay, presents the actual vector data for the specified pulse width, then inverts the data again for the remainder of the cycle. This signal format is the only format that will guarantee both setup and hold time in a single execution of the test vectors. SBC format is also known as *Exclusive–OR* (XOR) format.

Note: Setup and hold time parameters can be verified on test systems which do not support SBC formats by executing the test vector sequence twice. On the first run, define setup/hold pins in RZ format to test each time a logic 1 appears in vector memory. Redefine the pins as RO to test each time a logic 0 appears in vector memory for the second vector run.

ZD Z (Impedance) Drive/no drive allows the input drivers to turn on and off within a cycle. When the driver is off the tester channel is in the high impedance state; when the driver is on the DUT input will be driven to a logic 0 or 1 depending on vector data. This format is sometimes referred to as *Drive/Inhibit*

Developing Input Signal Timings

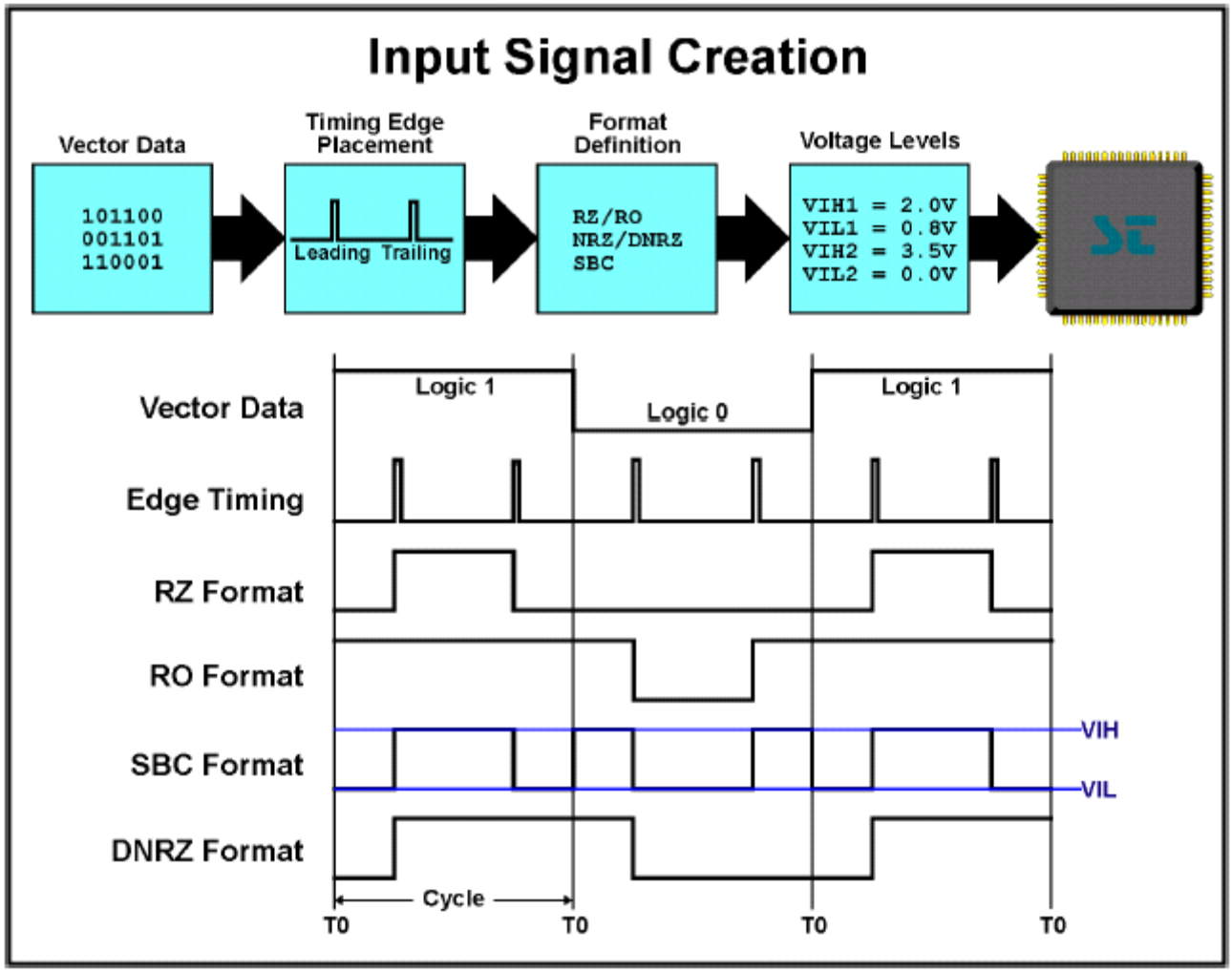


Figure 7-10

Once the cycle time has been determined, the placement of the control signals within the cycle can be defined. There are generally two types of input signals—control signals and data signals. Data signals provide data to the device while the control signals determine the point in time when data signals will be read or latched into the internal logic of the device.

First determine the active edges of the control signals and the amount of setup and hold time required for the data signals.
This information will help define the edge placement (timing) of each input signal within the test cycle.

Next determine the signal format required for each input signal. Clock signals are usually RZ (positive pulse) or RO (negative pulse) formats. Active high control signals such as CS (chip select) or READ are often RZ format. Active low control signals such as CS/ (chip select bar) or OE/ (output enable bar) are often RO format. Data signals that have a setup and hold time parameters require SBC formats. Other inputs may require NRZ or DNRZ formats.

Inputs signals are created by combining data from several areas within the test system. The waveform at the test head is a result of the test vector data, edge placement timing, format definition and VIL/VIH values as shown in Figure 7-10.

Output Data

Outputs are tested by combining:

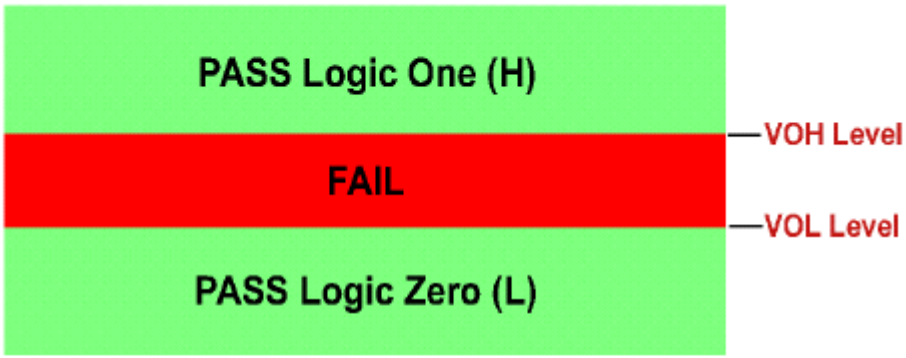
- Test vector data (expected logic states from the DUT)
- Output strobe timing (when to sample outputs within the test cycle)
- VOL/VOH (reference levels to determine output states from DUT)
- IOL/IOH (output current loading)
- Time set selections (if more than one time set is used)

Testing Outputs

During a functional test the voltage level of the output signals from the DUT are compared to the VOL and VOH reference levels by the functional comparators. An output strobe is assigned a timing value for each output pin to control the exact point within the test cycle for sampling the output voltage.

The test vectors contain the expected logic states for each pin. If the expected state is a logic zero (L), the DUT output must be equal to or less than the VOL reference level when the output strobe occurs. If the expected state is a logic high (H), the DUT output must be equal to or greater than the VOH reference level. Some test systems also have the ability to test for a high impedance state (Z) which is defined as greater than VOL and less than VOH.

Functional Testing of Valid H/L Output Levels



Comparator logic for valid output levels

DUT Output voltage must be equal to or greater than VOH to pass a High

DUT Output voltage must be equal to or less than VOL to pass a Low

Figure 7-11

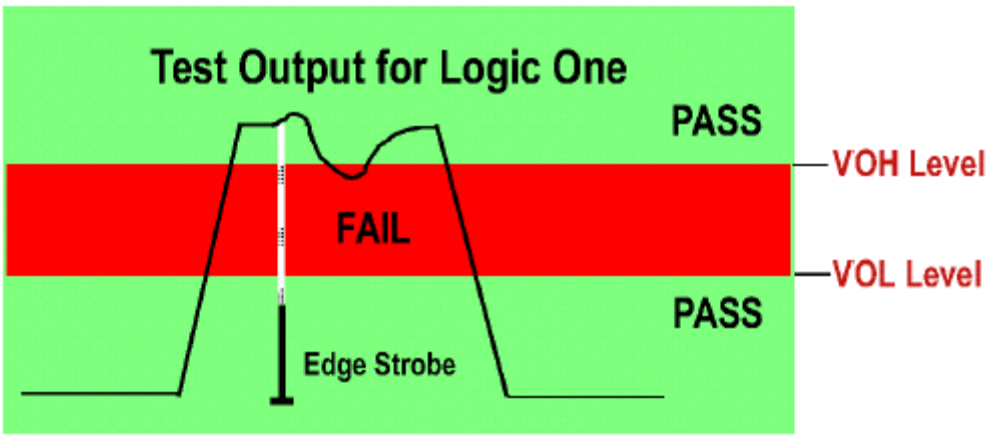
Testing Valid (L/H) Output Levels

Figure 7-11 shows the pass/fail/pass relationship between the DUT output and the VOL/VOH reference values for testing valid (normal) output levels.

The output voltage produced by the DUT must be equal to or greater than the VOH reference level of the comparator in order to qualify as a valid output high.

The output voltage produced by the DUT must be equal to or less than the VOL reference level of the comparator in order to qualify as a valid output Low.

Testing Outputs with an Edge Strobe



Result of test is a PASS
Edge Strobes are a single point in time
Pass/Fail evaluation made only at strobe time
Strobe timings are referenced to Time Zero

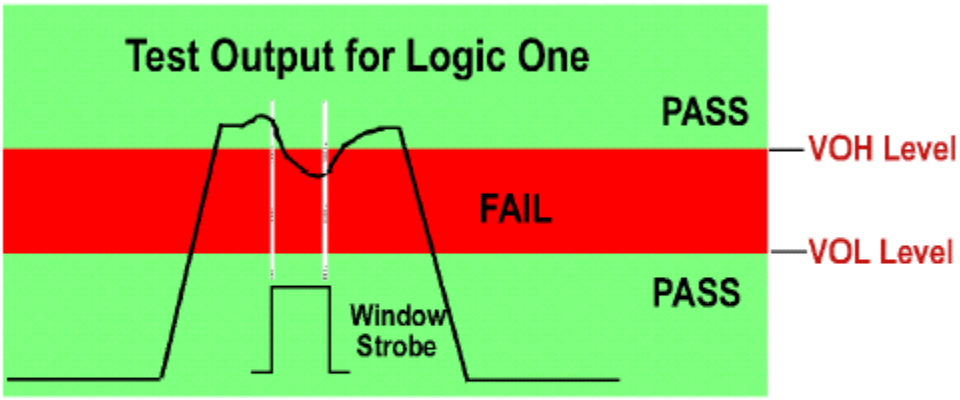
Figure 7-12

Output Testing using an Edge Strobe

The voltage value of a digital output signal is evaluated at a particular point in time to determine the logic state of the output. An output *Strobe* is the timing marker within the test system that is used as the timing reference for output signal evaluation. Many test systems offer an individual strobe marker on each tester channel, this allows each output signal to be evaluated independently. There are two types of strobe markers commonly available: edge and window.

Figure 7-12 shows the placement of an edge strobe and the point of output evaluation. In this example the output signal is being tested for a logic one. As shown above the output voltage is greater than the VOH reference level at the time that the edge strobe occurs, in this example the result of the test is a PASS. An edge strobe makes an evaluation at a single point in time. Edge strobes are often used in high frequency testing because they are less affected by noise or ringing of the output signal. The exact placement of the output strobe is determined from information contained in the device AC specification, but strobe timing is programmed relative to T0 (time zero of the tester cycle)

Testing an Output Using A Window Strobe



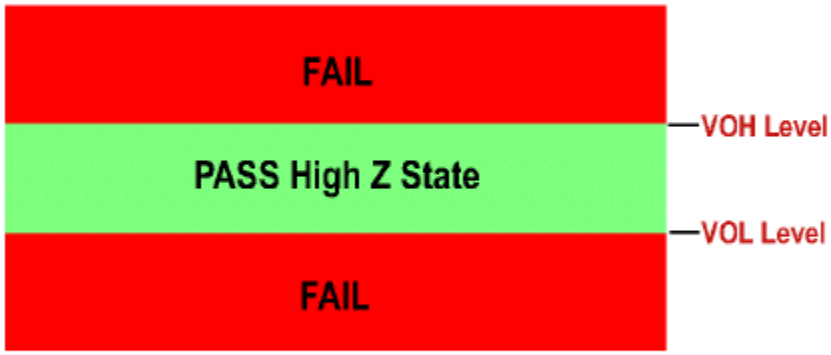
Result of test is a FAIL
Output voltage dropped below VOH level
PASS level must be maintained during entire "Window"
Window Strobes are sensitive to noise

Figure 7-13

Output Testing using a Window Strobe

Figure 7-13 shows the placement of a window strobe and the duration of output evaluation. In this example the output signal is being tested for a logic one. Notice that the output voltage falls below the VOH reference level during the time that the window strobe is active, the result of this test is a FAIL. A window strobe makes the evaluation during the entire strobe width. The test will fail if the output voltage fails to correctly qualify at any point during the strobe window timing. Ringing or noise on the output signal can cause the test to fail when using a window strobe.

Functional Testing of High Impedance Levels



Comparator logic for High Impedance Levels

Figure 7-14

Testing High Impedance (Z-state) Output Levels

Outputs can be functionally tested for a high impedance condition. In this type of testing, the comparator logic is inverted to expect a non valid logic level (not 1 and not 0). A high impedance state (or level) is defined as a voltage that is greater than the VOL reference level and less than the VOH reference level (see Figure 7–14). A voltage external to the DUT is required to pull the high impedance output to a non valid (intermediate) voltage. This is accomplished through the use of a load connected to a reference voltage. A 1.5 volt or 2.0 volt reference is often used to represent the intermediate or high- Z level. Remember, when an output enters a high impedance state, it loses the ability to supply voltage and current. A high impedance output will tend to stay at its last valid logic level unless something external to the device causes the output to change. The test system load is used to pull the DUT output to the specified reference voltage.

Figure 7-14 shows the fail/pass/fail relationship between the DUT output and the VOL/VOH reference values for testing high impedance output levels.

Output Current Loading

Current loads may be applied to the DUT outputs during a functional test. Programmable current loads (also called dynamic current loads) consist of circuitry located in the pin electronics and their voltage and current settings are defined within the test program. If the test system does not support programmable loads, resistive loads may be added to the external test hardware. Current loads apply the proper IOL and IOH currents to the outputs as the functional test executes.
By applying the specified IOL/IOH currents and testing for the specified VOL/VOH voltages, the output current and voltage parameters can be verified during the execution of a functional test. This is much faster than performing the same tests using the PMU.

Developing Output Strobe Timing

Output signal transitions are often controlled by a clock or control signal edge. To fully understand this you must review the device timing diagram and determine the active edge of the clock or control signals which cause output signals to change. Determine the amount of propagation delay time needed before the output reaches a valid logic level. This point within the cycle is where the output strobe should be placed for that particular signal.

The output strobe can be a point in time or a window in time depending on the test system hardware capabilities. When the output strobe occurs the output signal for the DUT is sampled. The signal must be equal to or greater than the VOH voltage if the test vector defines the expected output as logic 1. The signal must be equal to or less than the VOL voltage if the test vector defines the expected output as logic 0.

As a general rule it is best to define test timing so that outputs transition and are tested within the same cycle. This allows propagation delays to be accurately measured without crossing test cycle boundaries. Make certain that outputs have sufficient time to propagate out before the end of the test cycle. Some device output pins take longer than others to reach their final value; testing at a reduced frequency can expose propagation delay problems. Also be aware that some test systems have limitations regarding how close an output strobe can be placed to the beginning or end of the T0 test cycle boundary.

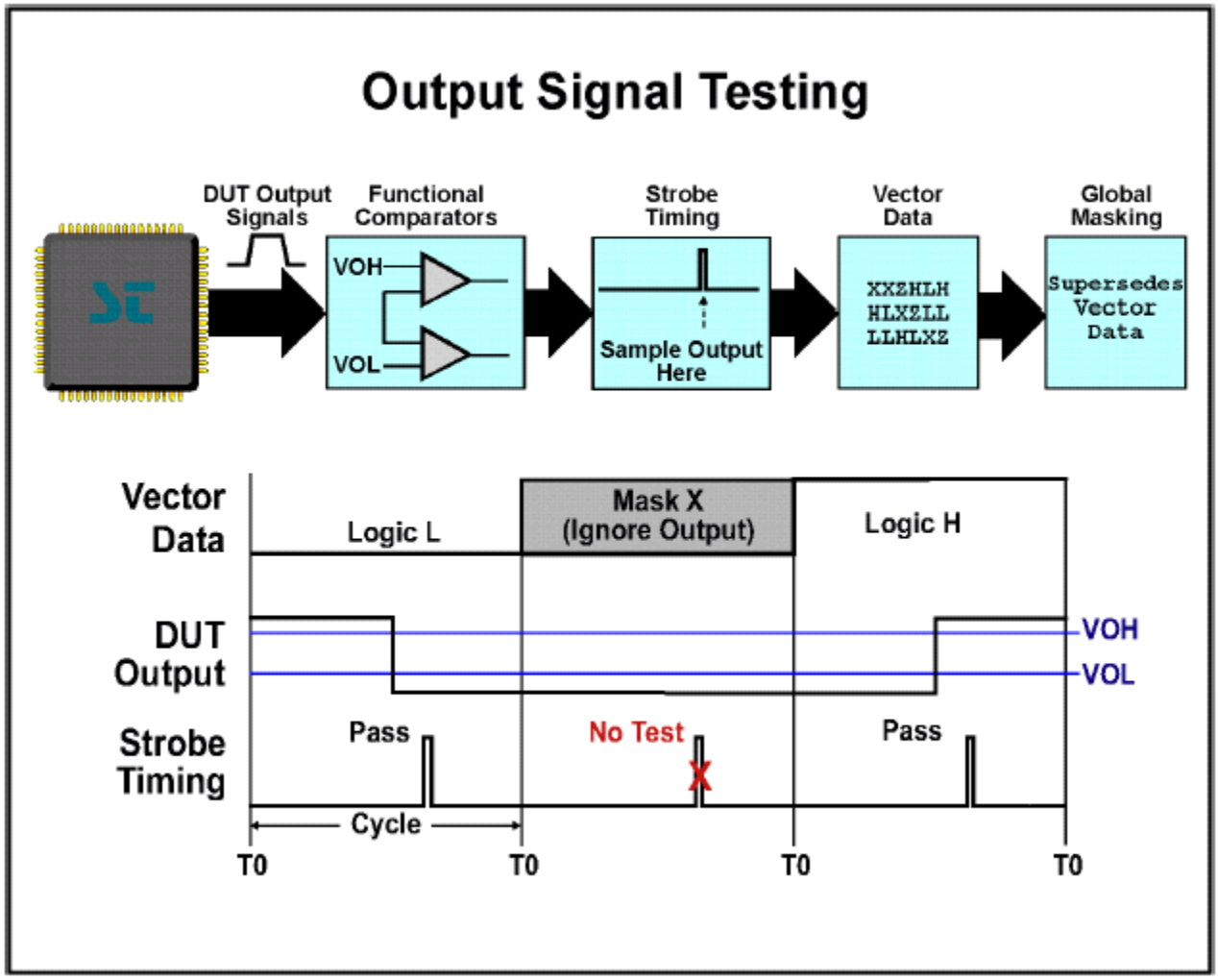


Figure 7-15

As illustrated in Figure 7-15, a combination of factors influence exactly when and how an output signal is tested:

. The vector data determine the expected logic state (L/H/Z/X).

. The VOL/VOH reference levels qualify the DUT output voltage.

. The output strobe timing defines the point within the cycle at which the output signal is evaluated.

. The output compare mask controls whether the result of the test will be used to make a pass/fail decision or if the result will be ignored. The output compare mask can be used in 2 ways. The character “x” is used to select on a pin by pin, cycle by cycle basis whether to test or ignore any given output pin. Many systems also have a Global Mask or a Master Mask which will override the data contained in vector memory.

Output Loading for AC Tests

A device specification may indicate that a current load must be placed on the output of the device when performing the AC timing tests. These loads are often resistor/diode/capacitance networks which simulate loading conditions of circuitry which will be connected to the device in its final application (e.g. in a computer or cellular telephone). This type of load is most often associated with TTL circuits.

Examine the AC load in Figure 7–16. As a starting point VCC is set to 5.0V and nothing is connected at point A. Under these conditions point B will be approximately 2.1V (0.7V dropped across each diode) and point A will also be at 2.1V. The voltage seen across RL is 2.9V (VCC-2.1), therefore 1.45mA of current will flow through RL and the three diodes to ground.

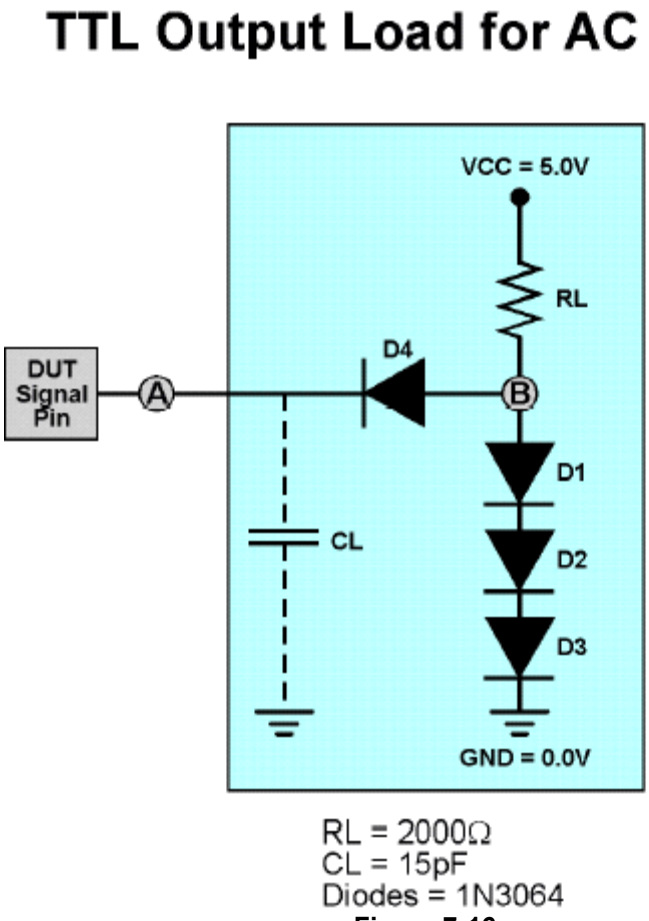


Figure 7-16

Next, a device output driving logic 0 (0.4V) is connected to point A. This forward biases diode D4, pulling point B to 1.1V (0.4V plus one 0.7V diode drop). There is now 3.9V across RL and the current flow through RL into the device output is 1.95mA, loading the device output when driving a logic 0.

When the device output drives a logic 1 (2.4V) D4 becomes reversed biased and eliminates the current loading effect. For this example the AC load provides a current load only for a logic 0; when the device drives a logic 1 the load is essentially removed.

Note: The test fixture will generally provide more capacitance than 15pF, so for load capacitance specifications less than 20–30pF (depends on the tester) the CL load capacitors are not physically added to the AC test load.

Vector Data

The test vector file contains the truth table which exercises the various functions the DUT is designed to perform. The vector file contains the logic states which must be applied to the DUT inputs and also the logic states which are expected to appear on the DUT outputs. Vector data often consists of the following set of characters.

Vector Characters	**Driver State**	**Comparator State**	
Pin State			
0 = logic 0	driver on	comparator off	input
1 = logic 1	driver on	comparator off	input
L = logic 0	driver off	comparator on	output
H = logic 1	driver off	comparator on	output
Z = float	driver off	comparator on	output
X = don't care	driver off	comparator off	ignore

The vector file may also contain instructions to the test system hardware. If the DUT has I/O pins (pins that act as both inputs and outputs) then the vector file must control when the input driver circuitry turns on and off. The I/O switching can occur on a cycle by cycle basis, changing a DUT pin from an input to an output, or from an output to an input.

The test vectors may contain masked output pins. A mask is used to control the testing of an output pin. When a DUT output is in a known logic state it can be tested, but there may be occasions when the output is in an unknown or don't care state. A mask is used to ignore the pass/fail result of an output. Masking is generally available for each individual pin and can be selected or deselected on a cycle by cycle basis.

If the tester supports multiple time sets the vectors may contain time set information. Multiple time sets are used to change the test timing as the vectors are being executed. For example, when testing a RAM it typically takes less time to write data into the RAM than to read data from the RAM. In this case there may be one time set that contains write timing and one time set that contains read timing. The vectors will contain control statements to select the appropriate time set for the appropriate vector functions (reading or writing). Time sets can control cycle times, input timings and formats and output strobe timings. See Chapter 10 *Test Vector Development* for more information on test vectors.

Executing a Functional Test

The following steps are required to execute a functional test:

1. Define VDD level
2. Define input drive and output reference levels (VIL/VIH/VOL/VOH)
3. Define output current loading (IOL/IOH/VREF)
4. Define test cycle time
5. Define input timings and formats for all input pins
6. Define output strobe timings for all output pins
7. Define start and stop locations for vector memory
8. Execute the test

Functional Specifications

Two methods are commonly used to functionally verify device specifications. In the first method, all input, output and timing parameters are set to their worst case conditions and the functional vector sequence is executed. This approach is fast and guarantees that the device meets the design specifications. However, if a failure occurs it is not apparent which parameter caused the failure.

An alternate approach is to test the parameters individually. For example, only set VIL/VIH to the values defined in the device specification, with other parameters relaxed. If a failure occurs, it is immediately known that the cause was either the VIL level or the VIH level. The testing continues until all parameters have been verified. This approach offers more detailed information regarding yield issues but increases test time.

Relaxed Parameters

When a parameter value is relaxed its value is adjusted in a way that will make it easier for the DUT to function properly. For example, if VIL is specified as 0.8V it could be relaxed by setting the value to 0.4V. By lowering the value of VIL it becomes easier for the DUT to detect the input voltage as a logic 0. To relax inputs, lower the VIL levels and raise the VIH levels. To relax outputs, set both VOL and VOH to 1.5V, the comparators will then sense any output voltage below 1.5V as a logic 0 and any output voltage above 1.5V as a logic 1. *Note:* output comparator levels can not be relaxed when testing for “Z” state levels. To relax timing values, reduce the test rate, increase setup and hold times and increase output propagation delays.

Timing Parameters

AC timing specifications are verified by presenting the appropriate wave forms to the device under test. Setup times, hold times, minimum pulse widths and propagation delays must be tested. In some cases it may be possible to apply all worst case conditions at once and guarantee that the device meets the complete device specification with only one test execution. Complex functional timings may require several test iterations with different conditions to guarantee the complete device specifications.

MIN/MAX Voltages

Device specifications often define the operating voltage range of VDD. For instance VDD=5.0± 10% indicates that for a VDD of 5.0V the device must function between 4.5V and 5.5V. This voltage range is often referred to as VDDMIN and VDDMAX. Functional tests must be executed over the entire range of the device specification, so the functional test vector sequence must be executed twice, once with VDD set to VDDMIN and again with VDD set to VDDMAX. Some device parameters (VIL/VIH/VOL/VOH) may be defined as a percentage of VDD. When this is true be sure to adjust these parameters when modifying the value of VDD.

Gross Functional Tests

The term gross functional test refers to performing a functional test with relaxed conditions. Frequency, timings, voltages and current loading are generally relaxed. This test may also be called a basic functional test or a wiggle test.

Why Perform a Gross Functional Test?

Performing a gross functional test indicates whether or not the device being tested is functionally “alive.” These conditions should afford the best possible opportunity for the device to function correctly and to pass the functional vector set. When the test program is initially being developed, the gross functional test conditions are often used to verify correct functionality of the entire vector set. The gross functional test is often executed early in the test program flow and is used to verify correct functionality of all test vectors which will be used throughout the entire test program.

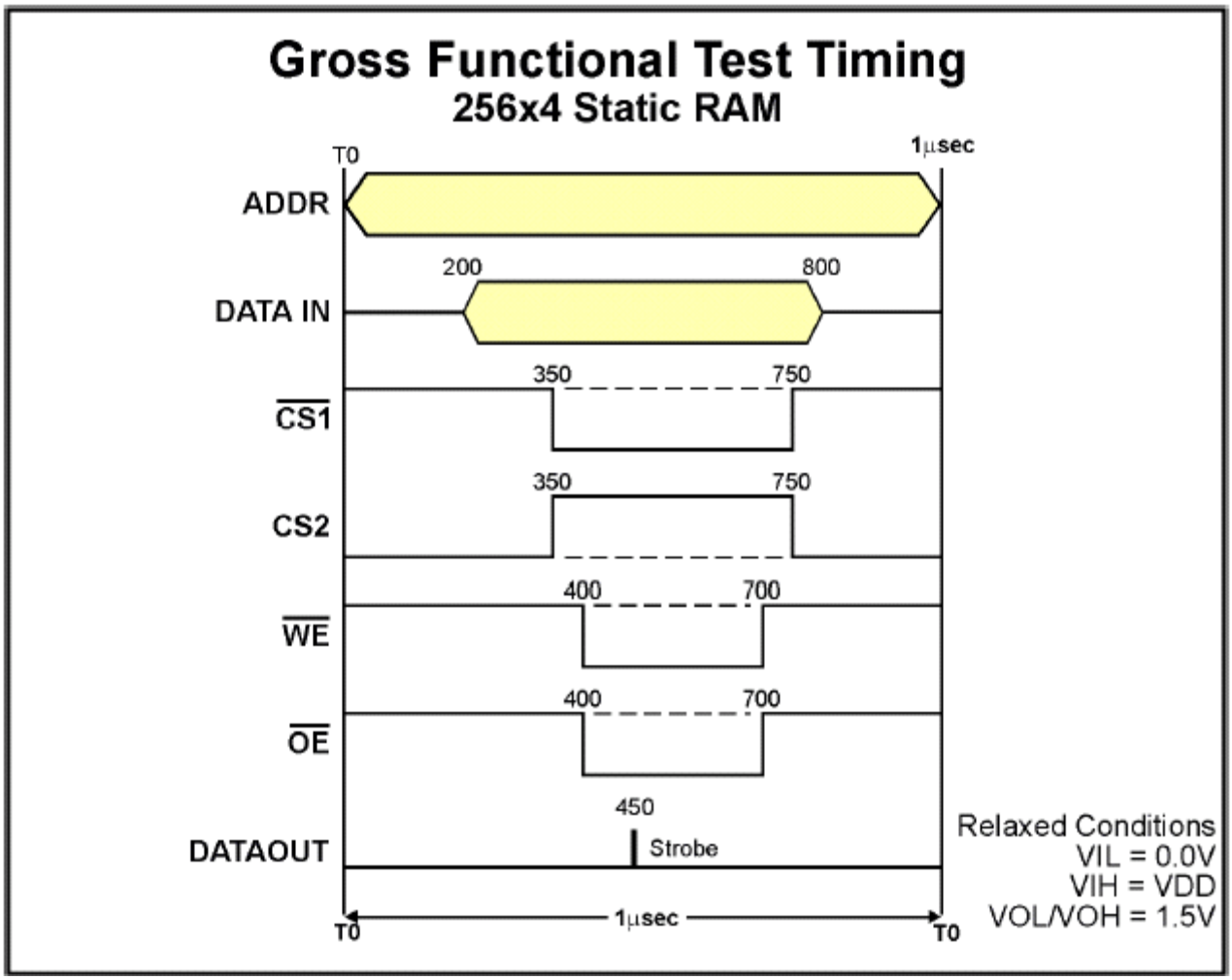


Figure 7-17

Gross Functional Test Method

Figure 7-17 represents the gross functional timing for the 256 x 4 static RAM. The RAM AC specification shown in chapter 8 states that the device operating frequency is 66MHz. The gross functional test is performed with relaxed input levels, output levels and timings. This example shows that the test frequency has been relaxed from 66MHz to 1MHz. All setup times, hold times and propagation delays are also relaxed. It is necessary to maintain the same relative edge placements of all signals when relaxing the functional timings. This example also shows the relaxed values for VIL, VIH, VOL and VOH.

Once the device has passed the gross functional test, a more rigorous test may be executed to insure that the device meets all its specifications. As a means of increasing test throughput, the gross functional test may be executed conditionally when a more rigorous test has failed. The gross functional test will often provide valuable yield information when used in production testing.

Gross Functional Test - Key Points:

- Purpose: to verify if the silicon is functional (alive)
- Functional test is executed with relaxed conditions
- Test conditions are not defined in device specifications
- Provides valuable yield information

Equation Based Timing

Developing the functional timing so that the timing values can be easily modified is very important in test program development and debug. The test timing may be developed as an equation using program variables. By simply modifying the value of the variable ***SCALE.,*** in this example, the entire program timing can be easily “tightened” or “relaxed”. This technique can also be used to modify the timing between various tests, for example slow timing used within the gross functional test and fast timings used to verify the maximum operating speed of the device. When ***SCALE*** is set to 1, this example represents the timing for the 256 X 4 Static Ram as defined in the device AC Timing Specifications.

SCALE = 1; /* Note: ***SCALE*** is a programmable variable that controls timing values */

WriteCycle = 15E-9 * ***SCALE***; /* 66MHz */
Address_edge1 = 0E-9; **Add_edge2** = 15E-9 * ***SCALE***;
Data_In_edge1 = 2E-9 * ***SCALE***; **Data_In_edge2** = 14E-9 * ***SCALE***;
CS1_edge1 = 2E-9 * ***SCALE***; **CS1_edge2** = 13E-9 * ***SCALE***;
CS2_edge1 = 2E-9 * ***SCALE***; **CS2_edge2** = 13E-9 * ***SCALE***;
WE_edge1 = 2E-9 * ***SCALE***; **WE_edge2** = 13E-9 * ***SCALE***;
OE_edge1 = 2E-9 * ***SCALE***; **OE_edge2** = 11E-9 * ***SCALE***;
OUTPUT_VALID = **WriteCycle** * 0.75; **STROBEWINDOW1** = 2E-9;

Scalable timing- Key Points:

- Purpose: to ease the process of modifying related timings
- Can be developed as a relational timing diagram as defined in the device timing specification
- Can be used to generate timings for multiple speed devices

Functionally Testing a Device

The intent of the following discussion is to provide an overview of the relationship between the device specification, the test system hardware and the test program for a basic functional test. A simple device, a Clocked Inverter, has been created to help explain this concept. This device features two inputs—one data input and one clock input; it also has one inverting output. The device functions as follows:

1. The clock is the control signal that synchronizes the movement of data through the device, from input to output.
2. Input data is latched (read) into the device on the rising (positive) edge of the clock.
3. Output data is gated out of the device on the falling (negative) edge of the clock.
4. Data is transferred only when the clock is active.
5. Output data is the logical complement of the input (output is inverted vs. input)

Sample Device Specification

The data below defines the levels and timing needed to control the Clocked Inverter. Specifications generally represent the worst case conditions that a device must meet. It is up to the test engineer to develop a plan for implementing these conditions on the test system. The Clocked Inverter diagram illustrates one possible way this specification may be implemented on a test system.

VDD = 5.0V

VIH = 2.0V VIL = 0.8V

VOH = 2.4V VOL = 0.4V

Operating frequency = 10MHz with a Clock duty cycle of 50%

Data In Setup time = 15nsec (minimum time required for input data before positive clock edge)

Data In Hold time = 5nsec (minimum time required for input data after positive clock edge)

Output propagation delay = 8nsec (maximum time from negative clock edge to data out)

Note: See Chapter 8 for additional information regarding Setup time, Hold time and Propagation delay measurements.

Clocked Inverter

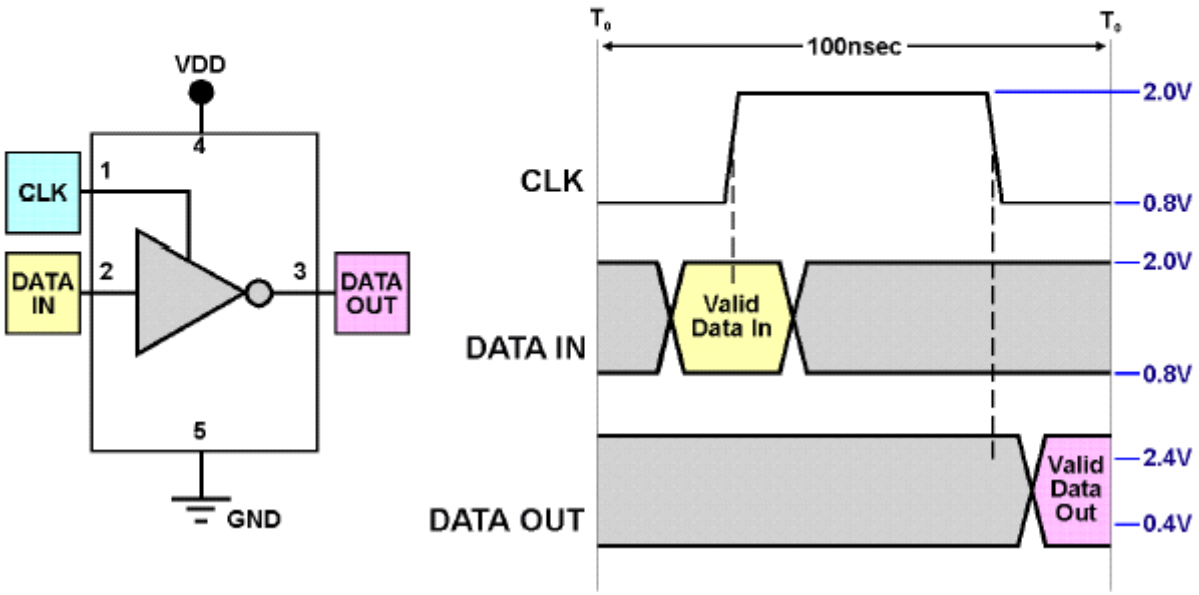


Figure 7-18

The Items Needed to Test this Device

- Test hardware and fixtures—loadboard, test socket, by-pass capacitors and wiring
- Device power, VDD and Ground
- Input levels, VIL (logic 0) and VIH (logic 1)
- Output reference levels, VOL (logic 0) and VOH (logic 1)
- Signal timing and format conditions for inputs and strobe timings for outputs
- A test vector pattern to verify logical functions

Programming Tester Resources

In order to guarantee that the device meets it's specification the test hardware must to programmed in a manner that will verify each parameter. First, one of the test system DPS units (device power supplies) will be used to provide VDD. VDD will be set to 5.0V as defined in the device specification. Ground will be supplied by the ground plane of the test hardware.

The RVS supplies will provide the pin electronics driver circuitry with the correct input levels. VIL (voltage in low) will be set to 0.8V and VIH (voltage in high) will be set to 2.0V). VIL is the voltage that will be applied to the input pins when a logic zero appears in the test pattern. VIH is the voltage that will be applied to the input pins when a logic one appears in the test pattern.

The RVS units also supply the pin electronics comparator circuits with the proper output reference levels. The comparators sense the voltage on the output pins of the DUT and compare the sensed voltage to the VOL and VOH reference voltages. The outputs will be functionally tested with the comparator reference levels set to 0.4 volts for VOL (voltage output low) and 2.4 volts for VOH (voltage output high). When an output logic zero appears in the test pattern the sensed output voltage will be compared to the VOL reference voltage, the test will fail if the sensed output voltage is greater than the VOL reference. When an output logic one appears in the test pattern the sensed output voltage will be compared to the VOH reference voltage. The test will fail if the sensed output voltage is less than the VOH reference.

The test system timing resources are used to define the test cycle time, the signal formats and edge placements for the input signals and the time at which the output signals will be sampled. The first step in developing the timing is to determine the test cycle time. The device specification defines the test frequency as 10MHz, this equates to a test cycle time of 100nsec by using the formula cycle_time = 1/frequency. The specification also states that the clock has a duty cycle of 50%, this indicates that the clock is low for 50nsec and high for 50nsec. For this example the clock will be low for the first 25nsec of the cycle, high for 50nsec and low for the last 25nsec. In order to produce the correct signal format for the Clock the RZ (return to zero) format will be used.

The DATAIN input signal timing is referenced to the Clock signal. The setup time specification states that the DATAIN signal must be valid 15nsec before the positive clock transition and must remain valid for the specified hold time of 5nsec after the positive transition, therefore the total pulse width of DATAIN will be 20nsec. In order to correctly verify the Setup and Hold time parameters the signal format (waveshape) must be SBC (surround by complement). Setup and Hold time parameter testing will be explained in detail in a later chapter.

The final step in developing the test timing is to establish the timing for the output pin. The device specification states that the output propagation delay is 8nsec and is referenced to the falling edge of the Clock. The falling edge of the Clock will occur at 75nsec so the output will need to be tested at 75nsec+8nsec into the cycle. Outputs are verified by using a section
of the test hardware known as the output strobe, so for this example the output strobe will be programmed to occur at 83nsec. If the output is in the correct logic state (as predicted by the truth table) at the time the output strobe occurs and if the output meets the correct voltage level as set by VOL and VOH the functional test will result in a pass conditions. The output propagation delay parameter will be explained in detail in a later chapter.

A test pattern will be needed to verify correct functionally of the DUT. The test pattern can be hand written since the function of the sample device is simple. The test pattern will be stored in the test vector memory and executed each time the functional test is active.

The Test Vector Pattern

A test vector pattern, also called a truth table, must be created. The test vectors will be used during the functional test to verify that the DUT is capable of performing its logical functions correctly. Test vector patterns typically consist of a set of characters which represent logic states to be applied to the input pins and the expected response of the output pins. The test vector pattern for this sample device consists of seven vectors, each vector representing the data for one cycle. The test vector data is combined with the timing, format and level information to create the complete test as shown in Figure 7–19. The test vectors for the clocked inverter are shown in Table 6–1. It shows seven test vectors for the clocked inverter—each vector represents the data for one cycle.

The characters used in the vector pattern represent:

1 Drive input high (to logic 1)
0 Drive input low (to logic 0)
H Compare output to a high
L Compare output to a low

Table 6-1 Test Vectors for Clocked Inverter

DATAIN	CLOCK	DATAOUT	REMARKS
1	1	L	Input 1 / Clock / Output 0
0	1	H	Input 0 / Clock / Output 0 to1 transition
1	1	L	Input 1 / Clock / Output 1 to0 transition
0	0	L	Input 0 / No Clock / Output remains 0
0	1	H	Input 0 / Clock / Output 1
1	0	H	Input 1 / No Clock / Output remains 1
1	1	L	Input 1 / Clock / Output 0

Specification Test Conditions for the Clocked Inverter

Figure 7–19 shows the activity that will occur during the execution of the functional test. The timing diagram shows seven cycles of test vector data combined with the signal timings, signal formats and voltage levels. The information shown in this diagram reflects the conditions as defined in the device specification. The functional test rate is set to 100nsec, the CLOCK has a 50% duty cycle and is in RZ signal format. DATA IN has the correct setup and hold timings and is in SBC signal format (Surround By Compliment). The input and output levels are also set to the values defined in the device specification.

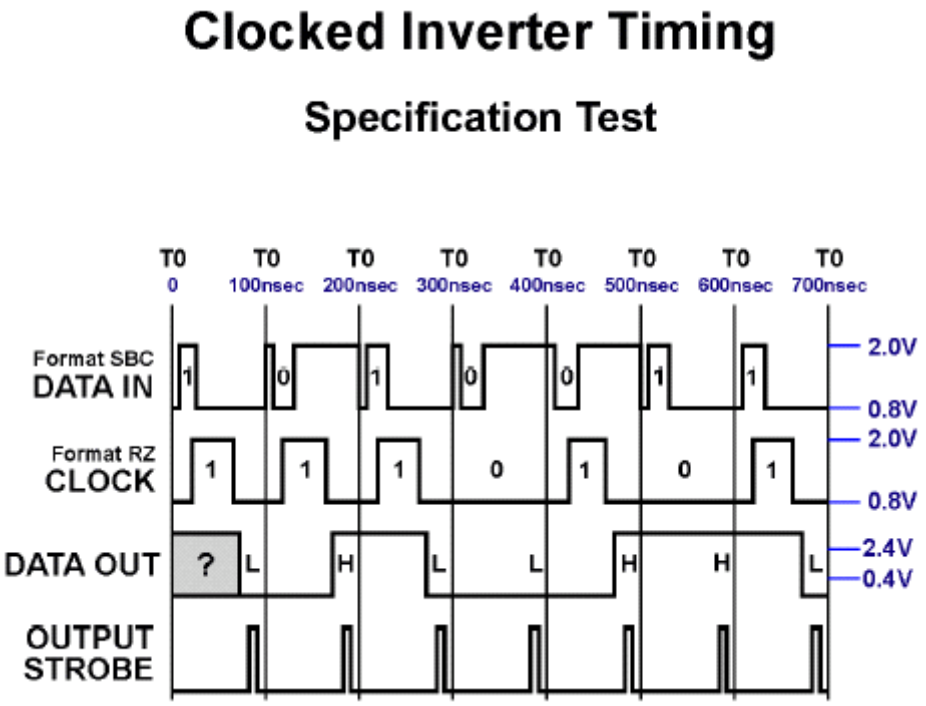


Figure 7-19

Gross Functional Test Conditions for the Clocked Inverter

It is often useful to perform a gross functional test. The purpose of the gross functional test is to determine if the device is functionally “alive” without regards to the exact values defined in the device specification. If the device is tested to its specifications and the result of the test is a failure, the failure may be caused by a sensitivity to one or more parameter values such as input levels, output levels or timings. It may also fail because of physical defects in the silicon or due to an error during the fabrication processing.

The waveform display shown in Figure 7–20 represents the timing and voltage conditions that would be used in a gross functional test. The input and output voltage levels have been modified (relaxed) in a manner that will make it easier for the device to function. The test rate has been relaxed (slowed down) from 100nsec to 500nsec. The DATA IN signal now changes only at the start of each cycle (NRZ format, no setup or hold times). The output propagation delay timing has also been increased proportionately giving the output more time to change state and stabilize.

If the device is tested to the device specifications as in Figure 7–19 and fails, then retested and passes with relaxed conditions as in Figure 7–20, the failure was not caused by dead (totally non-functional) silicon. Each parameter can then be individually tested to specification to determine the actual sensitivity that caused of the failure. Gross functional test conditions are often used during the course of developing the test vector patterns.

Note: When a parameter value is relaxed its value is adjusted in a way that will make it easier for the DUT to function properly. For example, if VIL is specified as 0.8V it could be relaxed by setting the value to 0.0V. By lowering the value of VIL it becomes easier for the DUT to detect the input voltage as a logic 0. To relax inputs, lower the VIL levels and raise the VIH levels. To relax outputs, set both VOL and VOH to 1.5V, the comparators will then sense any output voltage below 1.5V as a logic 0 and any voltage above 1.5V as a logic 1. To relax timing values, reduce the test rate, increase setup and hold times and increase output propagation delay limits.

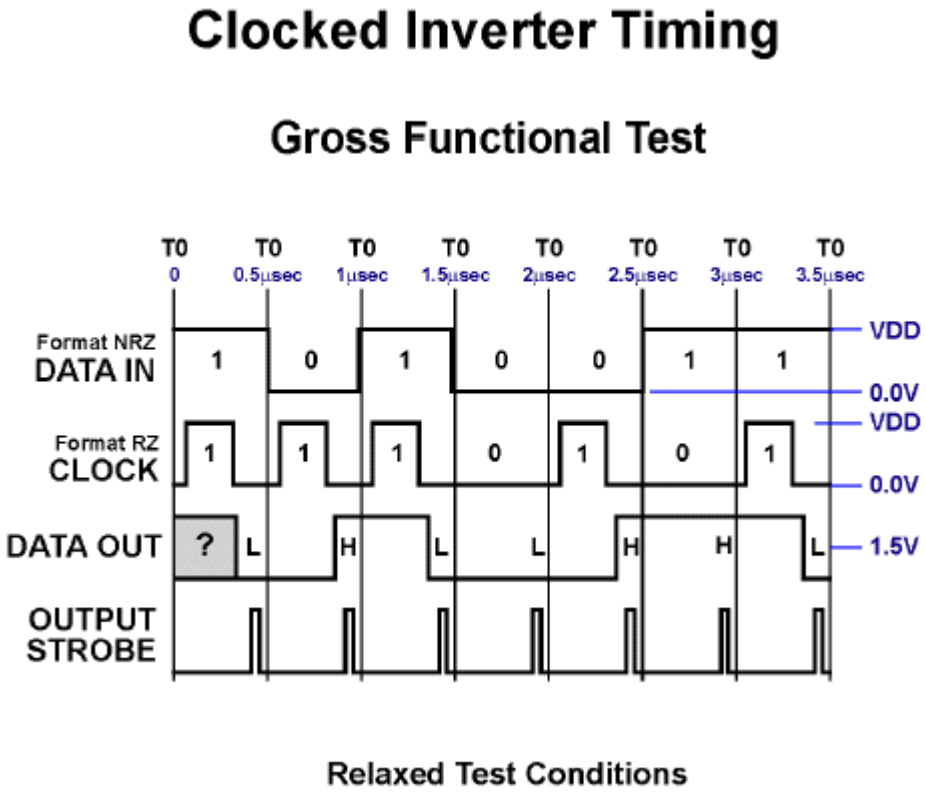


Figure 7-20

Test Program Statements for Clocked Inverter

Using a non-tester specific pseudo code a functional test program can be developed for the clocked inverter:

```
Begin Program;

/* define pin functions */
Connect DPS1 pin 4;
Connect GND pin 5; /* ground is hard wired */
Data_in input pin 1;
Clock input pin 2;
Data_out output pin 3;

/* set voltage levels */
Set DPS1 5V;
Set VIL 0.8V; Set VIH 2.0V;
Set VOL 0.4V; Set VOH 2.4V;

/* set timings */
Set Test Period 100nsec;
Set Clock start_edge 25nsec; Set Clock stop_edge 75nsec;
Set Clock format RZ;
Set Data_in start_edge 10nsec; Set Data_in stop_edge 30nsec;
Set Data_in format SBC;
Set Data_out start_strobe 83nsec; Set Data_out stop_strobe 93nsec;

/* load the test vector file into vector memory */
At Vector Location 0;
Load Pattern (test_vectors);
Start Pattern 0;
Stop Pattern 6;

Execute Test_pattern; /* this runs the test */

/* Turn off voltage levels */
Set VIL 0.0V; Set VIH 0.0V;
Set VOL 0.0V; Set VOH 0.0V;
Set DPS1 0.0V;

End Program;
```

Standard Functional Tests

Although each unique circuit design requires a unique set of functional test conditions, there are a few parameters that may be verified functionally. Let's look at a few of these parameters, keeping in mind that what is being explained is a standard methodology to functionally verifying the given parameter.

Opens and Shorts - Functional Method

A faster and less costly way to test for opens and shorts is to do it as a functional test rather than as a DC test. First, the functional test timing must be defined. For this example a 1 sec test period is used. Each pin is functionally tested, so output strobe timing must be set. The strobe window is set to occur at 900nsec and to have a 10nsec width. See Figure 7–21 for a functional timing diagram.

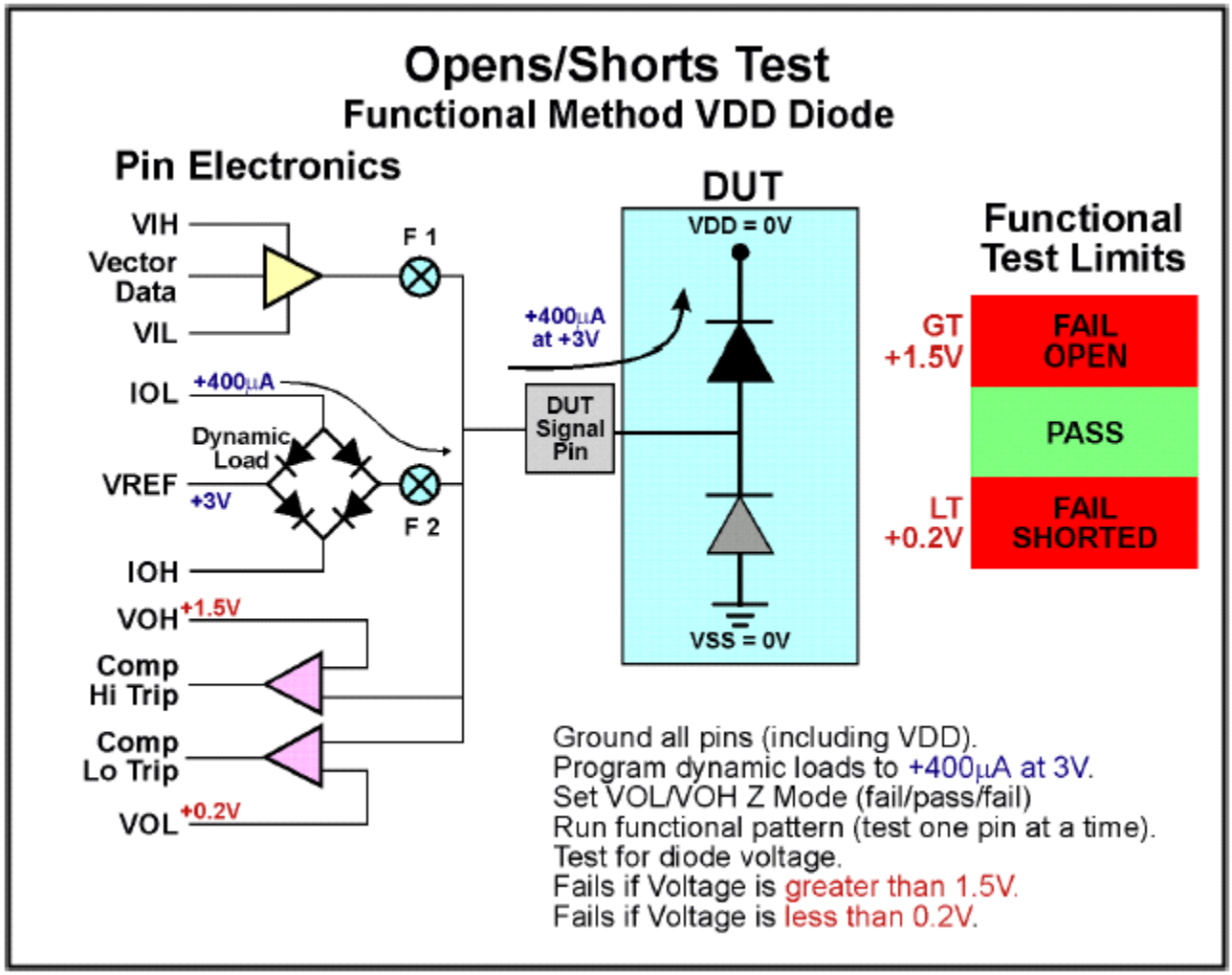


Figure 7-21

All signal pins must be tied to ground. This is done by defining all signal pins as inputs and applying VIL (set to zero volts) through the pin electronics. All power pins, VDD and VSS, must be connected to ground (zero volts). The dynamic current loads supply the current and voltage needed to forward bias the VDD protection diode. The programmable loads supply 400 A of current. The load reference voltage (VREF) is set to +3V. The output comparator levels must be

programmed so that a center pass region is defined (often called “setting the comparators to mid–band or Z state mode”). The VOL level is set to +0.2V and the VOH level is set to +1.5V. See Figure 7–21.

A functional test pattern must be developed which will execute the following sequence:

Cycle	Action
1	Define all signal pins as inputs and force VIL (zero volts). The character “0” in the vector file instructs the tester to perform this function on each pin.
2	Define the first signal pin as an output to be tested, turn off the tester drive on that pin, compare output for pass/fail. The “Z” character instructs the tester to perform this function on the first pin to be tested.
3	Turn the driver back on for the pin tested in the last cycle and repeat step two for the next pin to be tested.
4	Repeat steps 2 and 3 until all signal pins are tested through cycle 6.

Sample Test Vector File for Opens/Shorts

```
00000      /* cycle 1 ground all pins */
Z0000      /* cycle 2 test for diode on first pin */
0Z000      /* cycle 3 test for diode on second pin */
00Z00      /* cycle 4 test for diode on third pin */
000Z0      /* cycle 5 test for diode on fourth pin */
0000Z      /* cycle 6 test for diode on fifth pin */
           /* the next cycle is executed separately */
ZZZZZ      /* cycle 7 turns the drivers off and tests all pins */
```

When the walking Z test pattern is executed, the first signal pin is tested in cycle 2. Once the tester pin driver is shut off the dynamic current load begins to pull the device pin toward +3V, which is supplied by VREF. If a working diode exists it will turn on when the voltage reaches +0.65V. The diode will then clamp the VREF voltage at that point, while sinking the +400mA supplied by the IOL side of the programmable current load.

When the pass/fail comparison is made, the test will pass because the tester comparators will sense the +0.65V which is within the upper VOH limit of +1.5V and also within the lower VOL limit of +0.2V. If a short occurs, the comparators will sense 0V; if an open occurs the comparators will sense +3.0V. Either case will result in a failure.

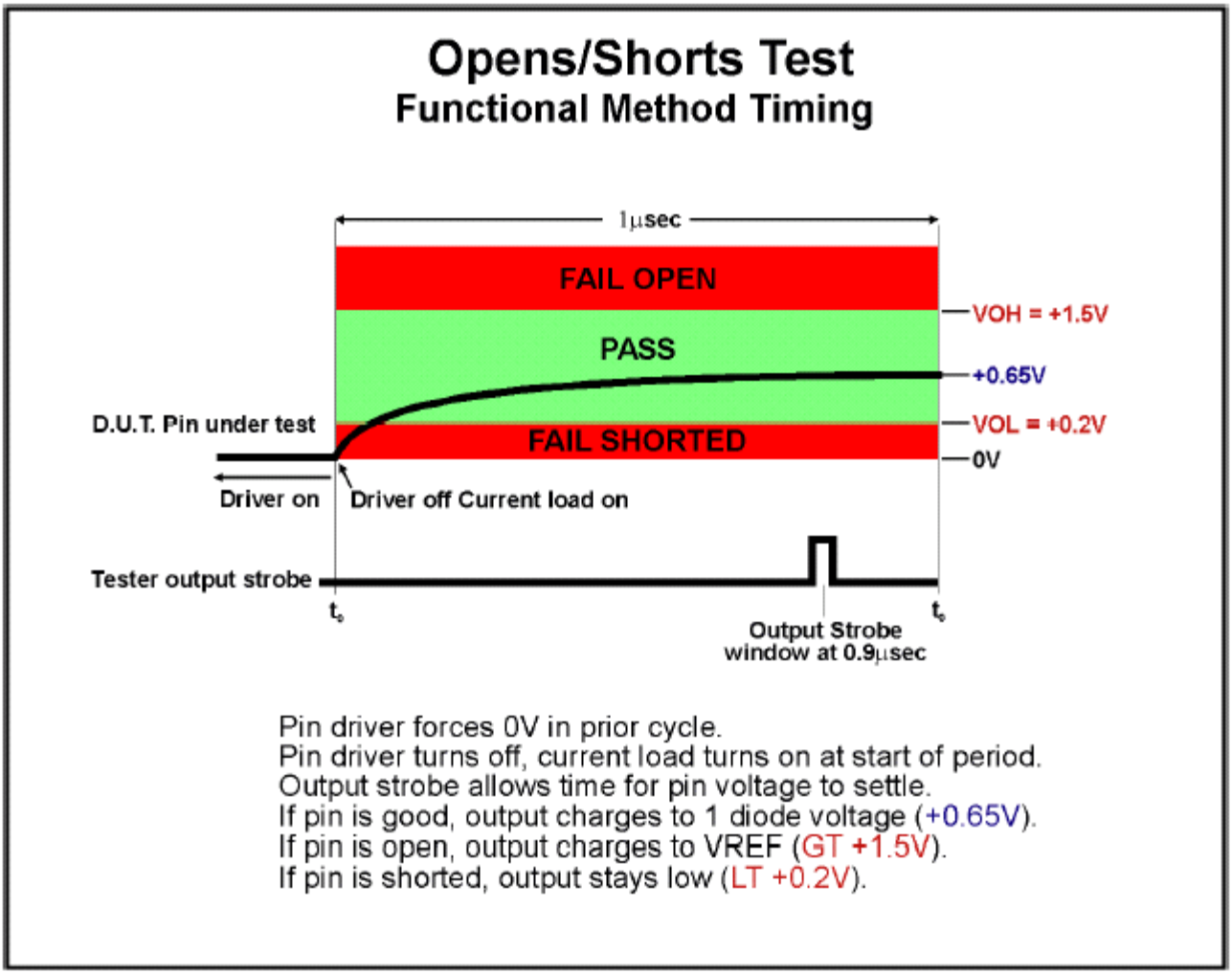


Figure 7-22

The timing diagram indicates a test period of 1microsecond (1MHz). At the start of the cycle the driver turns off for the pin under test and the current load turns on. The current load will pull the pin under test towards the VREF voltage. If there is a working diode with no shorts the comparators will sense a valid diode drop and the test will pass. Notice the output strobe is placed at 900 nanoseconds into the cycle, this allows sufficient time for the current load to stabilize and the diode to turn on, insuring a solid test result.

Note: the purpose of the waking Z pattern, while testing the VDD diodes, insures that no pin-to-pin shorts exist. If the pin under test is shorted to another device pin, it will show up as a failure due to the fact that the pin it is shorted to is tied to ground.

Once all of the VDD diodes are tested for both opens and shorts, the lower VSS diodes will need to be tested. If all of the VDD diodes pass the test, there are no shorts on any device pins, otherwise the test would fail. The lower VSS diodes only need to be tested to insure that they are not open. This can be accomplish by setting the conditions shown in the diagram below and then executing only test cycle 7, containing ZZZZ. All VSS diodes will be tested in parallel, in one single cycle, this will verify that no diodes are open.

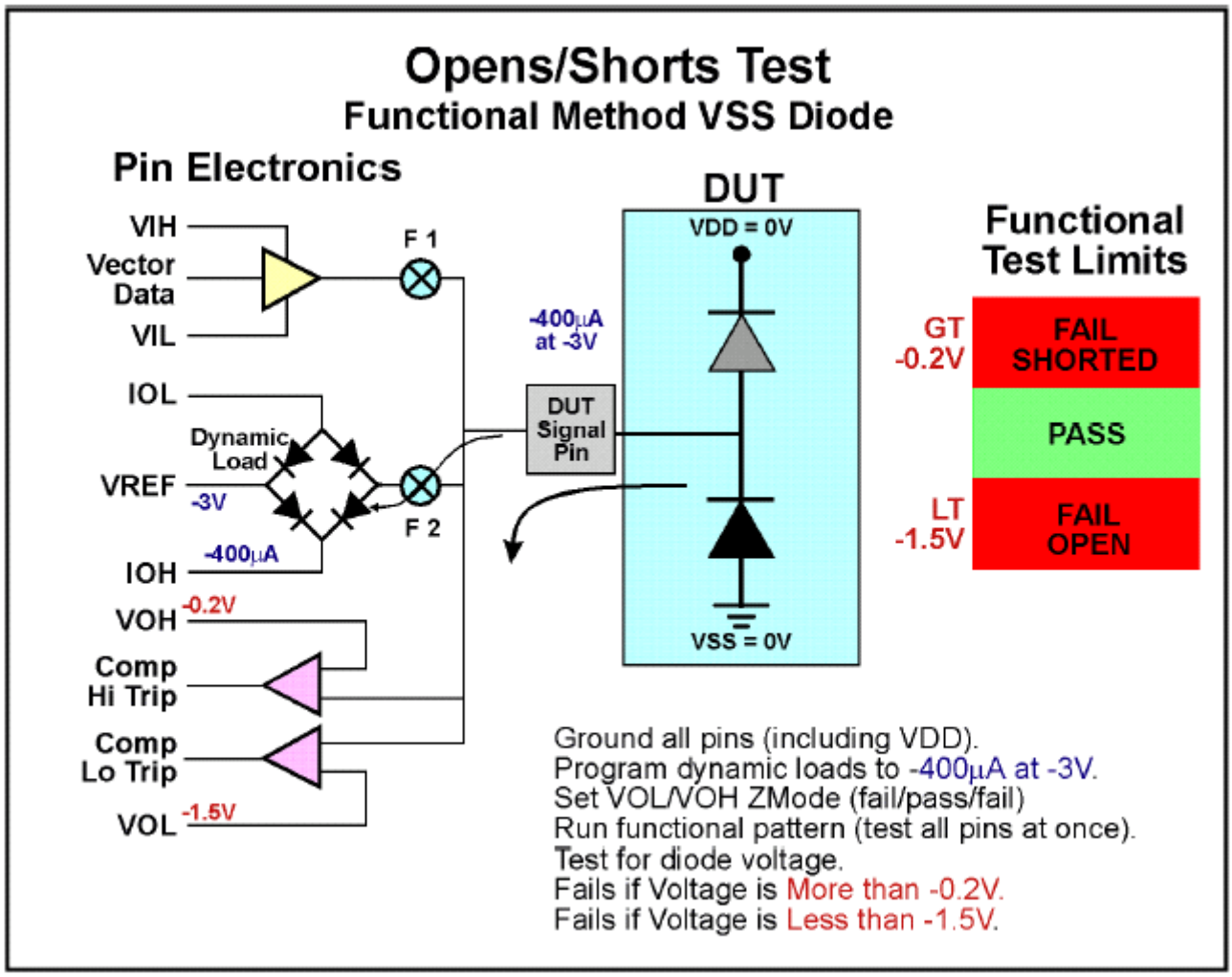


Figure 7-23

The advantage of the Functional Opens/Shorts test is speed. The test will execute very fast compared to the DC serial/static method. The disadvantage is that the datalog results are somewhat harder to understand. Often, test operators are accustomed to seeing the DC readings produced by the serial/static method. By executing the serial/static test only when the Functional Opens/Shorts test fails, any DC readings on failed pins can be datalogged for operators without adding a lot of additional test time.

Soft Test ST

Functional Shorts Datalog

```
Test 1:    4:Stage 9, SHORTS:Shorts

S:Pattern                          Start-Loc  Stop-Loc        Size  Mode

contact                                3        103           101  norm

Result *FAIL*
-------------------------------------------------------------------------

S          CRCEIITTTTIIIISDDIPPPPIPPPIIIIREIEEDOIOIDSSIODDTDDDDDDDDDDD
i          KSKXNNCMDDOOOOAOICISSSAOIOOOOOWRORRSCCLLOAYBBIORBBBBBBBBBBB
t          ITOMTTKSIOBBBBDE2KBEEECBDDBBBBNA AOEKKDDEDNFE1UA11111100000
e           B  10     IIIIDN_2FLLLKESSIIII M MML1111NDC11 TP54321098765
#                     TTTT22P__210 ___TTTT L H      111   1
                      3210__BPI___ OIO4567 O I
                      ___PP0BBODS BLL____
                      PPPPBB10FCOY EDDVVVV
1:       2 +00000000000000000000000000000000000000000000000000000000000
1:       6 0000+00000000000000000000000000000000000000000000000000000000
1:      31 00000000000000000000000000000+00000000000000000000000000000
1:      37 000000000000000000000000000000000+0000000000000000000000000
1:      44 00000000000000000000000000000000000+000000000000000000000
1:      55 0000000000000000000000000000000000000000000000000000000+00000
1:      56 00000000000000000000000000000000000000000000000000000000+0000
1:      57 000000000000000000000000000000000000000000000000000000000+00
```

The datalog above shows failures that occurred during the Functional Shorts test. The “+” character indicates the failing pins.

Key Points

- Purpose: to detect open or shorted device pins and verify proper connections between the test system and the DUT.
- Functional Test serial/dynamic test method
- Uses programmable current loads and functional comparators
- Test requirements not found in device specifications
- Much faster than the DC serial/static method

VIL/VIH

VIL (Voltage, Input Low) represents the worst case voltage applied to an input to represent a logic 0. VIH (Voltage, Input High) represents the worst case voltage applied to an input to represent a logic 1. The table below shows the VIL/VIH specifications for the 256 x 4 Static RAM.

Parameter	Description	Test Conditions	Min	Max	Units
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V

Why Test for VIL/VIH?

The VIL/VIH test guarantees that the input pins can correctly sense the proper logic states when the VIL/VIH voltages are applied. VIL represents the maximum voltage that the DUT is guaranteed to sense as a logic 0. VIH represents the minimum voltage that the DUT is guaranteed to sense as a logic 1.

VIL/VIH Functional Test Method

Although the VIL and VIH levels are often defined in the device specification under the heading of “DC Characteristics”, they must be verified by performing a functional test. The test is performed by applying the input levels defined in the device specification and then executing a functional test pattern. If the test results in a pass, the device has operated correctly and meets the VIL/VIH specifications. If the test results in a fail, the device has not met the intended specification.

For TTL logic, the VIL level is often defined as 0.8V and VIH as 2.0V. For pure CMOS logic, input levels may be defined as a percentage of VDD, where, for example, VIL is (0.3 * VDD) and VIH is (0.7 * VDD).

Notes: 1) VIL/VIH requires 2 test iterations, one at VDDMIN and one at VDDMAX. 2) Some CMOS device types and families have TTL compatible levels.

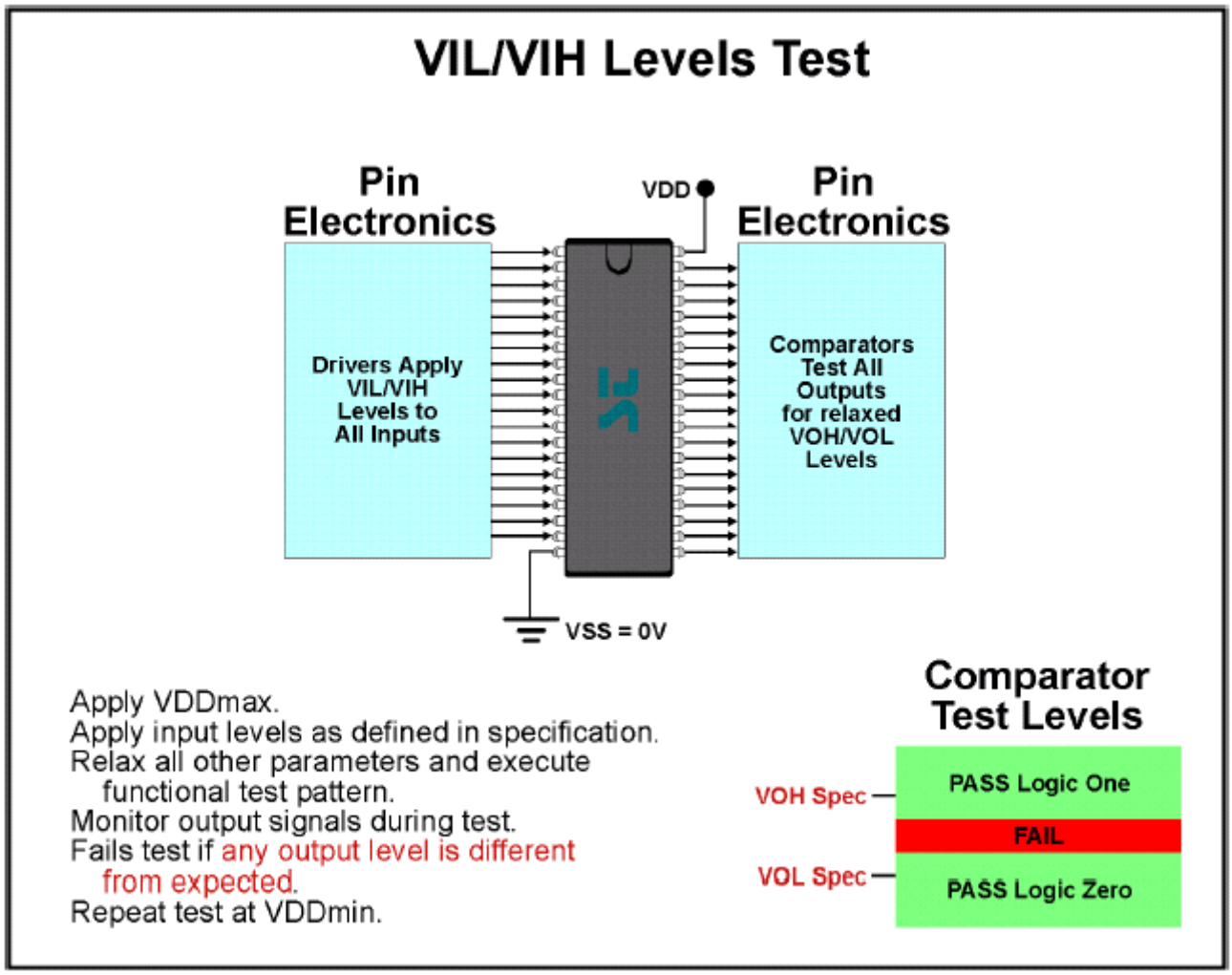


Figure 7-24

VIL/VIH—Trouble Shooting:

To begin trouble shooting, enable the datalogger and observe the test results. If a DUT standard (a known good device) is available, test it and observe the result.

When the VIL/VIH test fails, the failure appears as one or more incorrect output signals. When testing a complex device it is often difficult or impossible to determine which input is causing the failure unless each input is tested individually.

Start by relaxing both the VIL and VIH levels. Relax the VIL level by lowering it—zero volts is the most relaxed VIL voltage level. Relax the VIH level by raising it—VDD is the most relaxed VIH voltage level. After relaxing the input levels rerun the test. It should pass with the relaxed conditions since it has already passed the gross functional test.

Next set VIH back to the original specification voltage value and rerun the test. If the result is a pass then VIH meets the specification and the VIL parameter must be the cause of the failure. If the test fails, relax the VIH value on all but one pin and rerun the test. This will verify one pin at a time for VIH; this process can be repeated to verify VIL. Output current loading and high frequency testing can cause noise within the DUT which can affect the VIL/VIH test. It may be necessary to eliminate all output loading and to reduce the test frequency (increase the test period).

Read the device specification carefully—it may state that the VIL and VIH levels are valid only in a “static noise free environment.” This is an indication that the VIL and VIH levels may need to be relaxed somewhat when used in production testing.

VIL/VIH - Key Points:

- Purpose: to verify that the input buffers will properly detect VIL and VIH voltage levels
- VIL/VIH can only be verified by executing a dynamic functional test
- Test limits are defined in device specifications (often as DC)
- Output pins fail as a result of improper operation of input circuitry

VOL/IOL VOH/IOH Functional Test

VOL (Voltage, Output Low) represents the maximum voltage produced by an output when the output is in the low state. IOL (current(I), Output Low) represents the current sinking capabilities of an output when the output is in the low state. VOH (Voltage, Output High) represents the minimum voltage produced by an output when the output is in the high state. IOH (current(I), Output High) represents the current sourcing capabilities of an output when the output is in the high state. The table below shows the VOL/VOH specifications for the 256 x 4 Static RAM.

Parameter	Description	Test Conditions	Min	Max	Units
VOH	Output HIGH Voltage	VDD=4.75V, IOH = -5.2mA	2.4		V
VOL	Output LOW Voltage	VDD=4.75V, IOL = 8.0mA		0.4	V

Why Test Functionally for VOL/IOL VOH/IOH?

The VOL/IOL/VOH/IOH test verifies the resistance of output pins while driving valid output levels under current load. This test insures that the outputs will provide the specified IOL/IOH current while maintaining the correct VOL/VOH voltages. Performing the VOL/IOL/VOH/IOH test functionally has a significant speed advantage when compared to performing the same test using the serial/static PMU method.

VOL/IOL VOH/IOH—Dynamic Test Method

These parameters may be verified either statically or dynamically. To perform the test dynamically, the tester comparator levels are set to the specified VOL/VOH values, load currents are applied (dynamic load or resistive load) and the functional test is executed.

During functional test execution, the outputs must sink and source the proper IOL/IOH currents and the functional output comparator circuitry insures that the outputs maintain the proper VOL/VOH voltage. If an output is weak and cannot sink or source the proper current, the comparators will detect a voltage which is outside of the VOL/VOH limits and the test will fail.

Note: Loading the outputs during the execution of a functional test can result in high current flows within the DUT. If the DUT has a high number of outputs or it has high current output buffers, it may not be possible to load all of the outputs at the same time and still meet the VOL/VOH voltage specification. High currents within the DUT may generate noise which can appear on the outputs and cause the comparators to fail the VOL/VOH levels. If this problem occurs apply loads to a small number of pins and execute the functional test, repeat until all outputs have been properly loaded and tested. In some cases it may be necessary to decrease the test rate and move the output strobe time further into the cycle to achieve stable test results.

Be sure to check the specification for the exact voltage and current values to be used in the test program. VOL is often defined as 0.4V, VOH is often defined as 2.4V. Output levels may also be defined relative to power supply levels, e.g. VOL = (Gnd + 0.1 or VSS + 0.1) and VOH = (VDD - 0.1).

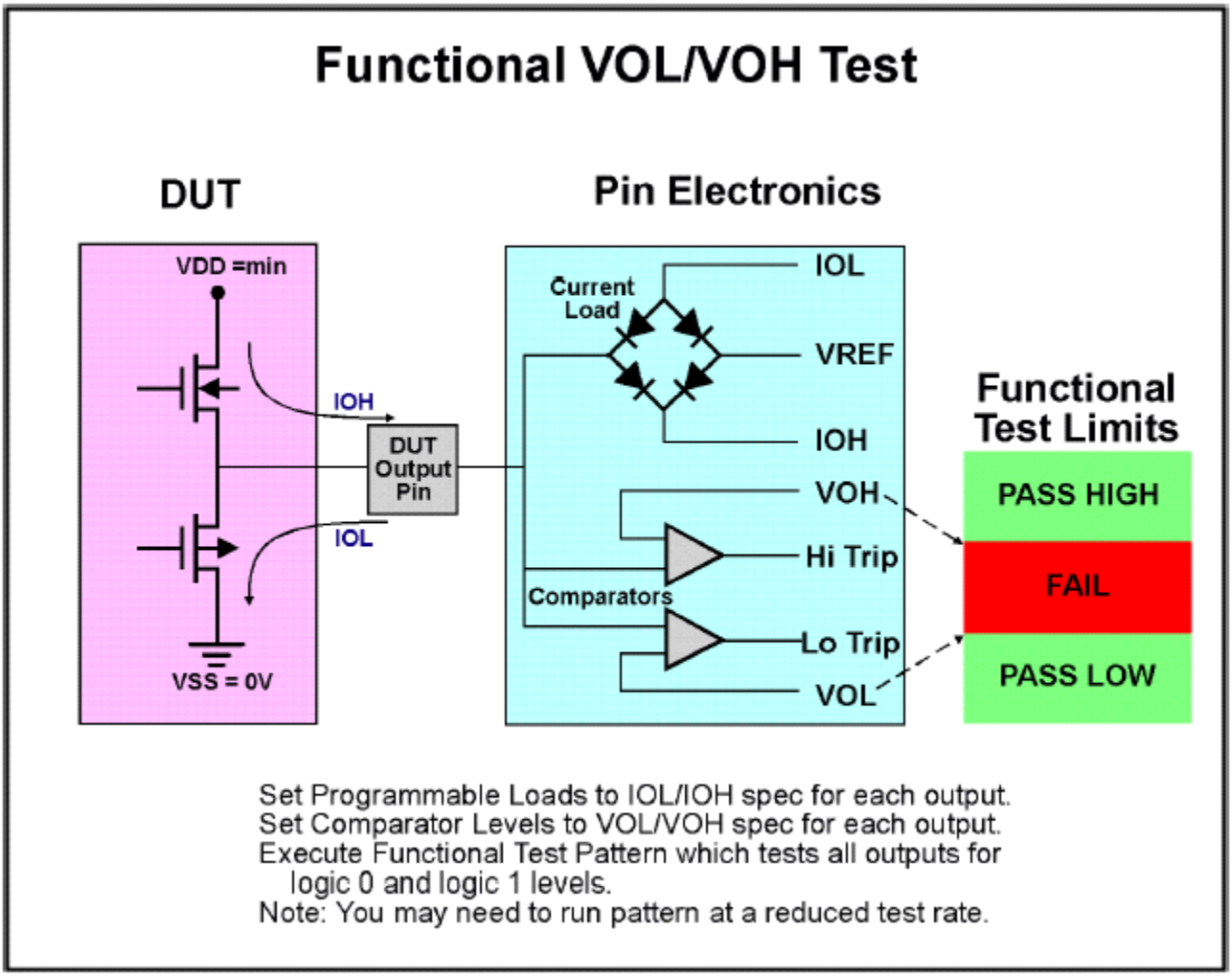


Figure 7-25

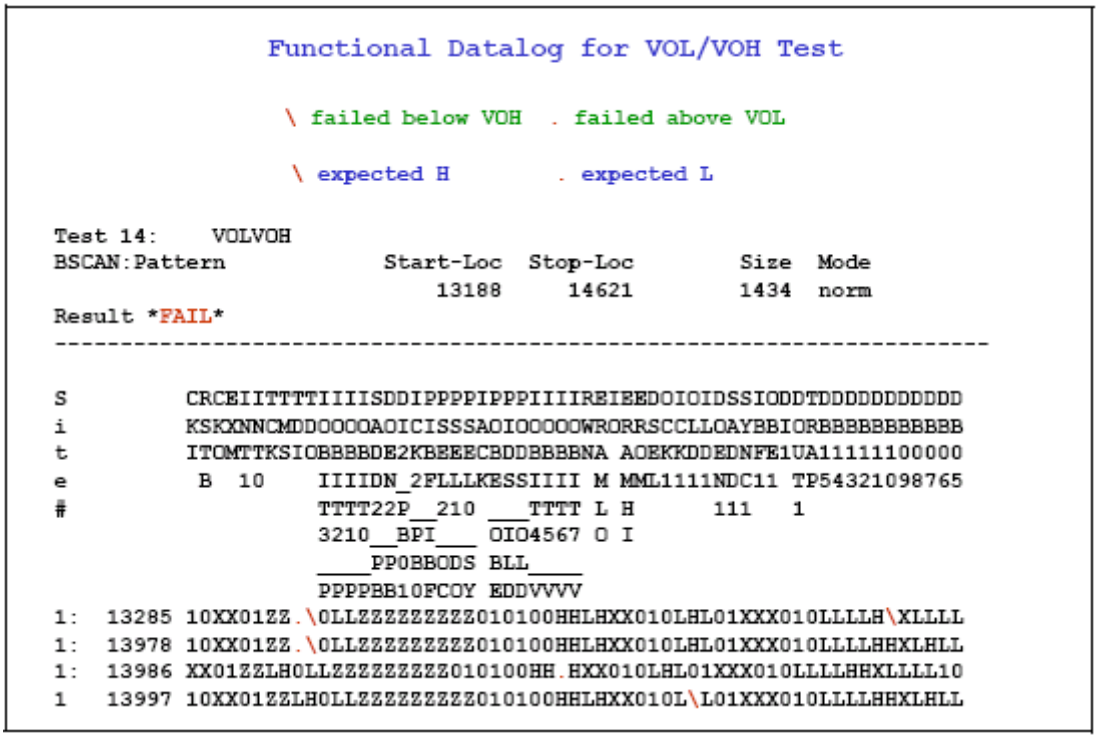
VOL/IOL VOH/IOH - Trouble Shooting:

To begin trouble shooting, enable the datalogger and observe the test results. If a DUT standard is available, test it and observe the result.

When the VOL/IOL VOH/IOH test fails, the failure appears as one or more incorrect output signals. The datalogger indicates which device pins failed and the failing states. Observe the failing results and see if failures are both ones and zeros. If so, both VOL and VOH levels are defective; if not, then only one parameter is causing the failure. Verify that the test can be made to pass by relaxing the VOL/VOH levels. The IOL/IOH currents can also be relaxed to try to make the test pass.

Note: The outputs will require more time to transition between logic levels when fully loaded with current, so it may be necessary to reduce the test frequency and move the output strobe timing to a later time in the test cycle to get the test to pass.

Functional VOL/VOH Datalog



```
                Functional Datalog for VOL/VOH Test

                 \ failed below VOH   . failed above VOL

                  \ expected H         . expected L

Test 14:    VOLVOH
BSCAN:Pattern            Start-Loc  Stop-Loc        Size  Mode
                             13188     14621        1434  norm
Result *FAIL*
-------------------------------------------------------------------------

S         CRCEIITTTTIIIISDDIPPPPIPPPIIIIREIEEDOIOIDSSIODDTDDDDDDDDDDD
i         KSKXNNCMDDOOOOAOICISSSAOIOOOOOWRORRSCCLLOAYBBIORBBBBBBBBBBB
t         ITOMTTKSIOBBBBDE2KBEEECBDDBBBBNA AOEKKDDEDNFE1UA11111100000
e          B  10    IIIIDN_2FLLLKESSIIII M MML1111NDC11 TP54321098765
#                   TTTT22P__210 ___TTTT L H      111   1
                    3210__BPI___ OIO4567 O I
                    ____PP0BBODS BLL____
                    PPPPBB10FCOY EDDVVVV
1:   13285 10XX01ZZ.\0LLZZZZZZZZZ010100HHLHXX010LHL01XXX010LLLLH\XLLLL
1:   13978 10XX01ZZ.\0LLZZZZZZZZZ010100HHLHXX010LHL01XXX010LLLLHHXLHLL
1:   13986 XX01ZZLH0LLZZZZZZZZZ010100HH.HXX010LHL01XXX010LLLLHHXLLLL10
1    13997 10XX01ZZLH0LLZZZZZZZZZ010100HHLHXX010L\L01XXX010LLLLHHXLHLL
```

The datalog above shows failures that occurred during the Functional VOL/VOH test. The “.”character indicates an output failing a zero (L) state, the “\” character indicates an output failing a one (H) state. The functional datalog makes it very easy to see whether the device is failing a one level or a zero level, the failing signal is also indicated.

VOL/IOL VOH/IOH - Key Points:

- Purpose: to verify that the output buffers will properly supply the correct amount of current (IOL/IOH) at the proper voltage (VOL/VOH)
- Dynamic functional test must be executed
- Test limits defined in device specifications
- Test requires current loads on output pins
- It may not be possible to test all output pins simultaneously when fully loaded, due to noise produced by high currents

Resistive Output Loading

When utilizing a test system which does not offer programmable current loads, resistive loads can be added to the external test hardware to supply the IOL/IOH currents. Figure 7-26 is an example of a single resistive load. The device specification defines the values of VOL/IOL and VOH/IOH. The following formula can be used to determine the reference voltage (VREF) and a single resistor value required to satisfy the specification.

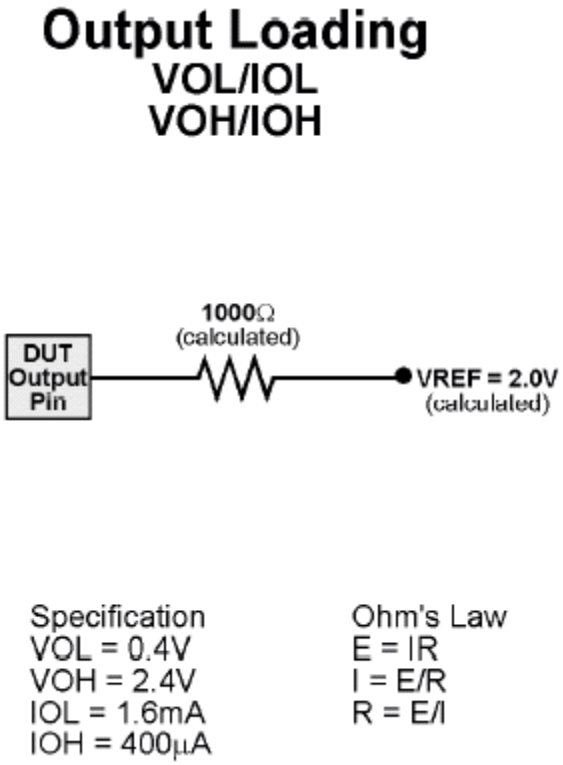


Figure 7-26

To calculate the reference voltage:

VREF = ((IOL * VOH) + (IOH * VOL)) / (IOL + IOH)

To calculate the resistance value:

Resistor = (VREF - VOL) / IOL

Input/Output Levels Relationship

Figure 7–27 shows the relationship between VIL/VIH levels and VOL/VOH levels. Typically, input levels are specified as 0.8V and 2.0V and output levels as 0.4V and 2.4V. In the use of semiconductors, the output of one device is most often connected to the input of another device.

When testing to the above specifications, it can be seen that a 400mV noise margin exists between input and output levels.
VOH is guaranteed to be 2.4V or greater, and the input level VIH is guaranteed to detect 2.0V or greater as a logic 1. The same applies to VOL and VIL—VOL is guaranteed by testing to provide a voltage which is 400mV lower than the specified value of VIL.

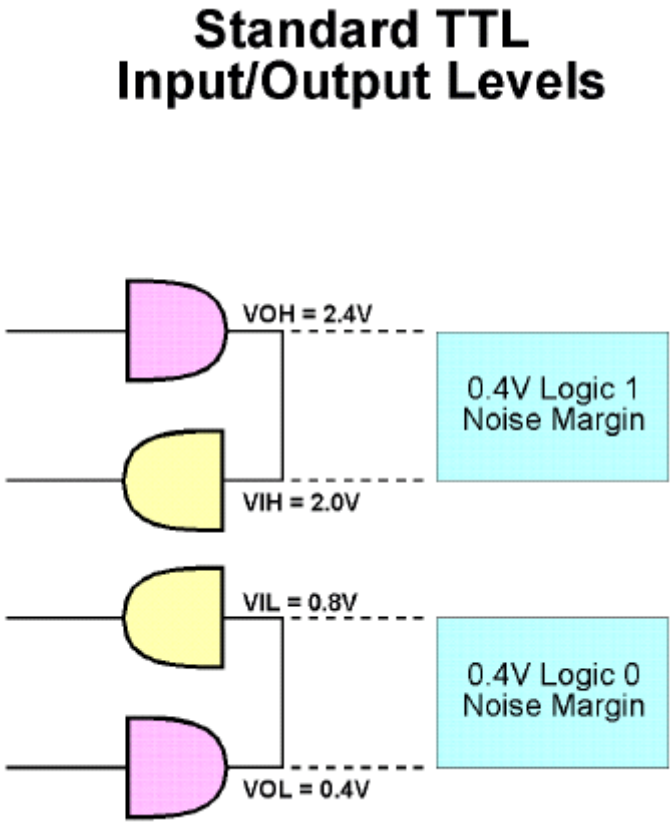


Figure 7-27

Functional Z–State — High Impedance Testing

When a device has bidirectional pins or three state outputs, there can be many logic conditions under which the output buffers must be in its high impedance state. These conditions depend on the functional test pattern sequences and outputs must be checked for the correct high impedance state under all of these conditions, thus the Z–State test.

Why Test Z–State Functionally?

The functional Z–State test verifies that the DUT outputs can achieve the proper high impedance state while the device is actively executing vector patterns.

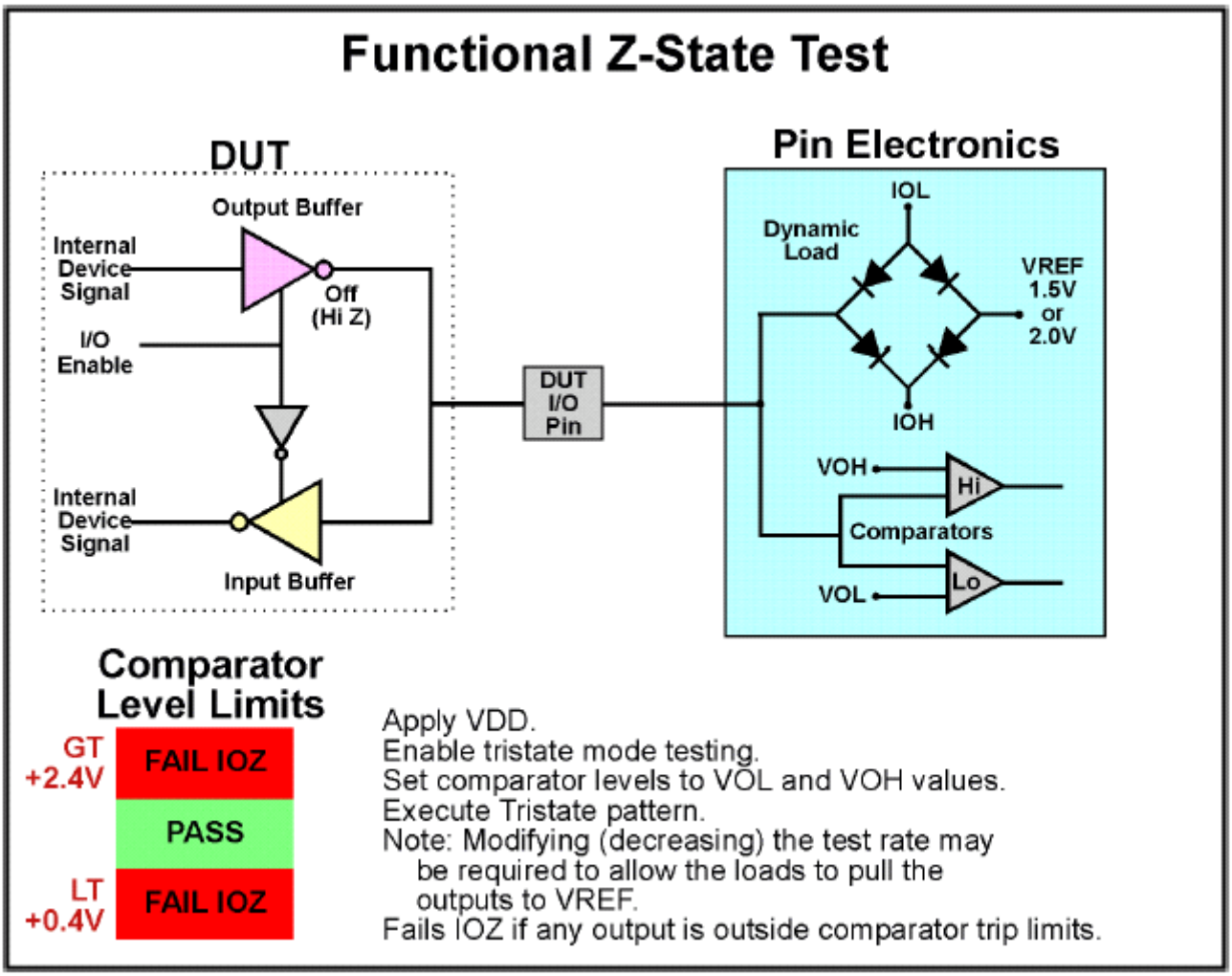


Figure 7-28

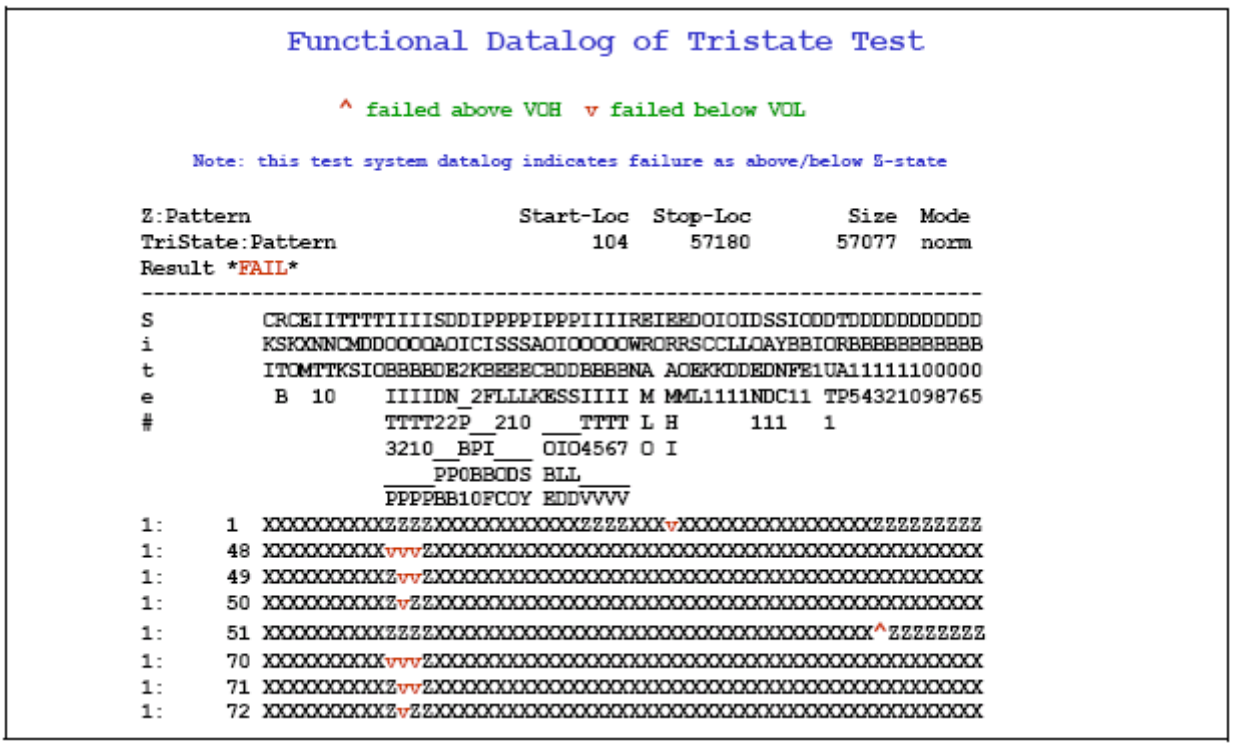
Functional Z–State—Test Method

The Z–State high impedance (tri–state) test is made to insure that bi–directional and high impedance outputs are capable of achieving a high impedance or off state. Some test systems have the ability to perform this test functionally. When performed as a functional test, tri–state output conditions are tested dynamically.

Functional tri–state testing requires the use of programmable loads or external resistive loads. The reference voltage of the load must be programmed to a value between the VOH and the VOL compare level. Often, 1.5V or 2.0V is used. As the output under test enters a high impedance state, it loses the ability to sink or source current. The test load then pulls the output to the intermediate value of 2.0V. The comparators are set to tri–state mode testing. This provides a pass zone, surrounded above by a fail zone (logic 1) and below by a fail zone (logic 0). Review Figure 7–28. for more on high impedance level testing.

The exact conditions for functional high impedance testing are not normally defined in the device specification, so some experimentation may be required to produce a reliable test. When the output of the DUT enters a high impedance state it becomes the responsibility of the test system to provide the intermediate voltage level. The amount of time required for the intermediate level to be reached depends on how much current the output load delivers and on the capacitive loading of the tester channel. This test can be sensitive to changes in test hardware—different results may be experienced between wafer test, hand test and auto handler test due to the different capacitive loads presented.

Functional Tri-state Datalog



Functional Datalog of Tristate Test

^ failed above VOH v failed below VOL

Note: this test system datalog indicates failure as above/below Z-state

```
Z:Pattern                        Start-Loc  Stop-Loc        Size  Mode
TriState:Pattern                       104     57180       57077  norm
Result *FAIL*
--------------------------------------------------------------------------
S         CRCEIITTTTIIIISDDIPPPPIPPPIIIIREIEEDOIOIDSSICODTDDDDDDDDDDD
i         KSKXNNCMDDOOOQAOICISSSAOIOOOOOWRORRSCCLLOAYBBIORBBBBBBBBBBB
t         ITOMTTKSIOBBBBDE2KBEEECBDDBBBBNA AOEKKDDEDNFE1UA11111100000
e          B  10    IIIIDN_2FLLLKESSIIII M MML1111NDC11 TP54321098765
#                   TTTT22P__210 ___TTTT L H      111   1
                    3210__BPI___ OIO4567 O I
                    ____PP0BBODS BLL____
                    PPPPBB10FCOY EDDVVVV
1:     1  XXXXXXXXXXZZZZXXXXXXXXXXXXXZZZZXXXvXXXXXXXXXXXXXXXXZZZZZZZZZ
1:     48 XXXXXXXXXXvvvZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1:     49 XXXXXXXXXXZvvZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1:     50 XXXXXXXXXXZvZZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1:     51 XXXXXXXXXXZZZZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX^ZZZZZZZZ
1:     70 XXXXXXXXXXvvvZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1:     71 XXXXXXXXXXZvvZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1:     72 XXXXXXXXXXZvZZXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

The datalog above shows failures that occurred during the Functional Tri-state test. The “**V**” character indicates an output which failed the tri-state level because it was below the VOL reference voltage. The “**^**” character indicates an output which failed the tri-state level because it was above the VOH reference voltage. The functional datalog makes it very easy to see which signal is failing.

Functional Z–State—Trouble Shooting:

To begin trouble shooting, enable the datalogger and observe the test results. If a DUT standard is available, test it and observe the result.

When the Functional Z–State test fails, the failure appears as one or more incorrect output signals. The datalogger will indicate which device pin(s) failed. If the test vector pattern is testing only for high impedance output states, remove the device from the test socket and repeat the test. An open socket will pass the test. If the test vector pattern is testing for both high impedance states and valid logic states an open socket test will fail the valid logic states.

This test is affected by the VOL/VOH comparator levels, the IOL/IOH currents which are supplied by the current loads and by the VREF voltage supplied by the current loads.

Additional time may be required to transition from a valid logic level to the tri–state level, so it may be necessary to slow down the test frequency and move the output strobe timing to a later point within the test cycle in order to get the test to pass.

Functional Z–State—Key Points:

- Purpose: to insure that bi–directional and high impedance outputs are capable of achieving a high impedance or off state
- Dynamic functional test must be executed
- Test conditions are not defined in device specifications
- Test requires loads on outputs to provide intermediate voltage
- Test is affected by external capacitance on outputs

Open Drain / Open Source Outputs

The device specification will indicate if any signal pins have open drain or open source outputs. Be sure to note this when reviewing the device specification because these outputs may require special considerations. Open drain outputs can only drive low—they only sink current. They cannot drive high because there is no pull–up circuitry in the DUT. The output signal must be pulled high by some external means such as a resistor or dynamic current load.

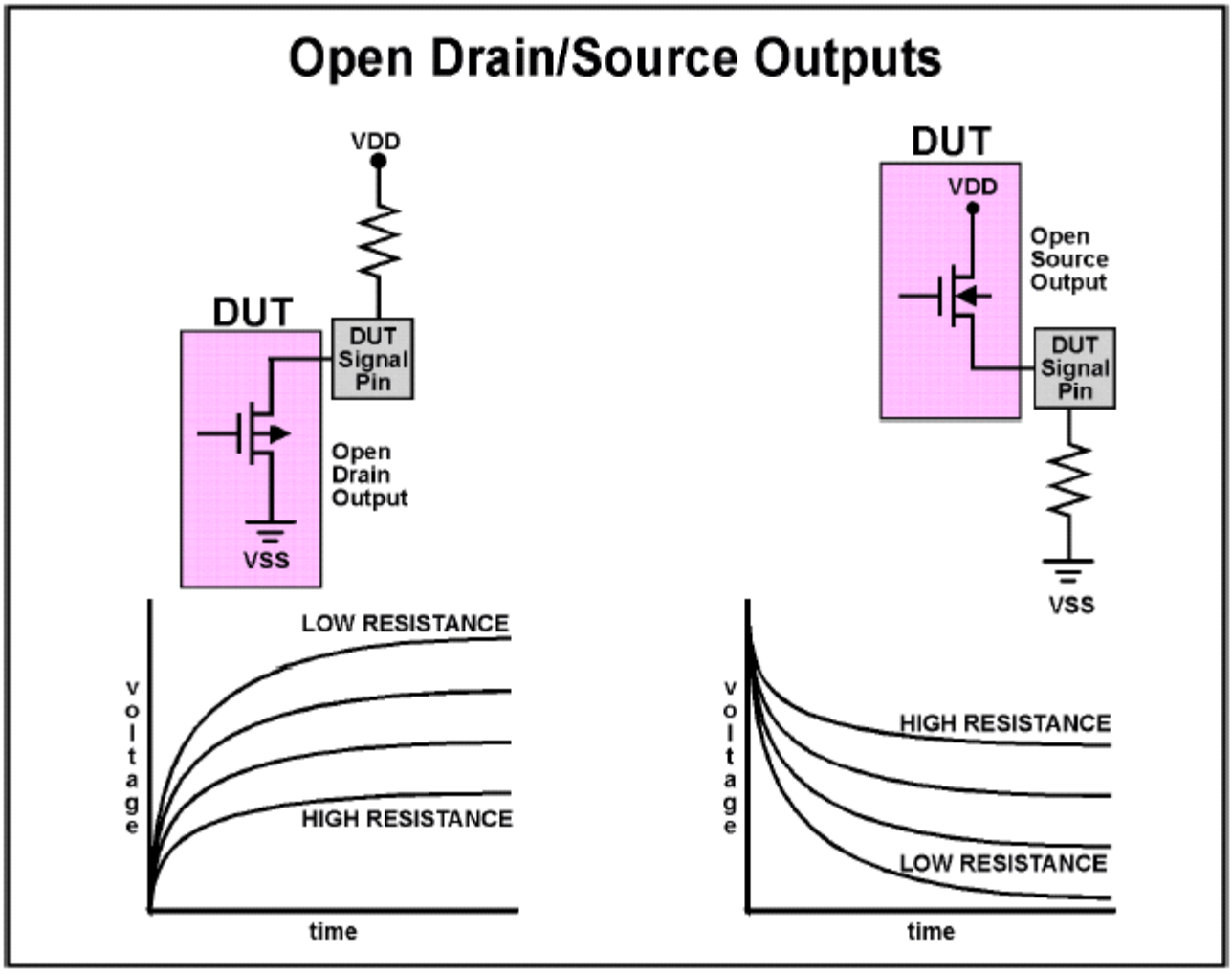


Figure 7-29

Open source outputs are just the opposite. They can drive high but not low—they can only source current because there is no pull–down circuitry in the DUT. The output signal must be pulled down by some external means such as a resistor or dynamic current load.

When executing high speed functional tests, consideration must be given to the value of an external load resistor (or the current supplied by a current load). The time required for open drain output signal to transition from a low to a high depends on the external load current and the associated capacitance. If too large a resistor (or too small a load current) is selected, the output may not reach its logic level soon enough to pass the functional test vector pattern.

These outputs can be sensitive to changes in test hardware. Different results may be experienced between wafer test, hand test and auto handler test due to the different capacitive loads presented.

Open Drain / Open Source — Key Points:

- May switch too slowly for high speed functional testing
- Outputs require current loads and reference voltage
- Output timings affected by external capacitive loading

Functional Test Review

1. If a device specification defines VDD as 5.0V ± 10%, what would be the value of VDDMIN?

2. If a device specification defines VDD as 5.0V ± 5%, the functional tests should be executed at what voltage?
   a) 5.00V
   b) 4.75V and 5.52V
   c) 4.50V and 5.50V
   d) 4.57V and 5.25V
   e) None of the above

3. The purpose of an AC output load is to simulate the environment the output would encounter when connected to one or more device inputs.
   a) True
   b) False

4. The VIL/VIH specifications are verified by executing:
   a) A DC test
   b) A Functional test
   c) An AC timings test
   d) None of the above

5. When a test is made to guarantee the VIL/VIH specification and a failure occurs, the failing pin (as seen on the failing datalog) will be:
   a) An input pin
   b) An output pin
   c) A power supply pin
   d) None of the above

6. If a VIH test is made with VIH set at 2.0V and the result of the test is a failure, how could the voltage be changed in order to make the test pass?
   a) Lower the VIH voltage by 0.4V
   b) Raise the VIH voltage to VDD
   c) Raise the VDD voltage by 0.4V
   d) None of the above

7. When performing the VOL/IOL VOH/IOH test functionally, the comparators are set to the specified VOL/VOH levels and the dynamic (or resistive) current loads are used to supply IOL/IOH. When this test results in a pass condition, all four parameters (VOL/IOL VOH/IOH) are guaranteed to have met the device specification.

a) True
b) False

8. When debugging, if the functional VOL/IOL VOH/IOH test fails, it may be necessary to increase the test period (slow the test frequency) and repeat the test in an attempt to verify correct operation of VOL and VOH.

a) True
b) False

9. When performing the Z-State (high impedance) test as a functional test, some method of output loading must be used to provide an intermediate voltage level. This voltage level will be sensed by test comparators when the output pin enters a high impedance state. A correct value for the intermediate voltage level would be:

a) 5.0V
b) 2.0V
c) 3.0V
d) 0.4V
e) All of the above

10. The purpose of performing a gross functional test is:

11. What is the advantage of using equation based timing?

12. Open drain outputs do not have the ability to (choose the most correct answer):

a) Drive a logic 0
b) Drive a logic 1
c) Source current
d) B and C

13. What is the meaning of RO in the statement “The input control signal is in RO format.”?

14. What signal format must be used to guarantee both setup and hold times?

15. When data changes only at T0, what is the signal format called?

a) DNRZ
b) RZ
c) RO
d) None of the above

16. What would the signal look like for an SBC/XOR signal when the functional data is a logic 0? Draw the wave shape:

17. What signal format switches between a high impedance state and a drive condition (controls driver FET switch)?

18. What is the formula used to find the test cycle time when the operational frequency is known?

19. What is the advantage of performing a functional opens and shorts test rather than a DC opens and shorts test?

20. When performing a functional opens and shorts test current is forced by:
    a) The PMU.
    b) The programmable (dynamic) current loads.
    c) Voltage is forced, not current.
    d) None of the above.

21. A functional opens and shorts test uses the comparators in the Pin Electronics to detect open shorted conditions.
    a) True
    b) False

Chapter 8 Testing AC Parameters

Objectives:

This section explains:

- The various types of AC tests
- How each AC test is made
- Why each AC test is made
- Various methods for performing AC tests
- Problems and compromises associated with AC testing

AC Parametric Testing

The purpose of AC testing is to guarantee that the device meets all of its timing specifications. AC testing is performed by setting up the appropriate timing values (edge placements) and signal formats as defined in the device AC specifications. A test vector sequence is then executed which will exercise the logic functions necessary to insure that all AC timing parameters have been tested.

There are two methods which may be used to functionally verify AC device specifications. All AC timing parameters can be set to their worst case conditions and a functional test executed. This approach is fast and guarantees that the device meets the design specifications. However, if a failure occurs it is not apparent which parameter is causing the failure.

An alternate approach is to test the AC parameters individually. In this case only Data bus Setup Time (for example) would be set to the values defined in the device specification, with other parameters relaxed. A functional vector sequence would be executed and if a failure occurs it is immediately known that the cause of the failure was due to the Data bus setup time. The testing would then continue until all AC parameters are verified. This approach offers more detailed information regarding yield issues, but increases test time.

Read & Record

Read and record refers to a style of testing in which the measured values for DC and AC parameters are measured and saved to either paper (via a printer) or a disk file. This type of testing enables the exact performance of the device to be analyzed. Military specifications often require this style of testing. The test system datalogger and the system utilities can be used to report this information.

Go-Nogo Testing

Go-nogo testing differs from read and record in that the exact performance values of the device are not known. Instead, each parameter is given a pass/fail limit. If the test passes, the device has met or exceeded the test requirements but the amount of margin is unknown. This is how most production testing is done because it is much faster than read and record.

Compromises

Device specifications may exist which cannot be implemented on the available test equipment. When these conditions arise compromises must be made or the proper test equipment must be found. With today's fast devices, maximum frequency testing (also call *at speed test)* can be difficult to perform and is often compromised.

One example of a compromise for maximum frequency testing is to execute a test with worst case input/output timings, but at a reduced test frequency (usually because the test system can not operate fast enough). When this is done, the test frequency is set to something less than the device specification and one or more input or output parameter are set to specification and verified. In this way, all input timings like setup/hold timings and pulse widths can be verified. The output propagation delays can also be tested to insure that the device speed meets the specification. So rather than testing the device at maximum frequency, individual parameters are tested in an attempt to guarantee the at speed performance.

Another approach, is to use *mux testing.* Mux testing involves running the DUT at twice the speed of the test system. ATE operating frequencies are sometimes specified as 50/100MHZ or 100/200MHZ, the first number indicates the uncompromised test frequency, for example a 100/200MHZ system can produce a minimum test cycle of 10ns and can perform all functions at that speed. The second number indicates the speed at which the test system can provide data to the DUT. During MUX testing the test system supplies two input signals and tests two output signals per tester cycle, so the DUT is running twice as fast as the test system. Mux testing usually has restrictions associated with it, like limited pulse formatting, limited vector memory sequencing, restrictions on output strobe placement, etc. The term Test Rate implies the maximum speed of the test system, the term Data Rate implies the speed at which the DUT is running.

It is the responsibility of the test engineer to develop compromises, when necessary, that will guarantee the device specifications to the fullest extent possible. When compromises seem unavoidable, discuss the problem(s) with the chip's design engineer; he or she may know a way to solve or work around them. Be sure to carefully document any compromises and verify with the design engineer and other responsible people that the device is being adequately tested.

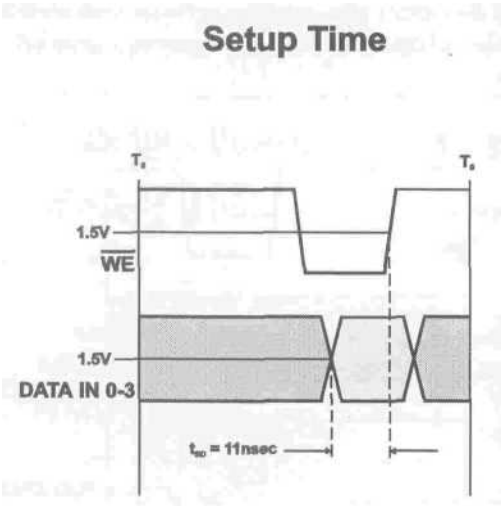
Standard AC Parameters

There are standard Ac parameters just like there are standard DC parameters. Lets look at the most common AC parameters next.

Setup Time

Parameter	Description	Min.	Max	Units
t_{SD}	Data Set-up to Write End	11		nsec

Setup time is the minimum amount of time that data must be present before a reference signal reaches a certain voltage point. For example, DataIn setup to the $\overline{WE}$ (low to high), measurements are referenced at the 1.5V point. See Figure 8-1.



Data Input Setup Time Relative to $\overline{WE}$ Low to High Transition .

Figure 8-1

Key Points:

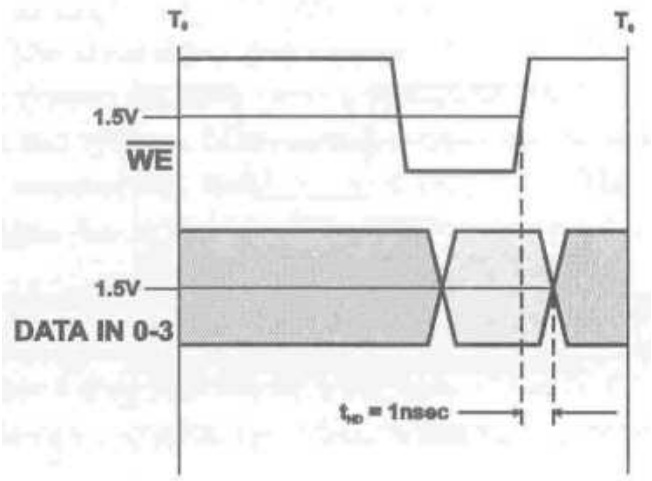
- Purpose: to guarantee that input data can be read (or latched) within a minimum amount of time before a reference signal occurs
- Performed only on input pins
- Setup time can be a negative number (occurs after reference signal)
  Setup_time = (reference_signal - result_of_binary_search).

Hold Time

Parameter	Description	Min.	Max	Units
t_{HD}	Data Hold from Write End	1		nsec

Hold time is the minimum amount of time that data must be present after a reference signal reaches a certain voltage point. For example, DataIn hold time after the $\overline{WE}$ (low to high), measurements are referenced at the 1.5V point. See Figure 8-2.

Hold Time



Data Input Hold Time Relative to $\overline{WE}$ Low to High Transition

Figure 8-2

Key Points:

- Purpose: to guarantee that input data can be read (or latched) within a minimum amount of time after a reference signal occurs
- Performed only on input pins
- Hold time can be a negative number (occurs before reference signal)
- Hold_time = (result_of_binary_search - reference_signal).

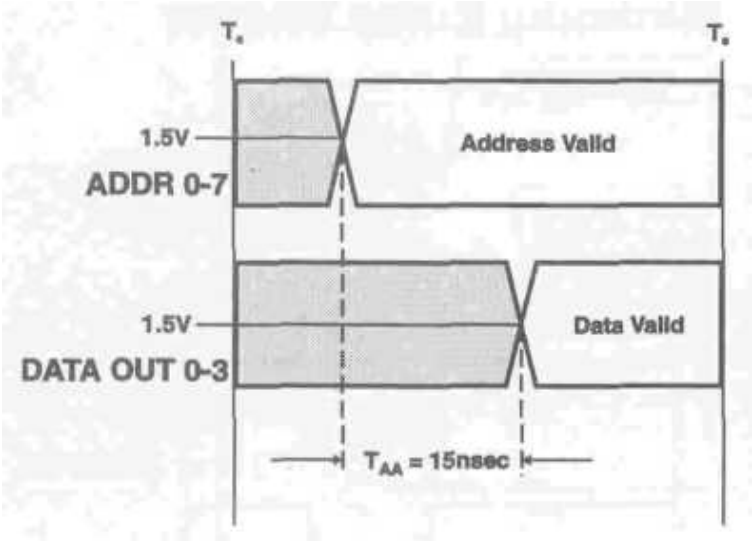
Propagation Delay Measurements

Parameter	Description	Min.	Max	Units
t_{AA}	Address to Data Valid		15	nsec

Propagation delay time is the amount of time between the transition of one signal and the resulting transition of another signal, measured at a specific voltage level (usually 1.5V). Most propagation delay measurements are made from an input signal to an output signal. For example, Figure 8—3 shows the propagation delay from the time an address becomes valid until the slowest data output pin crosses 1.5V with valid data.

Sometimes a propagation delay between two output signals must be measured. This will require a search on both signals to measure their relative positions. An output propagation delay may also be referred to as a "Critical Path Measurement".

Output Propagation Delay



Output Propagation Delay Relative to Address Valid Time

Figure 8-3

Key Points:

- Purpose: to guarantee that an output signal can occur within a specified amount of time after the occurrence of a reference signal
- Performed only on output pins
- Can only be a positive number (occurs after reference signal)
- Prop_delay_time = (result_of_binary_search - reference_signal).

Minimum Pulse Widths

Parameter	Description	Min.	Max	Units
t_{WL}	Minimum clock low time	20		nsec
t_{WH}	Minimum clock high time	25		nsec

Pulse widths often consist of two unique parameters:

Minimum pulse low time (or Minimum clock low width), the minimum amount of time which the pulse can remain in the logic 0 state

Minimum pulse high time (or Minimum clock high width), the minimum amount of time which the pulse can remain in the logic 1 state.

When performing any test that requires the movement or adjustment of a control edge, be sure that all timing edges set relative to that edge are properly maintained (e.g. setup/hold/prop delays). Figure 8-4 illustrates this measurement.

Minimum Pulse Widths

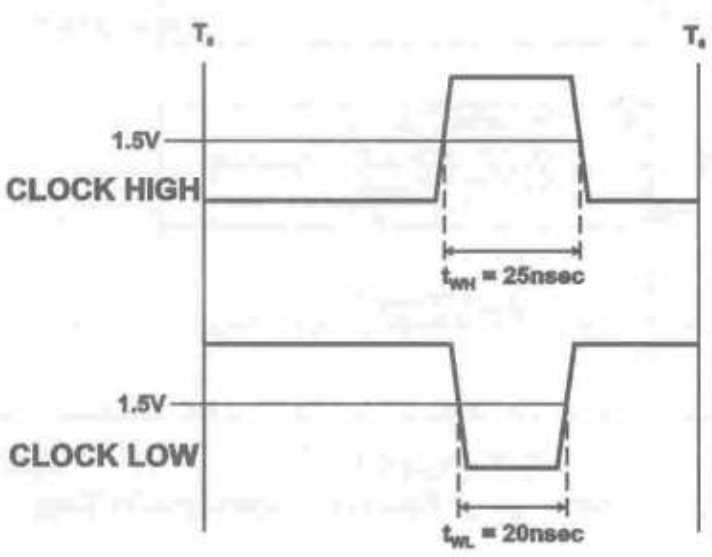


Figure 8-4

Key Points:

- Purpose: to guarantee minimum operational values for pulse low and high timings
- Be sure to adjust all related edge timings when control edge is changed

Maximum Frequency

Parameter	Description	Min.	Max	Units
f_{MAX}	Maximum clock frequency		22.2	MHz

The maximum operating frequency, sometimes referred to as "at speed test", involves testing the device at the maximum speed indicated on the device specification sheet. This is often a difficult test to perform because all worst case timings are applied to the device, which can cause increased test noise and the noise may cause the device to fail. When implementing at speed test, worst case setup/hold times, minimum pulse widths and worst case propagation delays are verified at the maximum device operating frequency. Be sure to carefully read the device specification to insure proper timings.

When performing any test that requires the movement or adjustments of a control edge (a clock signal for example), be sure that all timing edges set relative to that edge are properly maintained (e.g. setup/hold/prop delays). See Figure 8-5.

Maximum Frequency

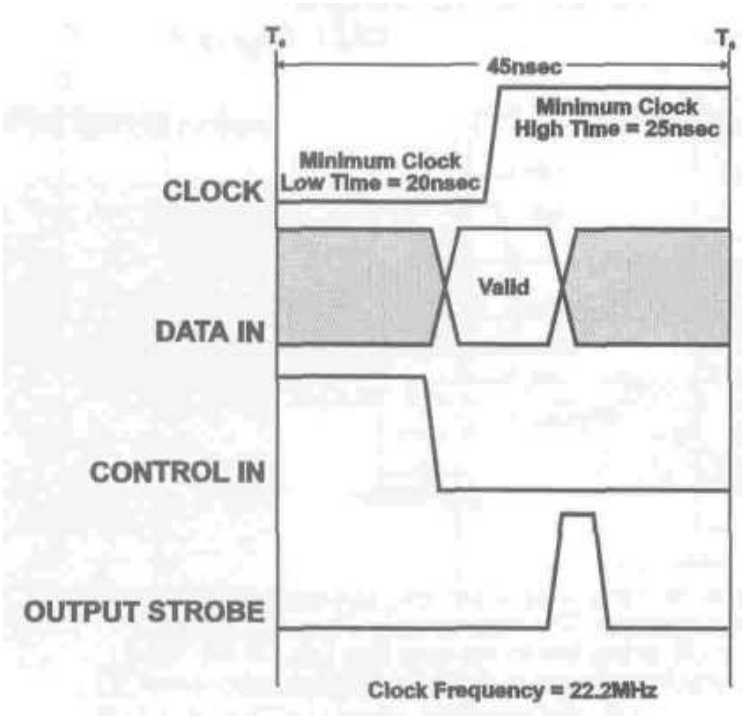


Figure 8-5

Key Points:

- Purpose: to guarantee the maximum device operating frequency
- Be sure to adjust all related edge timings when a control edge is changed
- Maximum Speed testing can cause increased test noise

Output Enable Time

Parameter	Description	Min.	Max	Units
t_{DOE}	$\overline{OE}$ Low to Data Valid		10	nsec

Output enable time is the time it takes an output to switch from a high impedance state to driving valid logic levels. The time is measured from a control signal to a switching output. For example, t_{DOE} (Output Enable to Data Out Valid) is the time from $\overline{OE}$ going high to low until all O0-O3 pins read valid logic levels. This test requires that the outputs be connected to a load with a reference voltage set at an intermediate logic level (2.0V).

In most cases valid logic levels are defined as VOL/VOH, but verify the correct values to be used by consulting the device specification. See Figure 8-6.

Output Enable Delay

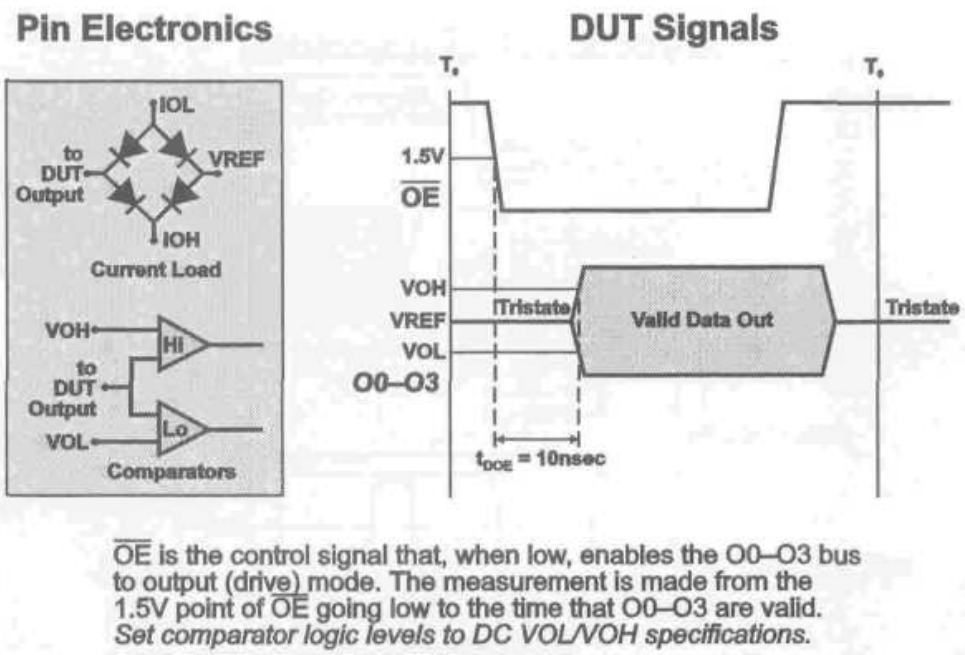


Figure 8-6

Key Points:

- Purpose: to guarantee that a high impedance output buffer can transition from a high impedance state to an on condition and drive valid output levels within a specified amount of time from the occurrence of a reference signal
- Performed only on high impedance outputs and bi-directional device pins
- Test result can only be a positive number when performing a binary search

Output Disable Time

Parameter	Description	Min.	Max	Units
t_{HZOE}	$\overline{OE}$ High to Data High Z		8	nsec

Output disable time is the time it takes an output to switch from driving valid logic levels to a high impedance state, as measured from a control signal. For example, t_{HZOE} (Output Enable High to Outputs 0-3 High Z) is the time required from the point of $\overline{OE}$ going low to high until the outputs stop driving valid logic levels. This can be seen in Figure 8-7 below.

For this test, high impedance state is defined as invalid output levels, meaning a voltage which is greater than VOL and less than VOH. The outputs must be connected to a load with a reference voltage set to an intermediate logic level (2.0V). The output loading causes the bus to be pulled to the intermediate (invalid) level once the output drivers turn off. Verify the correct values in the device specification.

Output Disable Delay

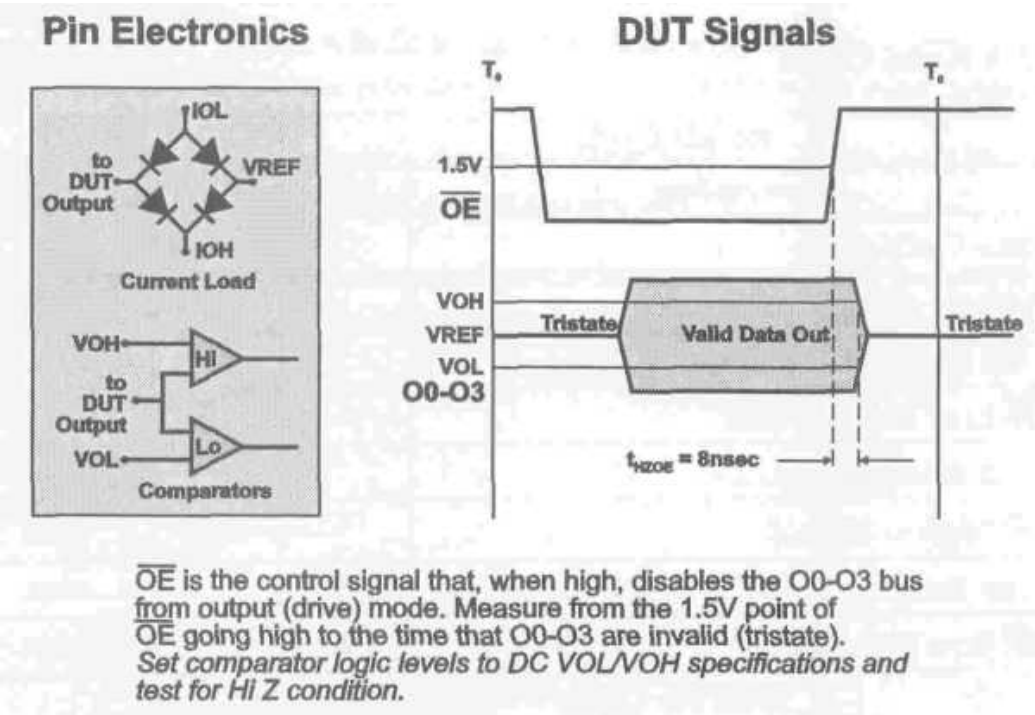


Figure 8-7

Key Points:

- Purpose: to guarantee that a high impedance output buffer can transition from a drive to a high impedance condition within a specified amount of time from the occurrence of a reference signal
- Performed only on high impedance outputs and bi-directional device pins
- Test result can only be a positive number when performing a binary search

AC Specifications from 256 x 4 Static RAM Data Sheet

The table below illustrates the AC specifications for a 256 x 4 Static RAM chip. The timing parameters for AC READ and WRITE cycles are shown.

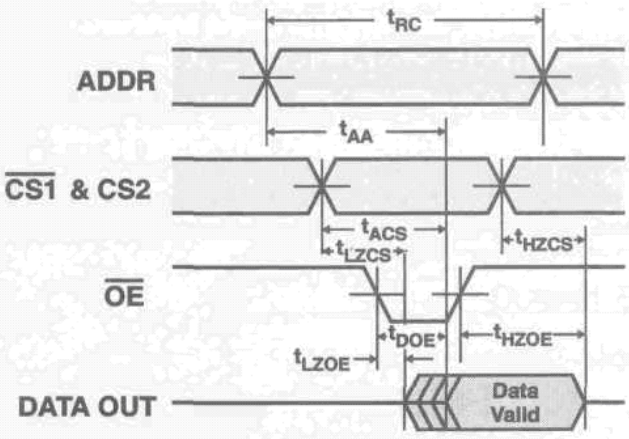


Figure 8-8 Read Cycle

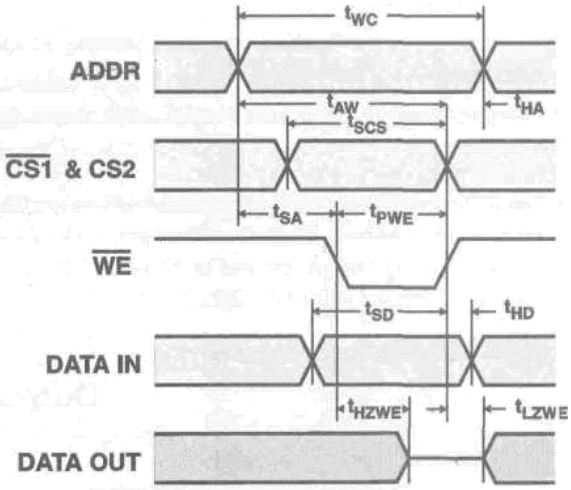


Figure 8-9 Write Cycle

READ Cycle				
Parameter	**Description**	**Min**	**Max**	**Units**
t_{RC}	Read Cycle time	15		nsec
t_{AA}	Address to Data Valid		15	nsec
t_{ACS}	Chip Select to Data Valid		10	nsec
t_{DOE}	$\overline{OE}$ Low to Data Valid		10	nsec
t_{HZCS}[6]	Chip Select to High Z		8	nsec
t_{HZOE}	$\overline{OE}$ High to High Z		8	nsec
t_{LZCS}[6]	Chip Select to Low Z	2		nsec
t_{LZOE}	$\overline{OE}$ Low to Low Z	2		nsec
WRITE Cycle				
Parameter	**Description**	**Min**	**Max**	**Units**
t_{WC}	Write Cycle time	15		nsec
t_{HZWE}	$\overline{WE}$ Low to High Z		8	nsec
t_{LZWE}	$\overline{WE}$ High to Low Z	2		nsec
t_{PWE}	$\overline{WE}$ Pulse Width	11		nsec
t_{SD}	Data Set–up to Write End	11		nsec
t_{HD}	Data Hold from Write End	1		nsec
t_{SA}	Address Set–up to Write Start	2		nsec
t_{HA}	Address Hold from Write End	2		nsec
t_{SCS}	$\overline{CS}$ Low to Write End	11		nsec
t_{AW}	Address Set–up to Write End	13		nsec

Developing Functional Timing

Device timing as defined in the device specification must be carefully reviewed to develop a test timing diagram. With the goal of a low test time, you must match the capabilities of the tester with the device timing requirements. With a highly capable tester you may be able to test all parameters in a single vector run.

If the target test system has limited resources, multiple test runs must be made, using new timing conditions with each functional test execution until all timing parameters have been verified. The first step in developing the test timing is to define the test cycle (also called test period or test frequency).

Spend as much time as necessary to fully understand the device timing. This is one of the most important steps in test program development. If the test vectors are being developed from simulation data, review the timing used during simulation. The test program will need to duplicate the simulation timing. The simulation timing should reflect the device specification timing and must also be compatible with the performance of the test system.

Study the timing diagram for the 256X4 Static Ram on the previous page and develop test timing for both the Write and Read cycles. Start first with the Write timing diagram and develop your timing as follows:

1. Define the Cycle time, draw two vertical lines to represent the cycle and show the duration of the cycle.

2. Determine how the control signals will fit within the cycle (what will the signals look like and where will the signals transition). Draw the signals on your timing diagram and show the edge placement timings.

3. Define how the data signals will fit within the cycle (what will the signals look like and where will the signals transition). Draw the signals and show edge placement timings.

4. Once the signals are drawn and the timings indicated, determine the signal formats that will be needed for each signal. Consider both the active and non-active states of the control signals.

Write Cycle Timing

Using the tuning specification for the 256 x 4 static RAM, a Write Cycle test timing diagram can be developed:

1. Determine the cycle time from the t_{WC} parameter.
2. Determine which signal controls the write function. It can be seen that the rising edge of $\overline{WE}$ controls when the input data is read into the RAM—this is the controlling signal. If you look closely you can see that all other timings are referenced to the $\overline{WE}$ signal.
3. Determine the placement of the remaining input signals. The correct position of the $\overline{WE}$ signal within the cycle is determined by the required setup and hold times of signals referenced to it. Address timing (tAw) requires 13nsec of setup time and 2nsec of hold time. The sum of these two parameters is the cycle time (15 nsec) and also defines the rising edge of $\overline{WE}$ as 13nsec into the cycle. Chip selects and input data timing must also be determined relative to $\overline{WE}$. There is a reference to "data out" on the write cycle timing specification but this can be ignored because output data is not tested during a write cycle.

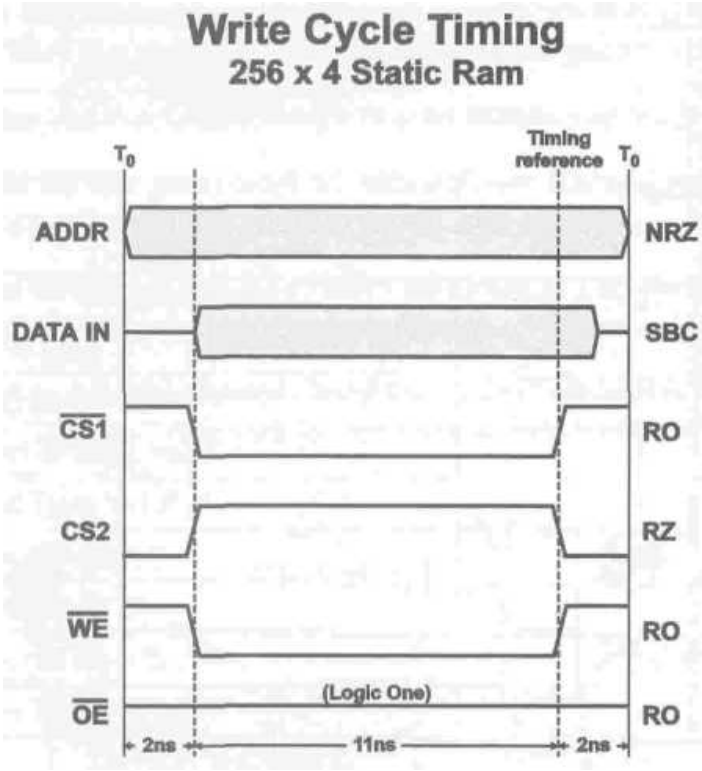


Figure 8-10

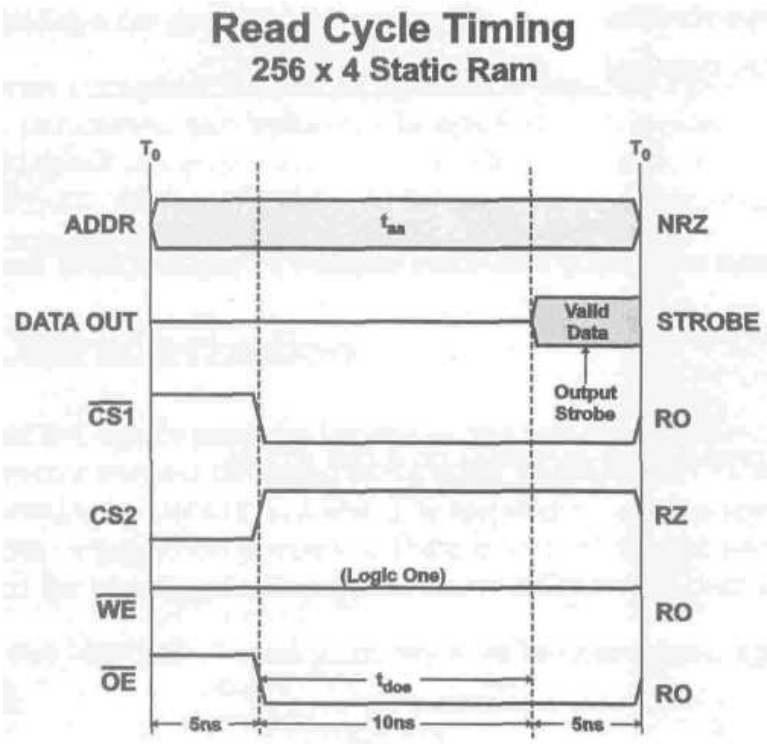
Draw a timing diagram showing the true relationship of all the signals so you correctly understand the timing relationships. Figure 8-10 is drawn to scale, meaning a grid was set to represent each nanosecond of the cycle and the signal transitions are drawn at their correct times. Often the timing diagrams shown in data sheet specifications are not drawn to scale and do not accurately show the timing.

By having a correctly drawn timing diagram, it will be much easier to know what each AC test is doing and to debug it. A debug loop will allow you to compare an oscilloscope picture of test operation to the timing diagram.

Read Cycle Timing

1. 1. Using the Read Cycle Timing specification for the 256 x 4 static RAM, review the AC timing specifications and develop a test timing diagram: Determine the cycle time from ^{t}RC. This parameter is the amount of time required to access the data stored in RAM memory. To test the DUT the data must first be accessed and then verified,
   so the actual test cycle time will be slower than the ^{t}RC specification.
2. 2. Determine which signal is the controlling signal for the read function. Notice that the falling edge of $\overline{OE}$ controls when the outputs will be enabled. Once the outputs are enabled the data stored in memory can be verified. $\overline{OE}$ is the controlling signal because all other timings are referenced to it.

Draw a timing diagram showing the true relationship of all the signals. Remember that input signals must be maintained long enough to test (strobe) the outputs. In Figure 8-11 the cycle time has an extra 5nsec added to the end of the cycle. This is where the output strobe is placed to verify the output data.



Like Figure 8-10, this diagram is drawn to scale. You may need to draw many timing diagrams for one test program, depending on the complexity of the circuit being tested.

AC Testing Review

1. The amount of time that data must be present after a reference signal reaches a certain voltage point is:
   a) Setup time
   b) Propagation time
   c) Minimum pulse width time
   d) None of the above.

2. If the device specification defines the setup time as 12nsec but the device functions properly with a setup time of l0nsec, the device:
   a) Fails to meet the device specification
   b) Exceeds the device specification
   c) I do not understand the question!
   d) None of the above
   e) All of the above

3. It is not possible (or at least not correct) to measure negative propagation delay time.
   a) True
   b) False

4. Propagation delay measurements are only made on output signals.
   a) True
   b) False

5. The output enable test is a measurement of an output going from driving valid data to a high impedance state.
   a) True
   b) False

6. The output enable and output disable tests require an external reference voltage to indicate when the output has entered a high impedance state.
   a) True
   b) False

7. Read and record is a method of testing often used in production testing because it is the fastest test method.
   a) True
   b) False

8. The Go/Nogo testing method is often used when the exact performance of a device must be recorded, as in military testing.
   a) True
   b) False

Chapter 9 Device Characterization

Objectives:

This section explains:

- Issues concerning device characterization
- Methods and tools used to perform characterization
- Common characterization parameters

The purpose of characterizing a device is to determine the extremes at which the device will operate. Binary and linear search routines and Shmoo plots are often used to accomplish this task. Voltage, current and timing parameters are all likely candidates for device characterization.

In order to characterize a device parameter it must be possible to generate a pass/fail result. For example, when characterizing the VIL parameter, zero volts may be applied to all inputs and the functional test pattern executed. This activity should result in a pass condition. If the VIL level is then set to 2V and the functional test pattern is executed the test should result in a failure. At this point we have known pass/fail values and a search can be performed to determine the exact VIL voltage threshold.

Test Vectors and Characterization

Characterization requires that the device perform functions that will enable its extreme operating limits to be found. In many cases a test vector pattern designed specifically to characterize a single device function is needed in order to extract meaningful data from a test. The test vector patterns used for functional testing may or may not be adequate for characterization purposes. The function of the test vectors must be clearly understood before being used for characterization, particularly AC parameter characterization!

The Binary Search

The binary search, also known as successive approximation, is an efficient method of finding a specific value contained within a larger range of values. This technique uses the "divide-and-conquer" approach.

The process begins by first dividing the range of possible values in half and testing this midpoint value against the specific (but unknown) value. (The test is based on a greater/less than result). If this midpoint value is less than the unknown value then the unknown value must reside in the upper half.

If the midpoint value is greater than the unknown value then the unknown value must reside in the lower half. This process continues until the range of possible values has been reduced below the desired measurement resolution (the smallest possible step).

For example, to resolve the number 4 (the unknown) from within a range of 12345678 9, the binary search would first test 4 (the unknown) against the midpoint value 5. Since 5 is greater than 4 the search continues in the direction of the lower half 1 2 3 4 5.

The midpoint is now 3, since 3 is not greater than 4 the search continues in the direction of the upper half 3 4 5. The midpoint of this range is 4; since 4 is not greater than 4 the search continues in the direction of the upper half 4 4.5 5.

The midpoint of this range is 4.5 and is greater than 4. If the desired resolution is 1, the range of possible values is now less than the resolution and the search would end with the result equal to 4.

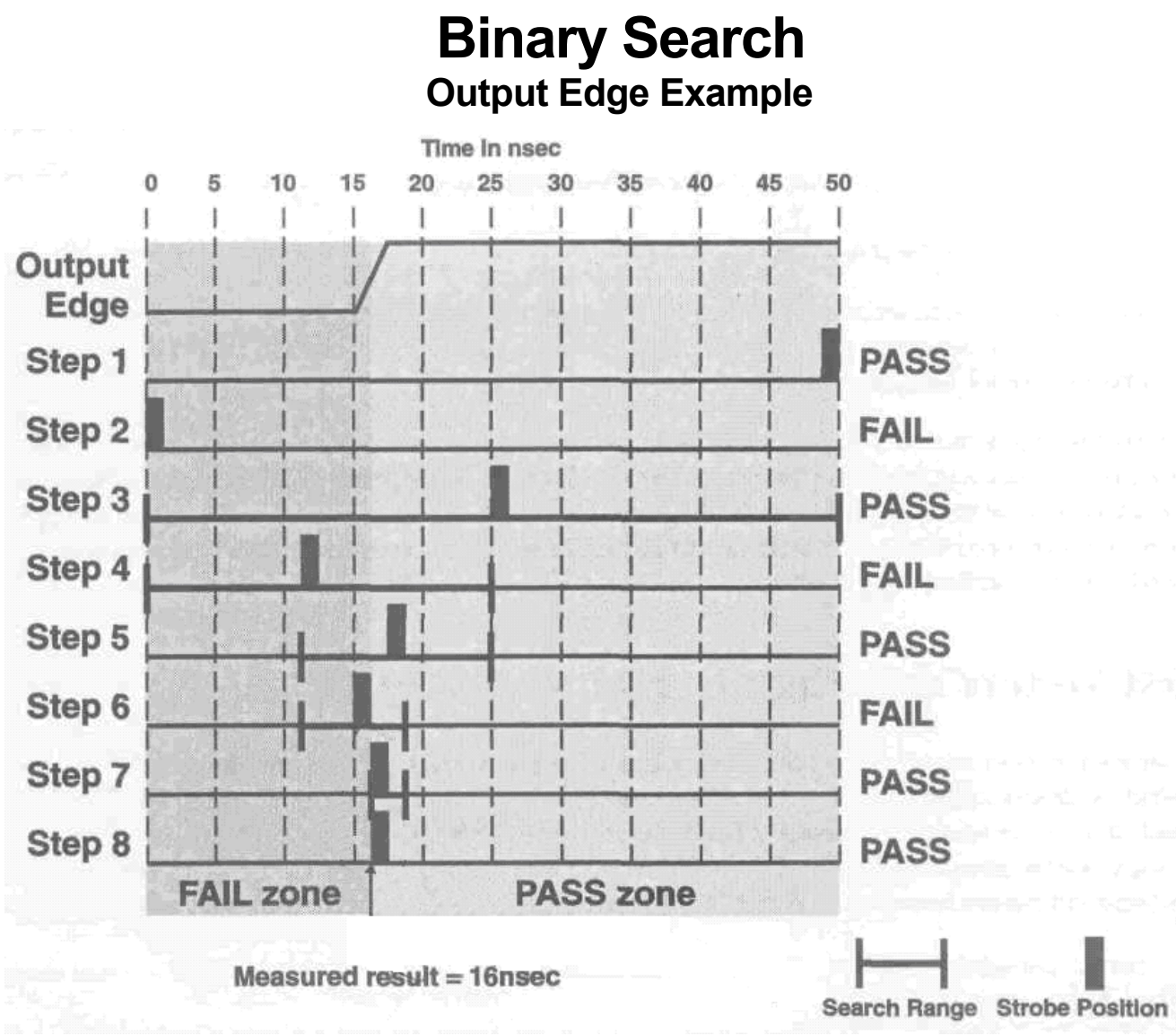


Figure 9-1

Binary Search Test Applications

The binary search is often used in test programs to convert analog magnitudes (voltages, currents and timings) into numerical values. The measurement of test parameters is performed by defining the range of parameter values to encompass both failing and passing zones. The search direction is controlled by the test result.

For example, to measure the output propagation delay of a specific output, a functional test that compares only the desired output transition (low to high / high to low) would be executed for each step of the search. The binary search routine varies the compare strobe through a range of values that includes both failing and passing test results. Because the search direction is controlled by the test results, the binary search will find the point where the test result changes from fail to pass.

The propagation delay is determined by subtracting from the binary search result the programmed timing value for the input signal that controls this output. See Figure 9-1.

Calculating AC Measurements from Binary Search Results

When a binary search is used to determine AC tunings, an equation is often needed to obtain the desired result. The search routine moves a signal edge but is unaware of the placement of the reference signal. Setup time has been defined above as the amount of time data must be present before a reference signal reaches a certain voltage point. For example, data bus setup to the clock low to high transition.

For setup time a binary search is performed by moving the leading edge of the data signal. The search will report the last position of the input data which resulted in a pass condition. To determine the setup time the following formula is used:

```
Setup_time =  (reference_signal  -  result__of_binary_search)
```

When searching for hold time the binary search would move the trailing edge of the data signal. The formula to find hold time is:

```
Hold_time = (result_of_binary_search - reference_signal)
```

When searching for a propagation delay the binary search would move the output strobe signal. The formula to find the propagation delay time would be:

```
Prop_delay_time = (result_of_binary_search - reference_signal)
```

The Linear Search

A linear search is also used in device characterization but there is one main difference as compared to the binary search. A linear search moves the voltage, current or timing unit in a singular (or continuous) direction.

For example, when characterizing the VIL parameter, zero volts may be applied to all inputs and the functional test pattern executed. This activity should result in a pass condition. The linear search would then increment the voltage one step in the positive direction and repeat the functional test. This sequence would continue until the functional test resulted in a failure. At that point the last pass condition would be reported as the maximum VIL threshold value. Certain types of tests require this linear test method, for example Schmitt trigger testing. On average, the linear search method is slower than the binary search and is therefore only used when necessary.

Common Characterization Parameters

VDDMIN	Determine the lowest operational VDD voltage.
VDDMAX	Determine the highest operational VDD voltage.
VIL	Search for the highest input level that will function properly as a logic 0. May be on a group or individual pin basis.
VIH	Search for the lowest input level that will function properly as a logic 1. May be on a group or individual pin basis.
VOL	Find the output low voltage for a given IOL current on a group or individual pin basis.
VOH	Find the output high voltage for a given IOH current on a group or individual pin

	basis.
IOL	Find the output low current for a given VOL voltage on an individual pin basis.
IOH	Find the output high current for a given VOH voltage on an individual pin basis.
Setup Time	Search for the minimum time a signal needs to be valid before its reference signal transitions.
Hold lime	Search for the minimum time a signal needs to be valid after its reference signal transitions.
Propagation Delay	Determine the amount of time required for an output signal to become valid after its reference signal transitions.
Minimum Pulse Width	Determine the smallest pulse width that will allow the device to function properly.
Maximum Operating Frequency	Compress all timing relationships and determine highest operating speed of the device. This may require detailed planning, due to the relationships and relative positioning of signals.

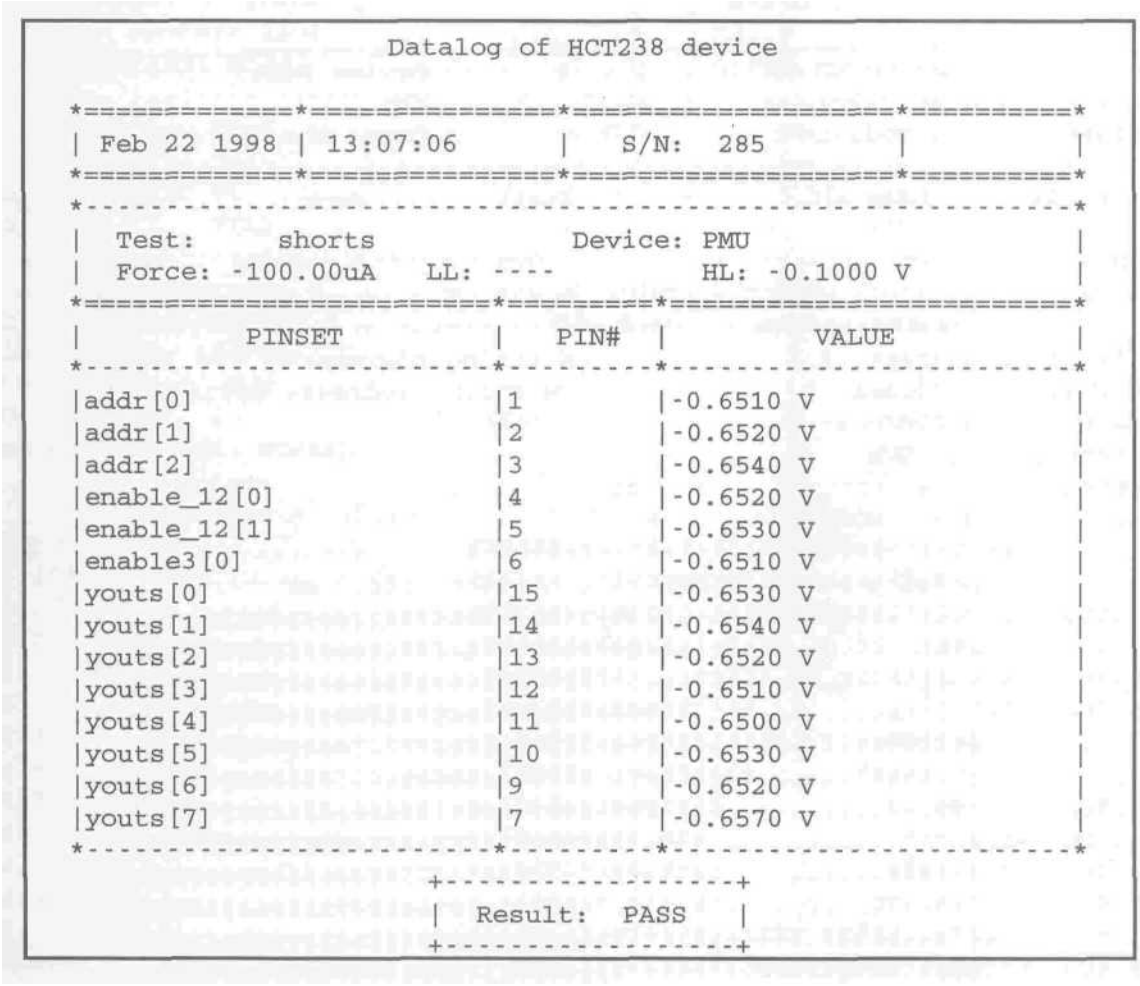
Defining Characterization Parameters

Performing device characterization requires careful planning. The device specification must be reviewed and a list developed of the parameters to be characterized. DC parameters (voltage and currents) as well as input and output thresholds normally do not present a problem. AC parameters, however, may be critical to the application in which the device is used, but may be untestable on a digital test system. AC parameters may also be defined in a manner in which a pass/fail test is not possible. Develop riming diagrams for complex tests—be sure to fully understand how the test will function before you begin to generate the test code. Each test must be able to find a pass/fail region and must report meaningful data. Consult with the design engineer to resolve any questions regarding timing relationships or device functions.

The Test System Datalogger

Each test system has some method of reporting pass/fail results. This information may also include the name of the test, the passing/failing tester pins, the program statement number, etc. The datalogger is a valuable tool, it can be used in device characterization to report exact measured values.

Note: It is important to understand how the datalogger works. Each test system has a particular way of capturing test results under certain conditions the manner in which the datalogger works can affect the logged results.



Datalog of HCT238 device

Feb 22 1998	13:07:06	S/N: 285	

Test: shorts	Device: PMU
Force: -100.00uA LL: ----	HL: -0.1000 V

PINSET	PIN#	VALUE
addr[0]	1	-0.6510 V
addr[1]	2	-0.6520 V
addr[2]	3	-0.6540 V
enable_12[0]	4	-0.6520 V
enable_12[1]	5	-0.6530 V
enable3[0]	6	-0.6510 V
youts[0]	15	-0.6530 V
youts[1]	14	-0.6540 V
youts[2]	13	-0.6520 V
youts[3]	12	-0.6510 V
youts[4]	11	-0.6500 V
youts[5]	10	-0.6530 V
youts[6]	9	-0.6520 V
youts[7]	7	-0.6570 V

Result: PASS

Figure 9-2 Datalog of shorts test

Use of Test System Tools

Use the standard test system tools when possible. These routines generally execute quickly and for the most part are "bug free." Do not spend time to recreate code which is already available and working. The system tools provide a quick method to perform device characterization. Test tools or test utilities that are available on most test systems are Shmoo Plot, Level Search, Binary Search, Linear Search, and datalog. See Figure 9-2.

Shmoo Plots

A Shmoo plot is a way of displaying the results of a test while varying one or more tester resources. A Shmoo plot is typically a two dimensional plot, showing one test parameter on the X axis and another on the Y axis.

The example in Figure 9-3 shows a plot of VDD on the Y axis starting at 5.5V, ending at 3.5V and VIL on the X axis starting at 2.0V and ending at 0.0V. The Y axis has 21 steps and the X axis has 51 steps. For each step on the X/Y axis a test is executed. In this example a functional test is executed at each point on the plot. If the test result is a pass a "$" is printed, if the test result is a fail a "." is printed. Shmoo plots can take a long time to produce.

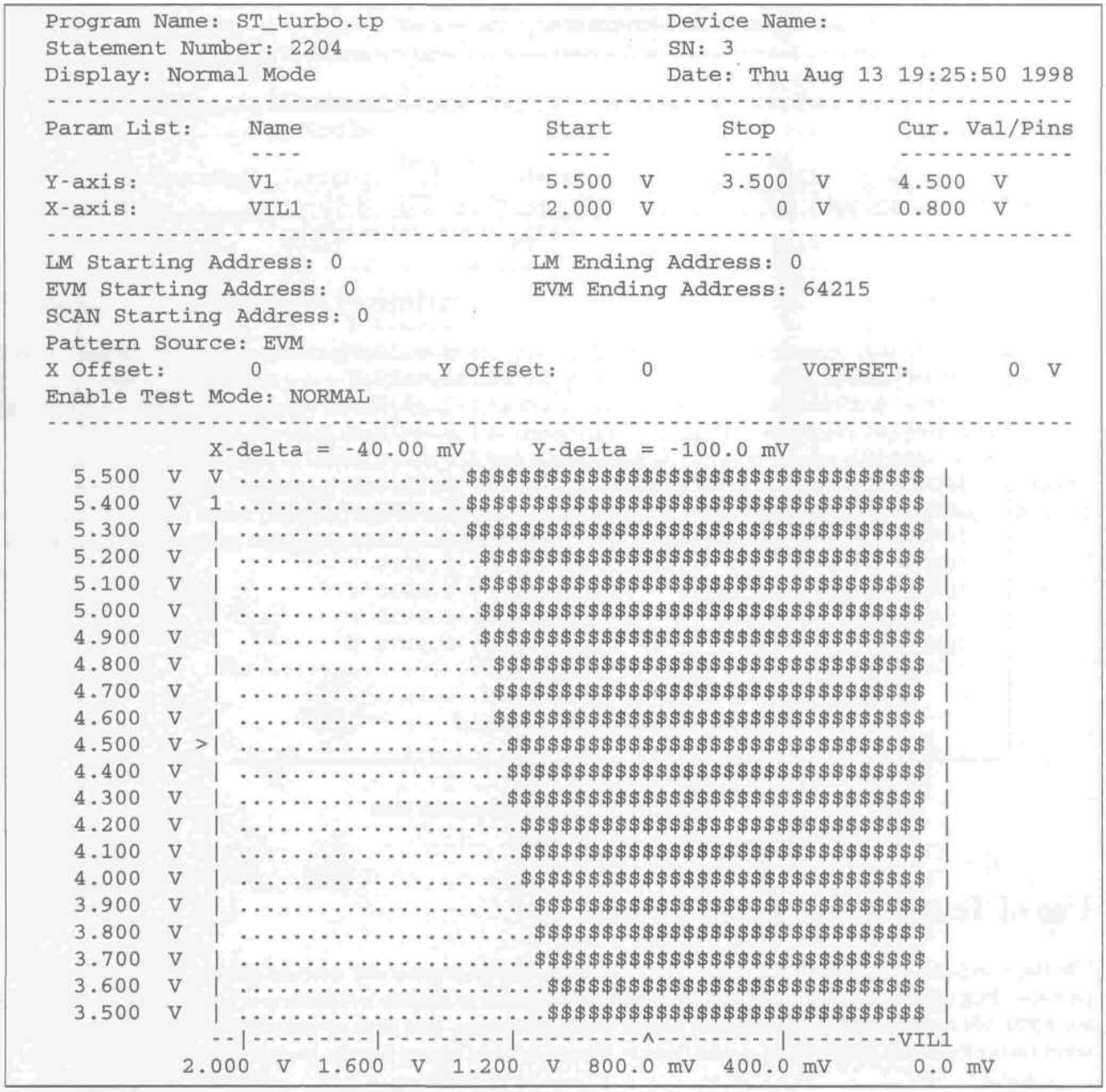


Figure 9-3 VDD vs. VIL Shmoo Plot

Most Shmoo plot utilities have the ability to set tracking parameters which allow other tester resources to follow along with (or track) the main X/Y parameters. When testing CMOS, VIH should be set to track VDD. This prevents the possibility of VDD becoming lower than VIH (which could latch up the device). When executing timing plots it may be necessary to have input or output timings track with the movement of a clock edge.

Figure 9-4 illustrates what is known as a *hole* in a Shmoo plot. Holes are seen when a device fails within its operational range but functions correctly before and after the failing region.

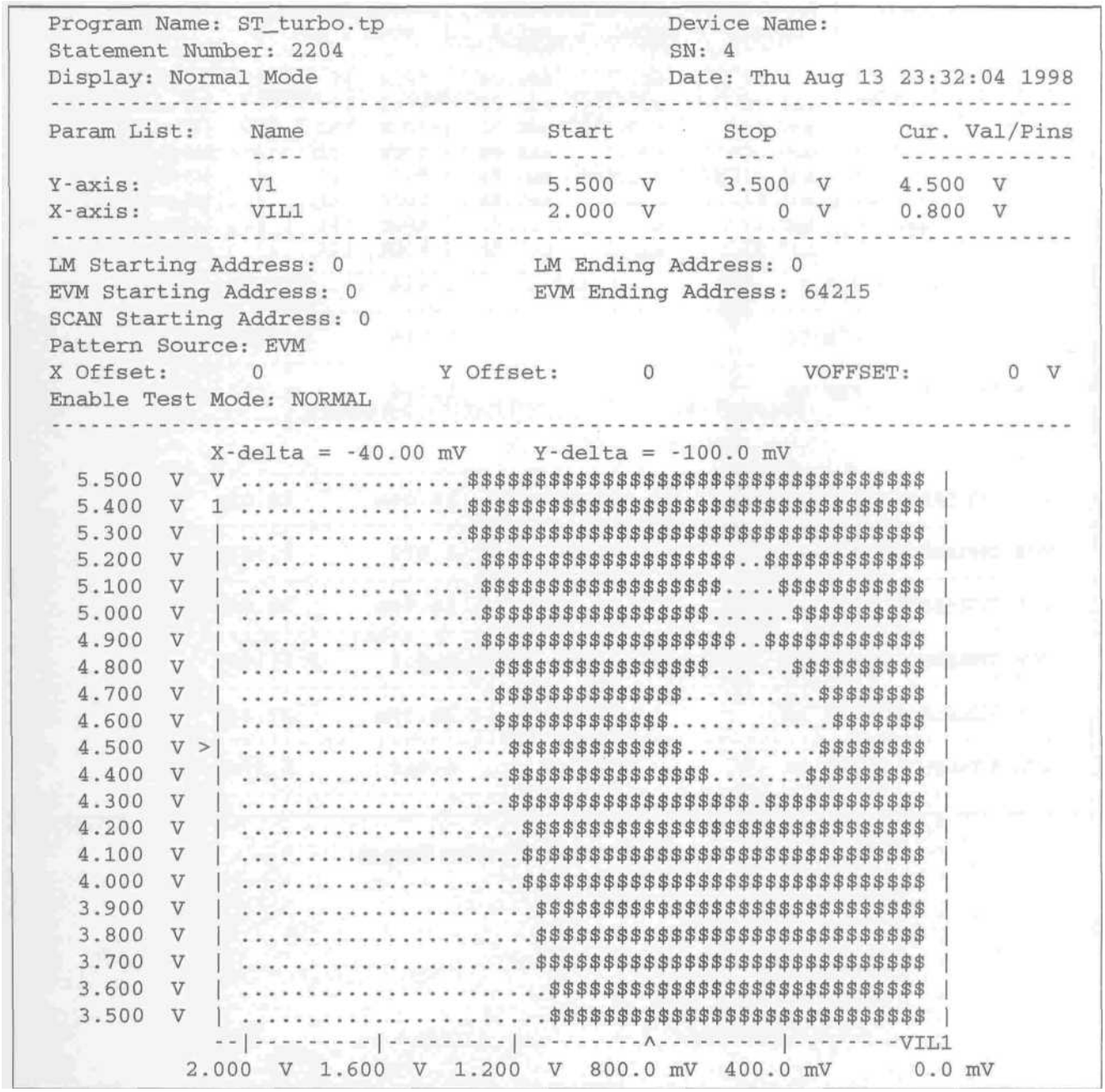


Figure 9-4 VDD vs. VIL Shmoo Plot showing a Hole In plot

Threshold/Level Search

Figure 9-5 is an example of the results obtained from a standard binary level search routine. A software utility was used to find the input/output threshold values, the data was then formatted and printed. Very little program coding is required to produce this type of characterization data.

ST_TURBO - TEST PROGRAM DEVICE CHARACTERIZATION					
INPUT / OUTPUT THRESHOLDS					
TEST PERFORMED			! VDDMIN ! 4.500	! VDDNOM ! ! 5 !	VDDMAX ! 5.500 !
INPUT THRESHOLDS					
VIL	THRESHOLD	PIN 87	1.266	1.346	1.430
VIH	THRESHOLD	PIN 87	1.332	1.436	1.544
VIL	THRESHOLD	PIN 88	1.242	1.340	1.432
VIH	THRESHOLD	PIN 88	1.332	1.436	1.544
OUTPUT THRESHOLDS					
VOL	THRESHOLD	PIN 23	28.00m	36.00m	28.00m
VOH	THRESHOLD	PIN 23	4.396	4.872	5.380
VOL	THRESHOLD	PIN 24	24.00m	26.00m	28.00m
VOH	THRESHOLD	PIN 24	4.396	4.868	5.380
VOL	THRESHOLD	PIN 30	22.00m	25.00m	27.00m
VOH	THRESHOLD	PIN 30	4.376	4.868	5.376

Figure 9-5 Characterization Output

Output Propagation Delay - Binary Search Results

Figure 9-6 is an example of the results obtained from a binary search for output propagation delay measurements. A search utility was used to find the time at which the output signal makes its transition, the data for each output was then formatted and printed.

pin name	Tester Pin	Search Mode	Original Value	Begin Value	Measured Value	END Value	Fail vector
dbus[7]	7	EDGE	45.0ns	1.0ns	20.1ns	80.0ns	10
dbus[6]	9	EDGE	45.0ns	1.0ns	20.3ns	80.0ns	9
dbus[5]	10	EDGE	45.0ns	1.0ns	19.9ns	80.0ns	8
dbus[4]	11	EDGE	45.0ns	1.0ns	20.0ns	80.0ns	7
dbus[3]	12	EDGE	45.0ns	1.0ns	19.5ns	80.0ns	6
dbus[2]	13	EDGE	45.0ns	1.0ns	19.6ns	80.0ns	5
dbus[1]	14	EDGE	45.0ns	1.0ns	19.2ns	80.0ns	4
dbus[0]	15	EDGE	45.0ns	1.0ns	19.6ns	80.0ns	3

Figure 9-6 Output Propagation Delay Results

Device Characterization Review

1. The binary search method is also known by another name. What is the other name?

2. Standard test system utilities such as Shmoo plots and level search routines are effective tools for device characterization.
   a) True
   b) False

3. It is often necessary to develop test vector patterns specifically for device characterization.
   a) True
   b) False

4. Shmoo plots are used because they operate faster than search utilities.
   a) True
   b) False

5. A linear search typically executes slower than a binary search.
   a) True
   b) False

6. What Hold Time formula is used when making a binary search?

Chapter 10 Test Vector Development

Objectives:

This section explains:

- Issues related to test vector generation
- Methods and tools used to generate test vectors
- Issues related to simulation data

Test Vectors

Test vectors are also called test patterns or truth tables. Test vectors are the input and output states which represent the logical functions that the DUT is designed to perform. Input and output data are represented by characters—1/0 are often used for input data, L/H/Z for output data and X for no input drive and no output compare. Any set of characters can be used to represent the truth table as long the test system can correctly perform the proper function for each character.

The test vector sequence is stored in vector memory and each individual vector represents the "raw" data for one test cycle. The input data from vector memory is combined with timing, format and voltage level data and supplied to the DUT via the pin electronics driver circuitry (the data is now "cooked"). The outputs of the DUT are monitored via the comparator circuitry located on the pin electronics cards and compared against the data stored in vector memory at the appropriate strobe time. This type of testing is called stored response because the expected response for the DUT is "stored" in vector memory.

The test vector sequence may have, in addition to DUT data, instructions to the test system. For example, timing may be changed on the fly, which means that timing values or signal formats may change on a vector by vector basis. The input drivers may be switched on and off and the output comparators may also be selectively monitored from one cycle to the next. Many test systems also support micro code instructions such as branching, looping, vector repeats, subroutines, etc. The way tester instructions appear within a vector file differs from one tester to the next. This is one of the reasons that vector translators are required when moving a test program to a different brand of tester.

Test vectors for complex devices are typically extracted from simulation data created during the design process. The simulation data is reformatted for the target test system and may also need to be processed to function correctly. It is not unusual for test vector sequences to consist of millions of individual vector cycles. Test vectors, or the simulation data, may be created by the design engineer, the test engineer, or the simulation engineer. The key to successful vector generation is a thorough understanding of both the DUT and the test system.

Example Vector File

The following example has been created as a typical vector file. It begins by defining the pattern name and the meaning of each character within the vector file. Three time sets are used. The time sets are named Reset, Write and Read. The timing values do not appear within the vector files. The main test program defines the timing values that will be associated with each time set name. The vector data defines when a particular time set will be in use. Some test systems support the use of time set names, others support only the use of time set numbers.

Notice that the I/O signal data uses all of the vector characters: inputs (0/1), outputs (L/H), three-state outputs (Z) and don't care or masked (X) conditions. The micro code statements represent additional vector control. In this example the micro code "Normal" indicates that the vector is executed as a single cycle, the "Repeat 25" statement repeats the vector for 25 cycles. The "Halt" statement terminates the execution of the vector sequence. Micro code statements add great flexibility and power to the test vector patterns, but they are actually part of the test programming language and are dependent upon the target test system.

```
Pattern (ST_TURBO) /* File name */
Character definition:
/* Define what the test system will do for each character */
{
0 = logic 0          driver on        comparator off          /* input */
1 = logic 1          driver on        comparator off          /* input */
L = logic 0          driver off       comparator on           /* output */
H = logic 1          driver off       comparator on           /* output */
Z = float            driver off       comparator on           /* output */
X = don't care       driver off       comparator off          /* ignore */
}
/* Memory       Input       Output       I/O          Time        Micro
   Location     Signals     Signals      Signals      Set         Code */
Start_loc:
0               11010       XXXXX        XXXXX        Reset       Normal
1               01010       XXXXX        ZZZZZ        Reset       Repeat 25
2               11010       HHXXL        01010        Write       Normal
3               01111       LHLLL        LHLHL        Read        Normal
4               11000       LHHHL        10101        Write       Normal
5               01101       HLLLH        HLHLH        Read        Normal
6               11000       LHHXX        11111        Write       Normal
Stop_loc:
7               11111       HLXXL        HHHHH        Read        Halt
End.
```

Working with the Design Engineer

The designer is the one most familiar with the functions of the device and is often responsible for vector generation. The test engineer should meet with the designer as early in the project as possible to discuss the details of testing, review the timing and voltage requirements and discuss at length how the test vectors are to be created. Give the designer a copy of Simulation for Test guidelines (see page 10-7). If necessary explain the resources available on the target test system, as well as the test system limitations.

Creating Vectors by Hand

Most device designs today require far more test vectors for verification than can be manually developed. If you must generate vectors by hand, find a way to automate the process—use macros to help generate the vectors or develop a program in C or Pascal. Block out the device functions and develop routines for each function. Example: Write/Read Bus, Reset, Read/Write Port, Send/Receive Data.

Tester Options (Memory Considerations)

Test vector patterns are executed from a high speed vector memory within the test system. The size of vector memory determines the number of test vectors that can be executed before it is necessary to reload the memory with additional vectors. Some test systems have a memory depth of 4096, while others contain vector memory storage of one megabyte or greater. If it becomes necessary to reload the vector memory during device testing, valuable test time is lost. It may be impractical to test a device that requires large numbers of test vectors on a test system with a small vector memory.

Test Vector Examples

```
#pattern chkbrd,"chkbrd.e9k"
PINDEF_TABLE = 256x4_pins;
TIMING "ram_timings";
chk_st;
; write checkerboard

;            a     d
;            d   d a
;            d   a t
;            r   t a c
;            e   a o s c w o
;            s   I u 1 s e e
;            s   n t / 2 / /
INC write {  00  5 x 0 1 0 1 }
INC write {  01  A x 0 1 0 1 }
INC write {  02  5 x 0 1 0 1 }
INC write {  03  A x 0 1 0 1 }
INC write {  04  5 x 0 1 0 1 }
INC write {  05  A x 0 1 0 1 }
INC write {  06  5 x 0 1 0 1 }
INC write {  07  A x 0 1 0 1 }
INC write {  08  5 x 0 1 0 1 }
INC write {  09  A x 0 1 0 1 }
INC write {  0A  5 x 0 1 0 1 }
INC write {  0B  A x 0 1 0 1 }
INC write {  FF  A x 0 1 0 1 }
; read checkerboard
INC read  {  00  F 5 0 1 1 0 }
INC read  {  01  F A 0 1 1 0 }
INC read  {  02  F 5 0 1 1 0 }
INC read  {  03  F A 0 1 1 0 }
INC read  {  04  F 5 0 1 1 0 }
INC read  {  05  F A 0 1 1 0 }
INC read  {  06  F 5 0 1 1 0 }
INC read  {  07  F A 0 1 1 0 }
INC read  {  08  F 5 0 1 1 0 }
INC read  {  09  F A 0 1 1 0 }
INC read  {  0A  F 5 0 1 1 0 }
HALT read {  FF  F A 0 1 1 0 }
```

Figure 10-1 Schlumberger ITS 9000 test vectors

Figure 10-1 is an example vector file for the Schlumberger ITS 9000 test system and shows a checkerboard pattern for the 256 x 4 static RAM. The first column represents the data for the address bus. The data is displayed in Hex format and increments from address 0 to address 255 (some vector statements have been removed for this example). The second column represents the input data. This data is also in Hex format—the data 5hex and Ahex will create a checkerboard pattern which will be stored in the RAM. The third column represents the output data and is also displayed in Hex. The output data is *"x"* during the write cycle. The character "x" indicates that the output is not tested. During the read cycle the characters 5hex and Ahex appear in the output field, which indicates that the outputs are tested. The fourth and fifth columns represent data for CS1 and $\overline{CS2}$. The sixth column represents data for $\overline{WE}$, with the last column for $\overline{OE}$. Notice the "read" and "write" pointers to timing sets. This vector pattern uses two time sets—the "write" time set contains the timing and format information need to write data into the RAM and the "read" time set contains the timing and format information need to read data from the RAM.

```
PATTERN Trillium_Example;
HEADER

{ signal names map to channel order }
%Xin20M_pd, Xin32K_pd, %StopClockb_pd, ClkBypass_pd, PwrDownb_pd,
Xout20M_pd, Xout32K_pd, %BAckb_pd, Runb_pd,
Exceptionb_pd, SysOutb_pd, GCLK80_pd,
%XEnb_pd, MemORdb_pd, MemOWrb_pd, %Wrb_pd, ALEb_pd;

{ format character definitions }
FORMAT 0/1 = DRON ,NRZ,NORM, ,BOTH, ,PAT ;
FORMAT Z/R = DRON , RZ,NORM, ,BOTH, ,PAT ;
FORMAT C/S = DRON ,SBC,NORM, ,BOTH, ,PAT ;
FORMAT r/o = DRON , RO,NORM, ,BOTH, ,PAT ;
FORMAT L/H = DROFF, , ,ACTIVE_EDGE,NORM,NORM,PAT ;
FORMAT x/X = DROFF, , , ,BOTH, ,PAT ;
FORMAT ?/# = DRON ,NRZ,NORM, ,BOTH, ,PAT ;

{ XX SCPXX BRESGC XMM WA }
{ ii tlwoo AuxyCL Eee rL }
{ nn okruu cncsLK nmm bE }
{ 23 pBDtt kbeOK8 bOO _b }
{ 02 Cyo23 b_pu80 _RW p_ }
{ MK lpw02 _ptt0_ pdr dp }
{ ___ oanMK pdib_p dbb d }
{ pp csb___ d o_pd ___ }
{ dd ks_pp npd pp }
{ b_pdd bd dd }
{ _pd _ }
{ pd p }
{ d d }

DATA_PATTERN Trillium_Example, 0, 0;
trill_start:
0X XXX0X XXXXXX XXX zz * 0
0X 11X0X XXXXXL XXX zz * 0 , rpt 3
1X 11X1X XXXXXL XXX zz * 0
0X 1100X HXXLLH XXX zz * 0
1X 1101X HXXLLL XXX zz * 0
0X 1100X HXXLLH XXX zz * 0
1X 1101X HXXLLL XXX zz * 0
0X 1100X HXXLLH XXX zz * 0
1X 1101X HXXLLL XXX zz * 0
0X 1110X HXXLLH XXX zz * 0
1X 1111X HXXLLL XXX zz * 0
0X 1110X HXXXHH XXX zz * 0
0X 1110X HXXLLH XXX XX * 0
1X 1111X HXXLLL XXX XX * 0
0X 1110X HLLHHH XXX XX * 0
0X 1110X HLLLLH XXX XX * 0
tril_stop::
1X 1111X HLLLLL HXH HH * 0
end;
```

Figure 10-2 Trillium Delta Master test vectors

Figure 10-2 shows a simple test vector pattern for the LTX Trillium Delta Master test system. Notice that the definition of each character used in the vector pattern is defined at the start of the file. This definition also contains the signal format (RZ, SBC, etc.).

```
SET MODULE 4095,SPM;
REM DYNAMIC IDD/IPP TEST VECTORS;
REM
ACVCSCVHHH HVVVVAAVVV VVVGGHHHHC HHVVVVRHHH HMMAVVVGCC
MOPOHOCCCC CIIIIGGNNP PDDNNSSSSO DREDRSOEEE E23GNPDNOO
PRPR0LLLLL LDDDDNNNNP PDDDDTTTTL IENIETWNNN NI NNPDDRR
BN N 0KKKK K0123DD AAAAE RSDRSAEDDD DD D NN
IE E 012 3 RRRRN E ERN012 3 EE
AR R TTTT T TT RR
SB T 0123 B B BT
L L RR
;
LSET DA* 1000001111 1000000000 0000011111 1101111000 0000000000;
LSET MA* 0000000000 0000000000 0000000000 0000000000 0000000000;
ENABLE DA,MA;
SET FC 25 1000001111 1000000000 0000000001 1001001000 0000000000;
SET F 1000001111 1000000000 0000000001 1101101000 0000000000;
SET FC 5 1000001111 1000000000 0000000001 1001001000 0000000000;
SET F 1000001111 1101000000 0000000001 1101101000 0000000000;
SET F 1000001111 1101000000 0000011111 1111111111 1000000000;
SET FC 158 1000001111 1101000000 0000000001 1101101000 0000000000;
SET F 1000001111 1101000000 0000000001 1101101111 1000000000;
SET FC 351 1000001111 1101000000 0000000001 1101101000 0000000000;
SET F 1000001111 1101000000 0000000001 1111101000 0000000000
LCALL CONTINSUB;
SET F 1000001111 1000000000 0000000001 1001001000 0000000000;
LSUBR CONTINSUB CONTIN;
ENABLE DA,MA;
SET F 1000001111 1101000000 0000011111 1111111111 1000000000;
SET FC 158 1000001111 1101000000 0000000001 1101101000 0000000000;
SET F 1000001111 1101000000 0000000001 1101101111 1000000000;
SET FC 351 1000001111 1101000000 0000000001 1101101000 0000000000;
SET F 1000001111 1101000000 0000000001 1111101000 0000000000;
SET F 1000001111 1000000000 0000000001 1001001000 0000000000
LEND;
END;
```

Figure 10-3 Sentry Series 20 test vectors

Figure 10-3 shows a simple test vector pattern for the Schlumberger Sentry Series 20 test system. Notice that all characters used in the vector pattern appear as either "0" or "1." This makes the vector pattern difficult to read. The definition of input drive and output compare states are defined in the "SET D" and "SET M" statements. Notice the use of the "LSUBR CONTTNSUB" statement. This statement defines a continuous subroutine which is used during a dynamic IDD test.

Simulation Data

Once the device is modeled and simulated, the results of the simulation can be saved as a tabular text file. The simulation results file can then be used for conversion to test data. The simulation data will contain all input/ output states and will generally have a time stamp value associated with each line of data.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  CYCLE TIME = 5000 UNITS                                      $
$                                                               $
$  Cycle based simulation capture file                          $
$                                                               $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
             vvvvvvvv vvvvvvvv vvvvvvv  bbbbbbbb bbbbbbbb  v t t t t t t t
             aaaaaaaa aaaaaaaa aaaaaaa  vvvvvvvv vvvvvvvv  c _ _ _ _ _ _ _
             22221111 11111198 7654321  dddddddd ddddddd   l c c u l m o o
             32109876 543210            11111198 76543210  k e a w w r u u
                                        543210               0 s r r e t t
                                                               0     s 1 1
                                                                       0 0
                                                                       0 1
       TIME
       4500  11111111 11111111 1111111  UUUUUUUU UUUUUUUU  X X X X X X U U
       9500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      14500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      19500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      24500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      29500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      34500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      39500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      44500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      49500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      54500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      59500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      64500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 0 1 1 1 1 H L
      69500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      74500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 0 1 1 1 1 H L
      79500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      84500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      89500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      94500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      99500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
```

Figure 10-4 Print On Strobe simulation data

Two methods are used in capturing simulation data. The print on strobe method (also called cycle based capture) records the states of all input and output pins at one specific point in time for each test cycle. This method results in one line of simulation data representing one test cycle. Since the data for each cycle is captured at a single point in time within the cycle, any signal transition which occurs at times other that the capture point will be lost. If all of the signal timing is in NRZ format no data will be lost because signals will only change at the start of each cycle. See Figure 10-4.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$  CYCLE TIME = 5000 UNITS                                              $
$                                                                       $
$  Event based simulation capture file                                  $
$                                                                       $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
             vvvvvvvv vvvvvvvv vvvvvvv  bbbbbbbb bbbbbbbb  v t t t t t t t
             aaaaaaaa aaaaaaaa aaaaaaa  vvvvvvvv vvvvvvvv  c _ _ _ _ _ _ _
             22221111 11111198 7654321  dddddddd dddddddd  l c c u l m o o
             32109876 543210            11111198 76543210  k e a w w r u u
                                        543210               0 s r r e t t
                                                               0     s 1 1
                                                                       0 0
                                                                       0 1
       TIME
          0  11111111 11111111 1111111  UUUUUUUU UUUUUUUU  X X X X X X U U
       5000  11111111 11111111 1111111  UUUUUUUU UUUUUUUU  1 X X X X 0 U U
       5272  11111111 11111111 1111111  UUUUUUUU UUUUUUUU  1 X X X X 0 X U
       5373  11111111 11111111 1111111  UUUUUUUU UUUUUUUU  1 X X X X 0 H U
       5444  11111111 11111111 1111111  UUUUXXUX UUUUUUUU  1 X X X X 0 H U
       5445  11111111 11111111 1111111  XXUUXXXX UUUUUUUU  1 X X X X 0 H U
       5446  11111111 11111111 1111111  XXXXXXXX UUUUUUUU  1 X X X X 0 H U
       5448  11111111 11111111 1111111  XXXXXXXX XXXXUUUU  1 X X X X 0 H U
       5454  11111111 11111111 1111111  XXXXXXXX XXXXXXXU  1 X X X X 0 H U
       5455  11111111 11111111 1111111  XXXXXXXX XXXXXXXX  1 X X X X 0 H U
       5500  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H U
       5597  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H X
       5715  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      10000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      15000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      20000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      25000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 0 H H
      30000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 0 H H
      35000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      40000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      45000  11111111 11111111 1111111  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      50000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 1 1 1 1 1 H H
      55000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 1 1 1 1 1 H H
      55500  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H H
      55636  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H Z
      55878  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      60000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 0 1 1 1 1 H L
      65000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      70000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 0 1 1 1 1 H L
      75000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  1 0 1 1 1 1 H L
      80000  00000000 00000000 0000000  ZZZZZZZZ ZZZZZZZZ  0 0 1 1 1 1 H L
```

Figure 10-5 Print On Change simulation data

The print on change or event based capture method produces one line of simulation data when any signal makes a transition. Since all signal activity is captured, this method can result in many lines of simulation data representing one test cycle. The print on change method of capturing device activity during simulation contains valuable information but can result in huge simulation files. See Figure 10-5

There are many important issues involved in using and converting simulation data to test data. Read over the following Simulation for Test guidelines for some helpful hints.

Simulation for Test

This Simulation for Test section may be copied and distributed

Test patterns are often generated by converting the output from logic simulations to the format required by the target test system. The purpose here is to provide guidelines to be used by any engineer running simulations for the purpose of converting simulation output to test vectors for a chosen test system.

Simulation Guidelines:

1. Make certain that the simulation begins by performing a complete initialization of the device. This initialization should not be based on the "Power Up" state of the device. The test vectors that result from simulation data should produce consistent results when executed repeatedly. Each simulation file must be a stand alone module, i.e. it must not require other files to execute successfully on the DUT.
2. The device model should be initialized through normal application type functions such as "Reset" when possible. For example, if an internal flip flop exists which can not be reset or initialized, the simulation output may appear correct and the design will work in the application. However, the test pattern may be intermittent depending on the state the flip flop powers up in each time the device is tested.
3. Internal and external nodes should never be forced to a predetermined state. Node states must be achieved through pattern initialization.
4. Outputs should NOT be tested until the device has been fully initialized, even if certain outputs are in a known state.
5. Use of 'Print_on_change Mode" for all external signals is recommended. This insures that all transitions are captured and will help in the post-processing.
6. Provide a means of identifying the active function of input/output or tri-state signals. This may be a signal internal to the device such as the "Enable" signal for the tri-state buffer. This signal can be used to clearly define when the test system drives, compares or ignores data.
7. Frequency and timing conditions should be identical for both simulation and test. Best and worst case simulations should be performed at the device test frequency. Results should be compared to insure that the output data does not cross cycle boundaries.
8. When a device requires multiple clocks, the clocks should share the same frequency or be multiples of each other. For example, testers can handle a clock of 4MHz and another of 8MHz, but may not perform correctly with a 4MHz clock and another of 7MHz.
9. Although it is possible to run simulations in an asynchronous manner, most digital test systems are synchronous machines. Simulation wave forms must be broken down into cycles during post processing, therefore try to group simulation stimulus into cycles.
10. When possible, avoid timing set requirements (timing changes on the fly) and reference timings for input signals to the device input docks.
11. Before printing simulation output data in HEX format verify that this will not cause a problem. Some conversion software will not handle hex and additional work may be required to reformat HEX into binary.
12. When AC parametric characterization is required to verify design specifications, the simulation timing should be organized in a manner similar to the AC electrical specifications. For example, when performing a setup time measurement, the signal being measured and the reference signal should both transition in the same cycle.

13. It is very useful to develop test patterns specifically for testing DC parameters. For example, if a dynamic IDD test is to be performed, a pattern which conditions the device to a specific state should be generated. This pattern may then be used to exercise the device dynamically while measuring the IDD current. Other DC patterns may include setup conditioning for:
    - VOL/VOH—all outputs should achieve both zero and one states. This test may be statically or dynamically performed.
    - VIL/VIH—all inputs should be exercised high and low.
    - IDD static and dynamic—condition the device as required.
    - Tri-state, functional or parametric—all bi-direction and tri-state outputs should achieve a high impedance state.

14. Timing diagrams, representing the input and output timings used in the simulation, are essential.
15. If you suspect that there may be problems with certain areas of the simulation, be sure to communicate your concerns to the test engineer (it could save a great deal of debug time).

    If possible, meet with the test engineer before creating the simulation. If it's too late for that, meet with the test engineer to discuss the simulation data.

    Be sure to document the meaning of all characters that appear in the simulation file.

Test Vector Development Review

1. When should the test engineer meet with the design engineer to discuss test vector development?

2. The majority of test vectors are currently being developed from simulation data.
   a) True
   b) False

3. Which method of simulation results in one line of simulation data representing one test cycle?

4. Where are test vector patterns stored within the test system?

5. Does the number of test vectors (test cycles) affect the total test time?
   a) Yes
   b) No

6. What is the purpose of a sequence of test vectors?

Chapter 11 Test Program Development Issues

Objectives:

This section explains:

- The various types of test programs
- Issues to consider before developing a test program
- Considerations for developing the program flow
- Power up/ power down sequencing
- Binning
- Test summary reports

What is the Primary Purpose of the Test Program?

Engineering

The initial test program is often an engineering tool used to verify the functionality of the device and maybe to gather information about it. The program should be structured to allow easy modification of voltages, currents and timings. It may also have routines to allow quick debugging and may contain some characterization routines.

Characterization

The characterization program is developed to determine the operational limits of the device. It is often an extension of the engineering test program. The characterization program should take advantage of routines that are included with the tester operating software. Standard routines may include Shmoo plots, level search and timing search routines.

Production

The main purpose of the production test program is to separate the bad devices from the good. The program may be used for wafer sort, final test or QA test (quality assurance). It should execute as fast as possible. The flexibility of the engineering program is traded for the faster test time of the production program.

Other Considerations

Hardware Limitations

Once the device specification has been carefully reviewed and a detailed test timing diagram has been developed, make certain the test hardware is capable of meeting the test requirements. Take into account the test rate (maximum frequency), as well as voltage and current requirements for inputs, outputs and VDD. Do not forget wafer probers, device handlers and other external equipment that may be required.

Throughput

Estimate the device test time on the target test system. Is the test time satisfactory? If the device goes into high volume production will the test throughput be adequate? If not, an alternate test system may be a better choice.

System Availability

Will the target test system be available when needed. Will the support equipment that interfaces to the target tester be available when needed?

Test Costs vs. DUT Cost

The cost of testing the device must be considered when selecting the target test system, it may not be cost effective to test a low cost device on a very expense test system.

Initializing the Program

The opening menu allows the operator to select the test options to be used during testing. Some of the standard options are: Wafer Sort, Final Test, QA Test, Characterization, temperature selection, and possibly the selection of various versions or options of the device. Additional information may also be collected, for example: Lot Number, Operator ID, Test system number. The opening menu may also provide the operator with instructions for setting up the test hardware.

Verifying the Test Setup

A loadboard test and a quick diagnostic must always be executed before testing devices. The program should be designed so that testing can not begin until the loadboard and diagnostic tests pass. The ability to rerun the loadboard test at any given time is also useful.

Power-on Sequencing

Care must be taken when powering up the device. CMOS devices may latch up if any voltage exceeds VDD or ground by more than 0.5V. Latch up can cause excessive currents to flow within the device and may result in the destruction of the device under test. A safe procedure for powering up the device is to first apply zero volts to all power and input pins (make sure the pin electronic drivers are turned on for the inputs). Power up the VDD supply, then program the VIL/VIH supplies. Power up any external circuitry or loads last. See Figure 11-1.

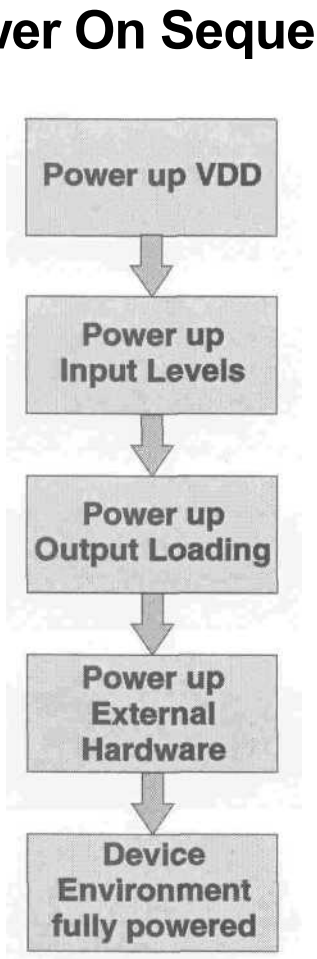


Figure 11-1

Power-off Sequencing

After the binning information has been completed, the device and the test system must be prepared for power-down. The device must be powered down in sequence—first any output loading or external hardware must be disconnected. The inputs may then be powered down and last the VDD supply is shut off (set to 0V then disconnected). The test system drivers should also be shut off at the end of test. See Figure 11-2.

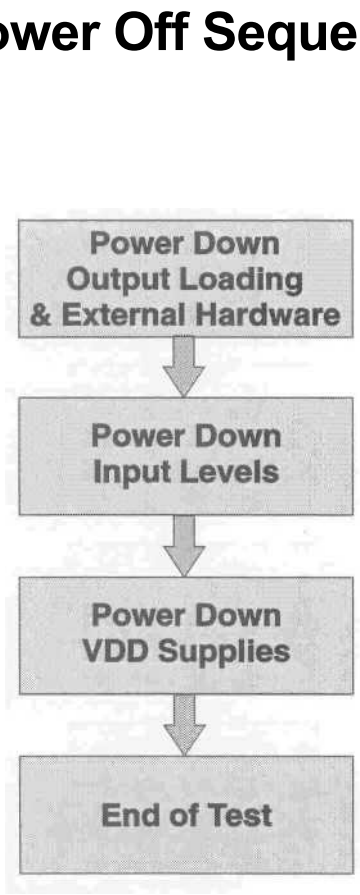


Figure 11-2

Test Program Development Review

1. What are the requirements for an engineering test program?

2. What is the purpose of the production test program and how does it differ from the engineering program?

3. What test system routines are likely to be used within a characterization program?

4. There are many issues which should be considered before test program development begins, name at least three:

5. Why is a loadboard test important?

6. If the power-up or power-down sequence is performed incorrectly, what may occur?

7. During production testing, what action is normally taken when a failure occurs?

   a) Testing stops
   b) The test summary is updated
   c) A fail bin is selected
   d) All of the above
   e) None of the above

Chapter 12 Creating a Test Program

Objectives:

This section explains;

- How to begin test program development
- Loadboard tests and hardware diagnostics
- How to debug a new test program

Developing the Test Plan from the Device Specification

Before you begin to develop a test program the entire device specification should be reviewed in detail. As you review the specification make a list of all the individual tests and include the exact conditions for each test. Make certain you understand all specifications, including a detailed understanding of the device timing. Be sure you have the current information—last minute changes to the specification (or device pin out) can ruin the best made plans. If possible, meet with the designer to clear up any uncertainties that you may still have. The following information may provide some helpful hints.

Take Notes!

While reviewing device specifications it is helpful to write down your ideas and concerns, and to make note of items which may need special consideration- Get a notebook or file folder to help organize all of the information available.

Power and Ground

Review the voltage and current requirements for the device under test, High current devices may require hardware with dedicated power and ground planes. Device pins that require high VDD voltages may need to be isolated from the tester driver and comparator circuitry.

Some test systems provide a driven or buffered ground, while others have only a hard ground; some test systems have both. The type of ground used for any given device should be taken into consideration in the early stages of program planning. Some test systems provide sense lines to monitor power and ground. If this option is available, proper connections must be made on the test hardware to insure accurate voltages at the DUT. Using capacitors or other means to reduce power supply noise caused by the DUT (decoupling the supplies) must also be considered.

Special Considerations

While reviewing the device specification make note of any conditions that will require special considerations. Open drain outputs, for example, will require some type of pull-up. A device with a high IDD current may require extra VDD or Ground paths. Some devices may require a high voltage signal to put the device into a special test mode or to program an EEPROM.

Notes and Exceptions

Exceptions or special notes may be associated with a particular parameter—be sure to read the details. It may be stated that the parameter is guaranteed by design and does not require testing. Other parameters may need to be relaxed during testing due to fixture noise. Some device specifications may define the amount by which a given parameter may be relaxed. Additional information may also be given which applies to specific tests.

Functional Test Timing

When developing a functional test all of the timing defined in the test specification should be carefully reviewed, then a test riming diagram should be developed for use within the test program. Consideration must be given to the speed, accuracy and number of resources available for the target test system. It is also useful to review the tester specification to insure that you understand the capabilities and limitations for the target test system.

It may not be possible to exercise all of the various input timing edges and signal formats or to test all of the AC parameters at one time. Develop a basic timing strategy that will work well for a gross functional test. Once the basic timing is developed you can then begin to add complexity to your timing diagram.

The first step in developing the test timing is to define the frequency (test rate) and placement of the clock and control signals. Next, determine the active edges of the clock or control signals, when input signals are read (latched), when output signals are gated out. Look closely at the delay time parameters for output signals. Make certain that outputs have sufficient time to propagate out before the end of the test cycle. Spend as much time as necessary to fully understand the device timing. This is one of the most important steps in test program development!

If the test vectors are being developed from simulation data, review the timing used during simulation. The test program will need to duplicate the simulation timing. The simulation timing should reflect the device specification timing and must also be compatible with the performance of the test system.

Designing the Test Hardware

If time allows, design and build only the test hardware for hand testing the device. In the event that extensive hardware changes are required during the initial device/program debug, it will be unnecessary to alter the probe card and interface hardware if they are not yet built.

Always attempt to minimize the complexity of the test hardware by keeping the number of external components and wiring to a minimum. Remember, hardware has a way of self-destructing when used in a production environment. It is up to the test engineer to use as much of the internal tester circuitry as possible to accomplish the task of testing the device.

Review the notes you have been collecting—does the device require special hardware considerations? If so now is the time to work out those details.

Writing the Test Program

Begin by entering the device pin to tester mapping. This should include the device pin, signal name, function (input, output, I/O, power) and the tester pin associated with each signal.

Once the definition of each individual pin has been entered, pin groups can be defined. Pin groups are created based on similar characteristics, timings, voltages, currents etc. Once entered, be sure to check your work, this information will be used again and again throughout the development of the test program.

Develop the Opens and Shorts and Gross IDD current tests next, then the Gross Functional test. At this point you have enough to get started. If time allows, add the input leakage test.

Loadboard Tests

Develop a complete loadboard test before you begin testing. The loadboard test must include a leakage test of all pins, a test of VDD and ground pins, and all external relays, resistors or other external components. A quick tester diagnostic must also be included as part of the loadboard test. See Figure 12-1.

Loadboard Test Flow Diagram

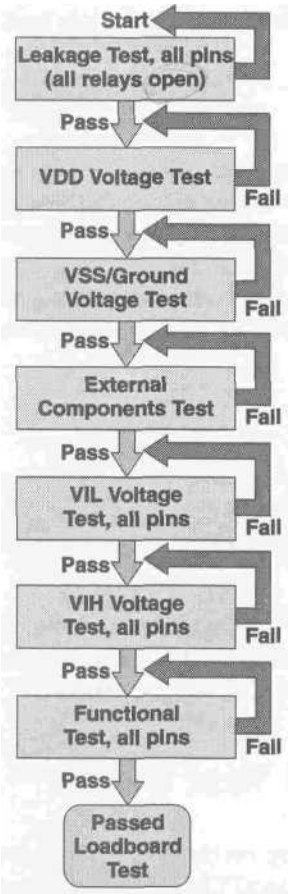


Figure 12-1

Tester Diagnostics

It is important to include a quick tester diagnostic as part of the loadboard test. A basic diagnostic can consist of forcing the input reference low supply to 0.8V and the input reference high supply to 2.0V, then test all pins with the DC measurement system.

A fast functional test of all pins is also very useful. The driver and comparator circuitry for each pin can be tied together to perform a self-test as an easy way to functionally verify that all pins are working correctly.

Running the Program the First Time

Hardware Verification

Once the test hardware is built and the loadboard test is written, the test hardware can be verified. Make certain that the complete loadboard test is working correctly and that the test system passed the hardware diagnostic. Do not attempt to verify the device test program until the hardware and test system are fully functional.

Opens and shorts

After the loadboard and test system have been verified, turn on the datalogger and run the opens and shorts test with an open socket. Make certain that all pins fail the opens conditions. Next, insert the device into the test socket and again run the opens and shorts test, but stop the program immediately after the opens and shorts test. Review the datalog measurements to make sure the readings are well within the pass/fail limits. If necessary adjust the limits as needed.

Gross Current Tests

Run the program again and this time stop immediately after the gross current test and datalog the measured value of the gross current test. If the positive supply (VDD) is measured the current must be positive. Does the reading make sense? Example: If the device specification for IDD is 5mA and the measured value is 48mA something is wrong.

The significance of this test, at this time, is to signal some type of gross problem. If the measured current appears incorrect, remove the device from the test socket and run the test again. The empty socket reading should be 0.0mA; if it is not, investigate and solve the problem.

Verifying the Functional Test Setup

Before functionally testing the device, loop on the test pattern and, using an oscilloscope, observe each tester channel. Are the timing and voltages correct? Check VDD and ground—do they look correct? Be sure to look at every signal (including the output strobe, if possible). If pull-up resistors or current loads are required, make sure they are connected and working correctly.

Basic Functionality

Running the gross functional test is when the fun really begins. Turn on the functional datalogger, press the start button, sit back and watch the results. What happened? If nothing failed there is probably trouble.

Remember that the lack of a failure does not guarantee that the test is being performed correctly. Remove the device from the test socket and run the test again. Did it fail? Review the failures and make sure that everything that should fail did fail. If the test still does not fail, review the test program—something is clearly wrong.

Chances are that the test failed the first time you ran it. If so, review the datalog to look for clues—are all the failures somehow the same, e.g. only logic 1 failures or only logic 0 failures? Possibly a certain bus is the only failure. See if any of the test pattern is working. Does the pattern fail at the very beginning? Maybe the vector pattern has an initialization problem or the output strobe is wrong. If the functional test is made up of multiple patterns, run all of the patterns to collect as much data as possible. Datalog the results to disk or paper because you will need to spend time off line to analyze the failures.

Leakage Tests

Run the leakage test, datalog the results and make sure the readings are correct. If the functional test failed and the leakage test failed, investigate the leakage test. Solving the leakage failure may improve the functional test results.

A Brief Discussion on Test Vectors

Test vector patterns combined with timing values and signal formats are the heart of the test program. Vector patterns are often developed by the design engineer during functional simulation of the device. They must then be converted to test vector format for use in the test program. If this is the case and the test vectors fail the functional test, be sure to collect as much information as possible (datalogs) and review the failures with the design engineer. Give the design engineer a copy of the *Simulation for Test* guidelines, found in chapter 10. Explain exactly what the program is doing and be sure the timing used during simulation is the same as the timing used in the test program.

Creating a Test Program—Review

1. What is the very first test that should be run when debugging a new test program?
   a) The opens and shorts test
   b) The loadboard test
   c) The gross IDD test
   d) The gross functional test
   e) None of the above

2. What is the purpose of the Gross IDD current test?

3. What steps should be taken to verify the functional test setup before testing the first device?

4. When executing functional tests for the very first time, what should be done if the test result is a pass?
   a) Go out for beer and pizza
   b) Remove the device from the socket and run the test again
   c) Test another device
   d) None of the above

5. If the functional test fails and the input leakage test also fails, which test should be debugged first?
   a) Functional
   b) Input leakage

Chapter 13 Troubleshooting

Objectives

Upon completion of this section you will understand:

- How to find where problems occur
- How to look for dues in the test results and isolate the problem
- The proper sequence for debugging
- What test system utilities are available
- The function of the various system utilities
- How to use tester tools for debugging

Introduction

The *Verifying DC Parameters* section has information on trouble-shooting each specific DC test. But it does not give an overall approach to finding a test problem, especially on a production floor where "down time" is very expensive. When you consider all the possibilities for a testing problem, it can be difficult to choose a starting point. To simplify the problem, begin by narrowing the possibilities to four main suspects. When a failure occurs during test, the cause may be:

- The Device Under Test
- The Interface Hardware (loadboard, test socket, pogo pin)
- The Test System
- The Test Program

While trouble-shooting, the exact cause of the failure may be identified, or if three of the four suspects are eliminated, then the last remaining possibility must be the cause; either way the problem will be found.

Although there are many approaches to trouble-shooting, the first step is to perform the test with the datalog utility enabled to allow the exact measured value to be observed and evaluated. Consider the test method in use when the test is performed. Also, compare a failing measurement with a passing value. Determine if the measured result is a limit (marginal) failure or a catastrophic failure.

If the failure appears to be catastrophic, try to determine the cause. First, consider the test method. Then notice if the measured result was produced by the PMU clamp circuitry. Or the measured result may be the maximum value of a PMU measurement range. The measured results will contain clues to the riddle—it is up to you to use the clues to solve it.

Once the results have been analyzed you should have an idea of what to do next. It is generally best to start by doing the easiest thing first that will result in gaining useful information. Keep in mind there will be many things to try, but first do the easiest thing that will give useful information. In many cases removing the device from the test socket will provide valuable information. Generally this is the easiest thing to do, but be aware that it will not always yield valuable information.

Where to Begin

Your starting point will be influenced by several factors. Among those will be:

- What test is failing?
- What is the history of the test program and device?
- What is known about the problem at that point in time?

First, find out what's failing, enable the datalog utility and look closely at the test results. Determine which test is failing—is it a DC fail, an AC fail or a functional fail? Notice the number of failures—is only one test failing, possibly one pin failing?

Look for basic clues. If the problem is not apparent, proceed methodically and *remember to solve one problem at a time.*

Verify the Test Hardware

Run the loadboard test if the program has one (the program *should* have one). Continue only if the test hardware is performing to specification. It may also be helpful to remove and reinstall the loadboard.

Test a Standard Device

Get a known good device and test it and note the differences in the measured results. Determine if the failing product has been successfully tested on this test system before.

Try to Verify the Basics

Make sure that both the test program and the failing device are the correct revisions. Make sure that the device is placed in the test socket correctly. Make certain that any data required by the test program is entered correctly.

Once you have determined:

- the diagnostics are passing
- the standard device fails
- the program is the correct revision
- the devices are the correct revision

Pick one test and find out why it is failing. Start by getting an oscilloscope and a DVM. Pause on the failing test and verify that VDD and the input levels are correct. Verify that all timings are correct. Be sure to look at every device pin with the oscilloscope. Look for noise or levels that appear to be incorrect.

Use Relaxed Values

Relax the test parameters to make it easier for the device to pass. Relax VIL/VIH/VOL/VOH and test the device again. Relax the timing parameters. Test the device again and see if the results change.

Try Another Test System

If possible, test the device on a different test system. If the results are the same, it is not a test system problem. If the device passes, contact the test maintenance department and explain the problem seen on the first test system.

Make Certain Opens and Shorts is Passing

The device will not function correctly if the opens and shorts test is failing. If a number of tests are failing, *get the opens and shorts test to work first.* Once you have the opens and shorts test passing, look into any input leakage failures.

Is it a Tester Problem?

If another test system is not available for use and all of the above suggestions have failed to provide a solution, it is time to contact the test maintenance department and ask to have a complete tester diagnostic run. Do this only after you have tried all of the items above.

Collect and Review the Datalog Results

A good way to start the investigation is to datalog the test results to a printer or file so they can be studied. Don't waste expensive tester time viewing results on the video display if you can do it off-line. Datalog both pass and fail measurements for all tests.

Analyze the Results of All the Tests

Look carefully at the results of all tests, both passing and failing. There will often be helpful clues in the datalog information. Notice if the device is drawing the correct amount of IDD current. Also, check to see if the input leakage tests are failing. See if the VOL/VOH voltages look correct. If the failure is a DC measurement, is it a limit failure or a catastrophic failure?

Has this Failure Ever Passed Before?

Verify that the failing test has worked in the past. Are there datalogs showing the results of a successful test? If so, compare the results of all tests.

Limit Your Debug Time

It is a good idea to limit the number of hours spent debugging one problem. This is an individual choice, but a good rule of thumb is *do not spend more than two or three hours working on the test system on a single problem.* If you work more that several hours on one problem, stop and take a break, discuss the issue with a colleague.

Double Check All Your Work

If failures occur when the test program is initially being debugged, collect as much data as possible and then go back to your desk and review all of your work. It is often helpful to have a colleague assist in reviewing your work—discussing the problem with another engineer may help you find a solution more quickly.

Example: IIL/IIH Test Failures

Let's say that the input leakage test (IIL/IIH) is failing This test forces voltage and measures current, therefore the failure will be due to excessive current flow. It is possible that the device under test may be defective and the inputs are simply consuming too much current. It is also possible that there may be a problem with the interface hardware, the test system or the test program.

The first step is to perform the test with the datalog utility enabled. This will allow the exact measured value to be observed and evaluated. Does the failure appear to be a limit failure or is it a catastrophic failure?

Step 1 - Eliminate the DUT

Since the test is failing due to excessive current it makes sense to remove the device under test from the test socket and repeat the test with an open socket. After the test is repeated, evaluate the test results by examining the datalogged measurement. If the current is now zero the test will pass, indicating that the device under test is most likely defective.

If however, the datalogged measurement indicates that current is still flowing when the test is performed with an open socket, then the problem is most likely the interface hardware or the test system.

Step 2 - Eliminate the Interface Hardware

If current still flows with no device in the socket, the next step will be to remove the interface hardware. The interface hardware consists of whatever is between the pin electronics and the DUT. This will include the loadboard and socket when performing hand test. The probe card is included at wafer sort and the interface to the handler is included if an automatic handler is in use.

Once the interface hardware is removed, the test should be repeated and again the measured results observed. If the datalog indicates zero current flow then the interface hardware most likely has a problem. If the datalog indicates that current is still flowing then the test system hardware may have a problem.

Step 3 - Eliminate the Test System Hardware

If current still flows with the interface hardware removed, the test system is the likely suspect. If possible, try the test program on a different test system. If no other test system is available, then it is probably time to run the test system diagnostics or call the maintenance department for help. If the test system has a problem then the diagnostics should identify the hardware failure.

Step 4 - Eliminate the Test Program

So you have heard that software can't break, right? What is the point of reloading the test program? There are several reasons why you may want to do this:

1. The test program and test vector data is stored in memory inside of the test system. It is possible for the test information to become corrupt during testing due to electrical power noise, a loose electrical connection, or some other unforeseen problem. Once the data is corrupt, the problem can only be corrected by reloading the test software.

2. Some test programs may present a series of questions that must be answered when the program is ini-
tially loaded. These questions may include selections for the type of test to be performed such as Wafer Sort, Package Test, or Auto-handler Test. Each option may require unique tester-to-device pin

connections. The program may request information regarding the type of device package such as a 64 pin package or an 84 pin package. The program may also request information regarding device options that may be available, such as on board ROM or RAM. When you reload the test program make certain that all of your answers are correct.

When reloading the test program make certain that you are using the correct test program name and the correct revision. It is easy to enter the program name as L17297B when the correct program is L17279B.

Debug Tools

Scope Tool

Most test systems have some method of looping on a functional test vector pattern and providing a means of creating a sync marker at a selected vector memory location. The test pattern can be started and stopped at any point within the vector set and failures can be ignored. The oscilloscope can be used to verify the proper timing and voltage levels when verifying a new test program or for investigating test problems. It is absolutely essential to become proficient at using this debug technique.

Shmoo Plot

Shmoo plots are useful for debugging test problems and for finding the pass/fail limits of device operation. Shmoo plots can be executed from the tester keyboard or from within the test program. This tool is very useful in developing device characterization programs and collecting data for both DC and functional tests.

Search

Search performs either a binary or linear search for both DC and Functional tests. This tool provides a means to take a quick look at input and output thresholds, or it can be used to provide accurate measurements of device timings. It is also very useful in developing device characterization programs.

Manipulating Tester Resources

Tester resource settings can be read and modified directly from the workstation keyboard. Power supplies, input references, comparator values, timings and functional data can all be manipulated. To debug a test problem, pause on the failing test and modify the appropriate parameters. The "Execute Button" can be used to verify the results of the modification. An oscilloscope can be used to verify pass/fail results. Become familiar with all of the tester tools—they will greatly enhance your ability to debug and resolve problems quickly.

Viewing Test Vectors

Each test system offers a way of viewing the vector patterns stored in memory. These utilities are essential for debugging test vector problems. They are also useful when a partial vector file listing is needed. Test vectors can be modified from the keyboard and the results tested.

Debug and Troubleshooting Review

1. How do you begin to debug a failure?

2. If all the devices being tested appear to have the same failure you should:
   a) Discuss the problem with a colleague
   b) Get a known good (standard) device and test it
   c) Call maintenance
   d) None of the above

3. If every test fails, what test should you debug first?

4. If the VIL/VIH functional test fails and all other tests pass what is the most likely problem?

5. When debugging a test problem is it important to understand the entire test program flow.
   a) Yes
   b) No

6. Explain why you selected yes/no in the previous question:

7. If the functional test is failing what could be changed in an attempt to make it pass?
   a) Relax the input levels
   b) Relax the output levels
   c) Relax the timing
   d) All of the above

8. What is the scope tool used for?

9. The search utility will generally provide two methods of performing a search. What are they?

10. Shmoo plots provide valuable information but they may require considerable time to execute.

    a) True
    b) False

11. Learning to manipulate the test hardware will greatly increase your productivity when developing or debugging test programs.

    a) True
    b) False

12. Most test systems provide a means of modifying test vector data stored in vector memory directly from the keyboard.

    a) True
    b) False

Chapter 14 Qualifying and Documenting the Test Program

Objectives

This section explains:

- How to qualify a test program
- How to create a test documentation package

Qualification

Sample Device Requirements

It is essential to serialize and save sample devices once test program generation is complete. If possible the test engineer should keep several samples separate from the production samples. Complete datalogs (with serial numbers) should also be kept for each sample device. The following items should also be noted on the datalog: test program revision, test system used and date tested. In the event that test problems arise, this information will be extremely valuable.

Checking the Margins

Margins refers to the amount by which a parameter can be varied before a failure occurs. Once the test program has been completed the margins should be verified. Use system utilities such as VLS and Shmoo when possible. Some test systems offer a utility designed to vary all parameters +/- a certain percentage. If a parameter is very close to failing, it will probably become a production yield issue. The idea is to identify the problem and resolve it before production testing begins. All voltages, currents and timings should be varied. Save this information for later reference.

Shmoo the Test Vector Patterns

If possible, Shmoo plot each test pattern to look for holes in the plots or other irregularities. Plot VDD vs. VIL/ VIH and VDD vs. output delays. Save this information for later reference.

Verify Binning and Summary

The binning and summary routines must be verified. A failure of each test type must be generated, then verify that the failure results in the proper binning. Keep track of failures, take a test summary and make sure the summary is correct.

Temperature Testing

Test the device over its specified temperature range. If it performs well, extend the temperature range by 10 percent and repeat the test. It is best to find out early if any potential problems exist. Record and save this information for later reference.

Verify testing on handlers and probers

If the device is to be tested at wafer test or with an auto-handler, be sure to set up the external hardware and verify correct operation.

Documenting the Test Program

Documentation

Once the program is completed a documentation package must be prepared. It is best to keep the information organized—a three ring binder works well for this. Begin by summarizing all of the test development activity. Make certain to note any deviations from the test specification and reference any supportive documentation such as Shmoo plots, datalogs, memos, etc. Include the test time and detail any special requirements such as external hardware, non-standard auto-handling, etc. All correspondence between the test engineer, design engineer, product engineer, the customer, etc. should also be included.

Program Listings

Include a listing of the program source code exactly as the program was initially released. It is also useful to save a compiled listing. A tape or diskette copy is the most convenient but a paper copy can be viewed anywhere.

Program Flow Chart

Include a program flow chart of all tests and show binning results of each test.

Datalogs

Include complete serialized datalogs showing the results of all tests. Also datalog an open socket and show the results of all tests. Review the open socket results to make sure each test fails. Check leakage and current readings too.

Test Specification

Include a copy of the test specification used for program development.

Hardware Documentation

Complete and thorough hardware documentation is essential. Include both drawings and a written description of all test hardware. This documentation must include enough information to build additional sets of hardware without the need to see the original hardware.

Standard Devices

A minimum of five standard devices must be included in the documentation package. These devices must be serialized and datalogged.

Backups and Archives

The test program must be saved on magnetic tape or diskette or other means of backup. Make several copies of the final software, one for the documentation package and one for your own files. It is also a good idea to save a copy of all files associated with the development of the program. Keep all simulation files, test cases, etc. This information can be compressed and saved on tape or diskette. Be sure to clearly mark each tape—the data on the tape must be easily identifiable.

Qualifying and Documenting the Test Program Review

1. Why is it essential to save sample devices?

2. To what does the term margins refer?

3. If a test vector pattern is working correctly, a Shmoo plot of the VDD vs. VIL will show no "holes."
   a) True
   b) False
4. List at least four items which must be included in a test program documentation package:

Chapter 15 CMOS Device Latch-up

-Advanced Topic-

by Vernon Rubright

Objectives:

This section explains:

- The cause of latch-up in CMOS devices
- The effects of latch-up
- Ways to prevent latch-up
- How to test devices for latch-up

What is Latch-up?

Complementary MOS (CMOS) evolved as a result of circuit design innovation using the combination of opposite polarity MOSFETs (P- and N-channel). This introduced the use of p- or n-wells to isolate the transistor types. These wells along with the desired MOS transistors form a (parasitic) four-layer silicon device (NPNP) commonly known as a Silicon Controlled Rectifier. Figure 15-1 illustrates how this structure is formed and shows the SCR as two lateral bipolar transistors in a regenerative configuration with the collector of one feeding the base of the other. The emitter of the PNP is the anode and the emitter of the NPN is the cathode. The bases function as complimentary gates.

The electrical characteristics of the SCR are shown in Figure 15-2. With a positive voltage applied to the anode, the device blocks the flow of current until the forward breakover voltage (VBO) is reached. At this point the device switches into a high conduction state and the voltage across the device drops to about 1V. In the high conduction state, current flow is limited only by the external circuit impedance and supply voltage. At voltages below the breakover voltage, the device can be switched into the high conduction mode with a small gate current. Once the SCR is in the high conduction *latch-up* state, it can only be switched off by reducing the anode current below the holding current (IH).

Parasitic SCR in CMOS Device

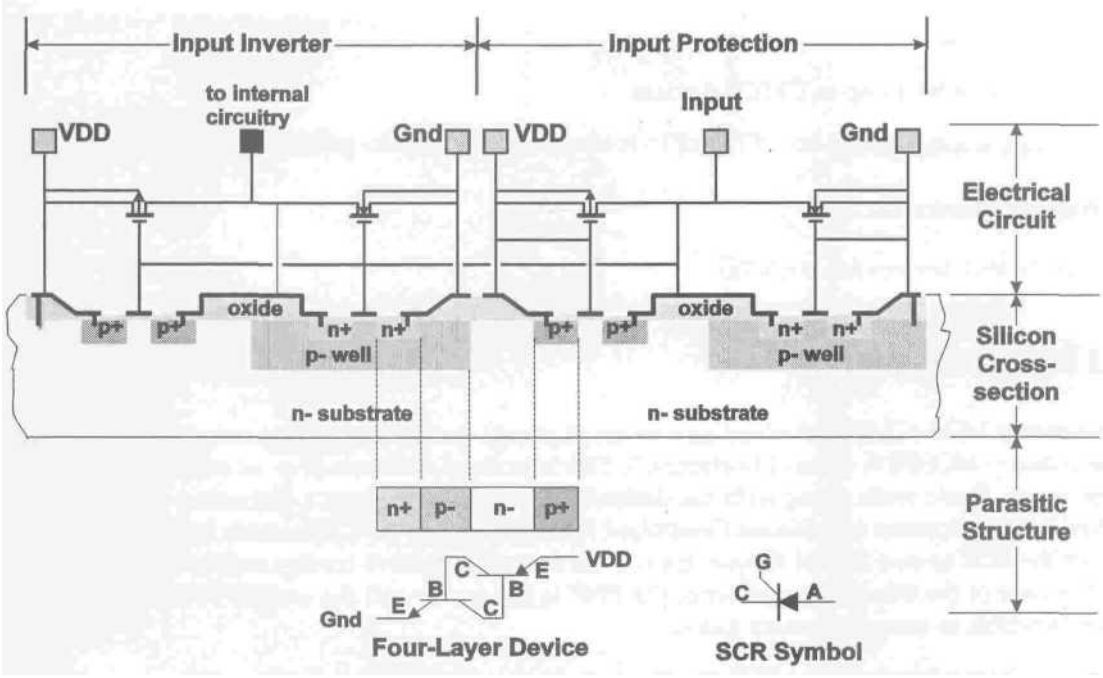


Figure 15-1

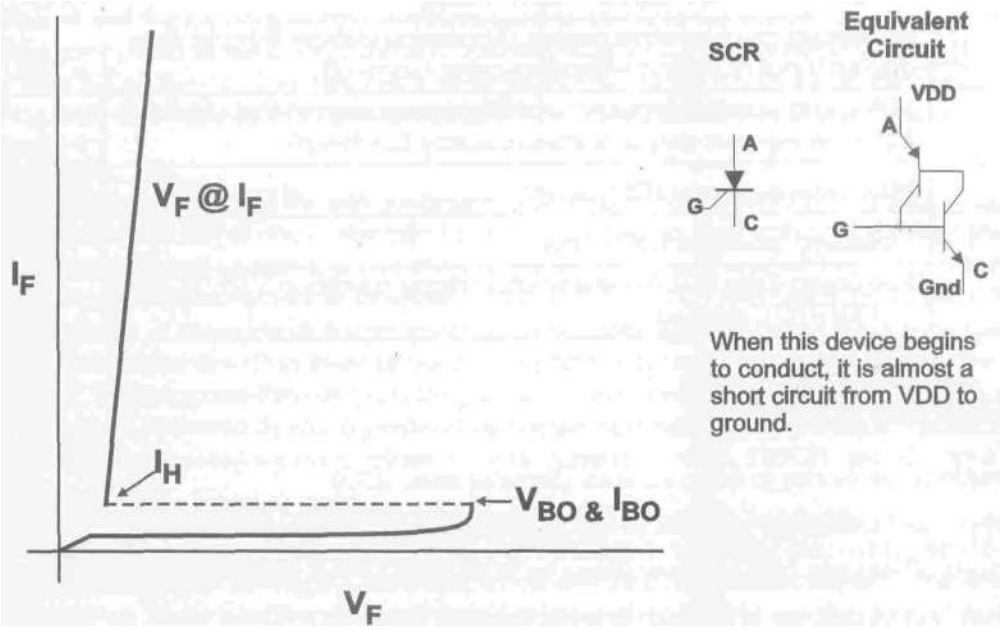
Since the gates of the SCR are connected to the substrate and isolation well, current flow in these areas can cause the SCR to enter the high conduction state. The most common sources of these currents are:

1. Input or output voltage swings that extend beyond the power rails (VDD/GND) and cause current to flow.
2. Improper sequencing of multiple VDD supplies.

Internal I*R voltage drops in the power rails resulting from large output drive currents.

SCR Electrical Characteristics

Figure 15-2



Latch-up can be prevented by controlling the electrical parameters of the parasitic SCR to guarantee the high conduction state can never be entered while the device is operating within the absolute ratings. Parasitic SCR parameters are therefore just like any other electrical parameter and must be thoroughly characterized to verify the fabrication design goals.

Latch-up Testing

The philosophy behind the Latch-up test is one of attempting to force the parasitic SCR into the high conduction state. This will occur if the VBO voltage is reached or enough gate current is supplied through the substrate or isolation wells to cause the SCR to trigger. VBO triggering is tested by applying elevated voltages to VDD and all input pins. Gate triggering is tested by forcing currents into and out of all input and output pins. The magnitude of these voltages and currents is derived from the fabrication specifications. Review the test methodology with the circuit design engineer and/or process engineer to establish test conditions that will verify the design goals.

Parameter	Definition	Typical value
VDD_{MAX}	Maximum operating VDD	5.25V
VDDOVR	Maximum elevated VDD	7.0V for a 5.0V design
VIHOVR	Maximum elevated positive input voltage	VDD + 0.5V
VILOVR	Maximum elevated negative input voltage	GND - 0.5V
TRIGIH	Maximum input positive current (Maximum current that can flow into an input or output without causing Latch-up)	>20mA
TRIGIL	Maximum negative input current (Maximum current that can flow out of an input or output without causing Latch-up)	>20mA
IDDQ	Maximum quiescent IDD current	Spec. Static IDD
IDD1	Measured quiescent IDD current	
IDDT	Maximum delta in IDD current after trigger condition VIH/LOVR or TRIGIH/L applied	<5.0%of IDDI

Test Patterns

1. Precondition device to a known state (Same as static IDD).
2. Ripple all inputs high and low in complement field.
3. Toggle all outputs high and low (Same as VOH/VOL).
4. Place high Z outputs in the high impedance state (Same as IOZ).

Since current is limited only by the external circuit, if the SCR triggers, the VDD power supply must limit current flow to a safe level (approximately 300mA). If the VDD supply can not be programmed to limit the current a resistor should be placed in series with the supply. Large power supply bypass capacitors (> 20uF) should also be removed during latch-up testing.

Testing starts with the program outputting a message identifying the following as elevated voltage tests for VBO triggering. VDD_{MAX} is then applied with the input levels set to the power rails (VIH = VDD_{MAX}, VIL = GND). Test pattern 1. that preconditions the device to a known state is then executed. The DC measurement system is then used to measure the amount of current flowing into the VDD pin(s). This value is compared to the IDDQ limit and also stored in a program variable (IDD1). If the measured value is greater than IDDQ the program outputs a message indicating the device failed the initial quiescent IDD current along with the value of VDD and the measured current. The device is then powered down and testing terminated.

If the measurement is less than IDDQ the test continues by increasing VDD to VDDOVR and then setting the input levels VIH = VIHOVR and VIL = VILOVR. Test pattern 2. is then executed to individually toggle each input pin high and low while the remaining pins are in the opposite state. The input levels then VDD are returned to the original values and test pattern 1. is executed again. The DC measurement system is used to measure IDD and the resulting value compared to a dual limit that requires the measurement to be greater than (IDD1 - IDDT) and less than (IDD1 + IDDT). If the measured value is outside of these limits the program outputs a message indicating the device has latched up (failed). If the measured value is within these limits the program outputs a message indicating the circuit did not latch up (passed). The program then outputs the value of VDD, VDDOVR and IDD. The device is then powered down in preparation for the next test.

It is important to point out that VDD is never reduced below VDD_{MAX} between the first and second IDD measurements (Do not power down between the first and second IDD measurements.) If the test system being used can not functionally drive the voltage levels required for the inputs, the DC measurement system should be used to individually toggle each input while the remaining inputs are functionally driven to the maximum capability of the tester.

Testing continues by outputting a message that identifies the following as trigger current tests for the input pins. VDD_{MAX} is then applied with the input levels set to the power rails (VIH = VDD_{MAX}, VIL = GND). Test pattern 1. is executed and the DC measurement system used to measure the current flowing into the VDD pin(s). This value is compared to the IDDQ limit and also stored in a program variable (IDD1). If the measured value is greater than IDDQ the program outputs a message indicating the device failed the initial quiescent IDD current along with the value of VDD and the measured current. The device is then powered down and testing terminated.

If the measurement is less than IDDQ the test continues by forcing current (TRIGIH / TRIGIL), using the DC measurement system, into or out of one of the input pins. The current is then removed and test pattern 1. executed. The DC measurement system is then used to measure IDD and the resulting value compared to a dual limit that requires the measurement to be greater than (IDD1 - IDDT) and less than (IDD1 + IDDT). If the measured value is outside of these limits the program outputs a message indicating the device has latched up (failed). If the measured value is within these limits the program outputs a message indicating the device did not latch up (passed). The program then outputs the pin name/number, VDD, IDD and (TRIGIH / TRIGIL) values. The device is then powered down in preparation for the next test. Starting from the point of applying VDD_{MAX} this sequence is repeated for both values of input current TRIGIH, TRIGIL and all input pins.

The High Z output pin test begins by outputting a message to identify the following tests as High Z outputs. The testing procedure is identical to the input pin tests with pattern 4. replacing pattern 1. Pattern 4. preconditions the High Z pins to the High Z state and when executed must terminate with the High Z pin(s) being driven by the tester. Make sure the pin being tested is in the High Z state and the same ending address is used for preconditioning before and after the test current is applied.

Output pin testing is very similar to the High Z tests and begins by outputting a message that identifies the following tests as output pins. The major difference in this test is that each pin must be tested in two states and only one test current is forced for each state. The outputs are first tested one at a time in the low state by using the appropriate vector addresses in pattern 3. The procedure is the same as was used in the Input and High Z tests, but only the TRIGIL test current is applied. The procedure is then repeated with the output pins preconditioned to the high state and TRIGIH test current applied.

Once the tests have been verified, special features can be added to the test program like allowing the operator to enter the test parameters, automatically varying the test parameters, or separating the two test types under operator control.

Test Procedure Summary

1. Apply maximum operating voltage conditions.
2. Precondition device to known state.
3. Measure and store quiescent IDD current, IDD1.
4. Compare quiescent current to IDDQ limit.
5. Output VDD and IDD values and abort test if current greater than IDDQ.
6. Apply test voltages or currents that could trigger SCR

Remove trigger conditions and return to maximum operating conditions without powering down.

Precondition device to known state.

8. Measure quiescent IDD current.
9. Power down.
10. Compare quiescent current to first measurement IDD1 ± IDDT limit.
11. Output Pass/Fail result and value of VDD, IDD, and trigger parameter (VIHOVR, VILOVR, TRIGIH, TRIGIL).

Applying Test Stimulus

It is important to point out that Latch-up is prevented only if the absolute maximum ratings are not exceeded. Since these ratings specify that the input levels must never exceed the power rails (e.g. VDD+ 10% to GND -10%), care must be taken to properly sequence VDD and the input stimulus. I/O pins must also be initialized to prevent high output currents due to contention between the device and tester. Devices with multiple VDD pins may require VDD to be applied in a given sequence. Check with the circuit designer to establish the proper sequence. The following sequence is recommended for applying and removing stimulus for all CMOS tests:

Power Up Sequence

1. Initialize tester to drive all input pins to 0V. Initialize tester I/O pins to the non drive mode with the output loads set to 0V.
2. Apply test voltage to the VDD pins in the proper sequence.
3. Set input test voltages VIL/VIH.
4. Set output load reference test voltage.
5. Set output compare test voltages VOL/VOH.

Power Down Sequence

1. Initialize tester I/O pins to the non drive mode.
2. Set output load reference to 0V.
3. Set input voltages VIL/VIH to 0V.
4. Remove voltage from the VDD pins in the proper sequence.

Chapter 16 Principles of Scan Testing

-Advanced Topic-

by Vernon Rubright

Objectives:

This section explains:

- The purpose of scan testing
- The benefits of scan testing
- Device types which benefit from scan testing
- Scan test circuitry
- Equipment required for scan testing

What is Scan Testing?

Scan testing is a way of decreasing the test development time and test time for certain types of devices. It depends on designing special test circuits into the devices (called Scan design). Scan design is a structured design methodology that greatly reduces the complexity of functional test generation. The philosophy is one of "divide and conquer" and can best be described with reference to the general model for a clocked (synchronous) logic circuit shown in Figure 16-1.

Synchronous Logic Circuit

Feedback makes testing difficult

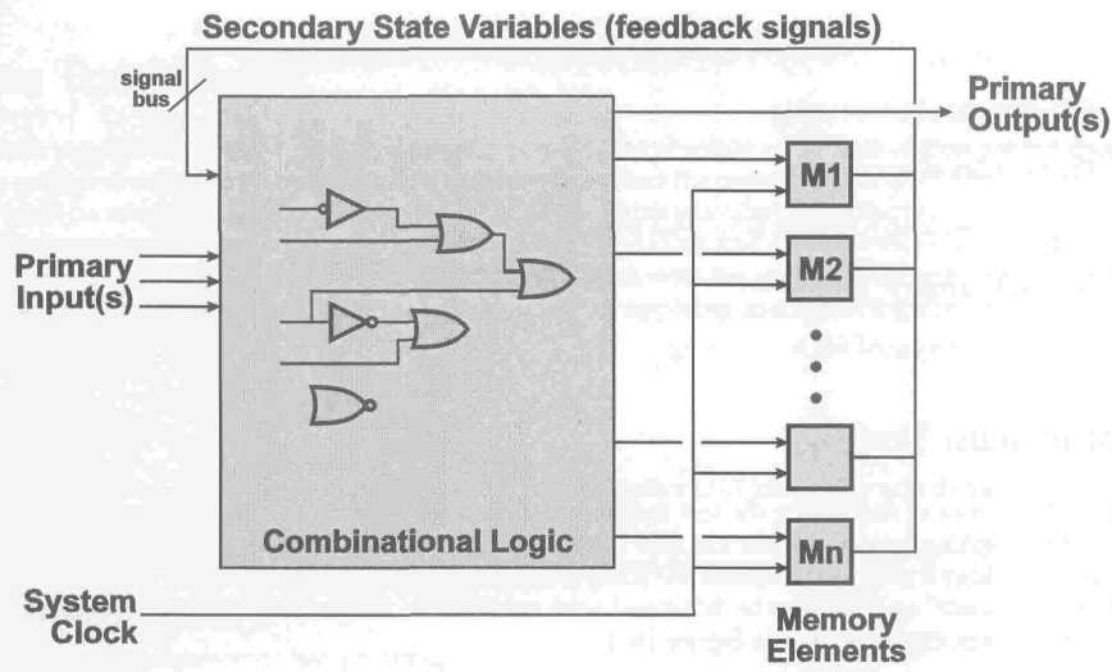


Figure 16-1

In this model, the major elements of the circuit are identified as a combinational logic section together with a bank of memory elements Ml, M2,..., Mn under control of a system clock. Inputs to the combinational logic consist of the device inputs (Primary Inputs) and the fed back secondary-state variables from the memory elements. The device outputs (Primary Outputs) are therefore a function of the present state of the device inputs together with the current states of the memory elements.

The future state of the memory elements also depends on both the primary inputs and the current recorded state of the memory elements themselves. It is this dependency of the future state on the present state that causes all the problems in test generation. The device inputs are the only inputs the test has direct control of and likewise the device outputs are the only outputs the test can directly observe. The problem is which section can be tested first since neither section is directly controllable or observable and the sections are mutually dependent on each other for correct operation.

The scan design methodology provides a solution to this problem by reducing the complexity of the circuit. Scan designs are based on the principle of providing the following test facilities:

1. All memory elements can be tested in isolation from the rest of the circuit.
2. The future state of the secondary-state variables can be set to any value independent of their present values.

The outputs of the combinational logic that drive into the memory elements can be observed directly.

Scan design is implemented by establishing a scan path through the memory elements as shown in Figure 16-*2.* Effectively, each memory element is now preceded by a 2 way switch (multiplexer) under the control of a common Scan Select signal. When Scan Select is off, the multiplexers connect the outputs from the combinational logic to the input sides of the memory elements, i.e. the circuit functions in its normal mode. When Scan Select is on, the memory elements are reconfigured into an isolated serial-in, serial-out shift register.

Scan Enabled Synchronous Logic Circuit

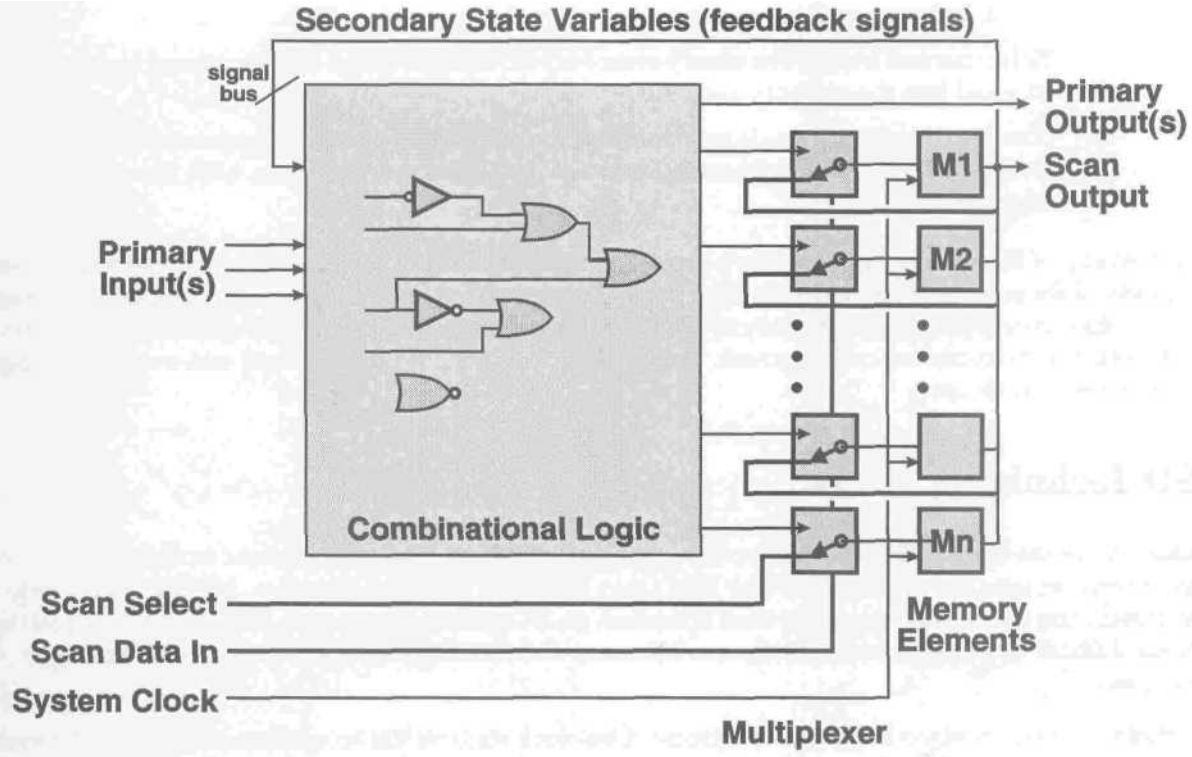


Figure 16-2

The serial data input is called Scan Data In and the serial data output is called Scan Data Out. In the scan mode the memory elements can be preset to any particular set of values simply by placing the values in sequence on the Scan Data In input and clocking the shift register with the System Clock. The testing strategy now becomes:

1. Select the scan-path mode, i.e. memory elements reconfigured into a shift register. Test the status and operation of each memory element using the Scan Data In, Scan Data Out and System Clock facilities. Suitable tests for a scan-path register are as follows:

a. FLUSH TEST - In this test all memory elements are initialized to logic 0 and a single logic 1 is clocked through from the Scan Data In input to the Scan Data Out output using the scan path (system) clock. The procedure can be repeated with a single logic 0 flushed through a background of logic 1s. This sequence checks the ability of each memory element to assume both logic states.

b. SHIFT TEST - In this test, the data sequence 00110011.. is shifted through the register. This sequence exercises each memory element through all combinations of present state and future state.

2. Determine a set of tests for the combinational logic, assuming
   a. total control of all inputs (primary and from the memory elements);
   b. direct observation of all outputs (primary and to the memory elements).

3. Apply each test in the following way:
   a. Select scan-path mode. Load the memory elements with test input values and establish additional test input values on the primary inputs.
   b. Select normal mode. The steady-state output response of the combinational logic can now be clocked into the memory elements.
   c. Return to scan-path mode and clock out the contents of the memory elements. Compare these values, plus the values directly observable on the primary outputs, with the expected response.

The philosophy of the scan design can now be seen more clearly. Rather than test the circuit as a single entity, the addition of the scan path allows each major segment to be tested separately and in a procedural manner. Standard tests can be defined for the memory elements (Step 1. above). The only test-generation problem is to generate tests for the combinational segment. This problem has been well researched and a variety of logic or fault simulators can be used.

LSSD Technique

A number of circuit design techniques have been developed to implement the scan methodology with Level Sensitive Scan Design (LSSD), developed by IBM Corp. receiving the most attention. LSSD is specifically aimed at reducing the dependency of system operation on AC parameters such as clock edge rise or fall times, which are difficult to simulate in the design environment and difficult to monitor in a manufacturing environment.

The technique provides direct control of two clocks. One clock controls the acceptance of data into the memory elements while the second clock controls their outputs. With direct dock control the timing of the circuit can be controlled to avoid potential race conditions resulting from the primary or secondary inputs propagating through the combinational circuits.

Scan Test Equipment

The total number of memory elements in a VLSI circuit can result in a very long scan path shift register (scan chain). Memory elements are often divided into multiple scan chains to simplify and speed up testing. Since scan test patterns are serial representations of the circuit states, it is not uncommon for these patterns to be millions of vectors long. Standard functional test pattern memory is designed to supply high speed parallel (pin wide) data and is not cost effective for scan testing.

Test equipment designed for scan testing has two functional test pattern memories. One is for parallel (pin wide) data and another is for serial (scan) data. Each tester channel (pin) can be programmed to use the parallel or serial test pattern. The serial memory is generally two bits wide (serial data in and out) and can be configured into multiple depth/width ratios under program control.

Glossary

AC Testing	AC testing guarantees that the device meets all of the timing specifications. AC testing is performed by setting up the appropriate timing values (edge placements) and signal formats as defined in the device AC specifications and then executing a functional test sequence.
Assembly Verification	Assembly verification is primarily to verify that the devices survived the assembly process and that they were assembled correctly. The tests performed during assembly verification are similar to that of package testing and may be a subset of package testing.
Bi-directional Pin	A device pin that functions as an input, an output and is also capable of turning off (going to a high impedance state).
Binning	A means of categorizing or sorting the tested devices into their appropriate groupings, either hardware bins or software bins.
Clamps	Hardware which limits the amount of voltage or current that is supplied by the test system during a test. Clamps are used to protect the test operator, the test hardware and the DUT.
Comparators	The circuitry located on the pin electronics card which senses the logic 0 and logic 1 levels from the DUT. The comparators are used during functional testing.
DC Testing	When a voltage or current is measured during the test and the pass/fail results are based upon the measured value. The PMU is designed to perform DC tests.
Device Characterization	Device Characterization is the process of determining the operating extremes of devices.
Device Specification	The device specification defines the exact performance conditions of the device. The specification includes voltages, currents, timings and a description of the device functions.
DPS	Device Power Supplies are used to supply voltage and current directly to the DUT. The power pin (VDD or DCC) of the DUT will usually be connected to a DPS.
Drivers	The circuitry on the pin electronics card which supplies the logic 0 and logic 1 levels to the DUT. A pin is said to be driven if the test system driver applies a voltage to it.
DUT	The semiconductor device being tested is often referred to as the DUT (Device Under Test). It is also sometimes referred to as a UUT (Unit Under Test).
Dynamic	A term used to indicate that the DUT is actively changing states, dynamic tests are associated with executing functional test vectors.
Dynamic Loads	The circuitry located on the pin electronics card which acts as a load and can be programmed to supply positive and negative currents. The dynamic loads can be used to supply IOL and IOH currents for loading DUT outputs. Dynamic loads are also referred to as programmable current loads.

Failure Analysis	Failure Analysis is the process of determining why a device has failed.
FDATA	Formatted vector data (logic 1s and 0s) combined with timing and signal format information.
Force Line	The force line is the current carrying line of a four wire system such as a power supply. The four wires of a four wire system are the High Force, High Sense, Low Force and Low Sense. In test systems the Device Power Supply and the Precision Measurement Unit are four wire systems.
Forcing	The term force, as in forcing voltage or forcing current, is often used to describe certain activities during testing. Force is used to describe the act of applying a certain value of voltage or current by the test system. Apply can be substituted for the word force.
Functional Testing	When the device is actively performing logical functions. Input data is supplied to the DUT and output data is read from the DUT. The functional comparator circuitry located on the pin electronics cards is used to determine the pass/fail results of the test.
Gross Testing	Performing a test with relaxed conditions (levels and timings) usually made to verify if the device is "functionally alive" without regard to the device specification parameter values.
Ground	To connect a signal pin or other electrical node to the test system reference node or to VSS.
Hot Switching	Occurs when a relay is opened while current is flowing through it, or when current immediately begins to flow after a relay is closed (i.e. when the two terminals of a relay are at different voltages when the relay is closed). Opening or closing a relay while current is flowing through it may result in damage to the relay. Through careful test programming this can be avoided.
ICC	The current consumed by the circuitry of a TTL device.
IDD	The current consumed by the circuitry of a MOS device.
IDDQ	The IDDQ test measures the quiescent current under varying logic conditions and provides improved test coverage as compared to the standard Static IDD test.
IIH	Input Leakage High is the maximum amount of current that is allowed to flow into an input pin when a high voltage value is forced onto the pin.
IIL	Input Leakage Low is the maximum amount of current that is allowed to flow out of an input pin when a low voltage value is forced onto the pin.
Incoming Inspection	Incoming Inspection is performed by the customer to insure the quality of the devices purchased before using them in an application.
Input Pin	A device pin that acts as a buffer between external signals and the internal logic of a device. The input senses the voltage which is applied to it and transmits a logic 0 or logic 1 level to the internal logic of the device.
IO Conflicts	A condition that exists when the test system and the DUT are simultaneously driving voltage into the same tester channel, this is also called Bus Contention.

IOH	Current Out High is the amount of current that an output must source when driving a logic 1. The output must be capable of supplying the specified IOH current while maintaining the correct VOH voltage.
IOL	Current Out Low is the amount of current that an output must sink when driving a logic 0. The output must be capable of accepting the specified IOL current while maintaining the correct VOL voltage.
IO Switching	The DUT alternates between receiving data from the test system (reading data) and supplying data to the test system (writing data). The same pin or set of pins functions as both inputs and outputs.
IOZH	Output High Impedance Leakage Current High is the maximum amount of current that is allowed to flow when a high voltage is applied to a bi-directional or three-state pin, and the pin is in the off or High-Z state.
IOZL	Output High Impedance Leakage Current Low is the maximum amount of current that is allowed to flow when a low voltage is applied to a bi-directional or three-state pin, and the pin is in the off or High-Z state.
Kelvin Connection	The Kelvin connection or Kelvin connect point is the point where the force and sense lines of a four wire system are connected. Four wire systems provide a very accurate method of supplying voltage or current to a remote location. In test systems the Device Power Supply and the Precision Measurement Unit are four wire systems.
Latch-up	A high current condition which exists within a CMOS device caused by applying an improper voltage level to a signal pin or by an improper voltage applied to VDD or ground. This condition can weaken the device or cause a catastrophic failure.
Margins	Margins refers to the amount by which a parameter can be varied before a failure occurs.
Military Testing	Military Testing involves performing more rigorous testing over temperature and documenting the results.
Negative Current	Current flowing from the DUT into the test system.
Noise Spike	When a signal level abruptly changes its voltage (and/or logic) level for a very short time. Sometimes called a Glitch.
Output Mask	A method of enabling or disabling an output comparison for a tester channel during a functional test.
Output Pin	A device pin that acts as a buffer between the internal logic of a device and the external environment. An output pin is capable of providing the correct voltages to produce a logic 0 or logic 1 level and also supplies the IOL/IOH current.
Output Sampling	The point in time at which the output signal of a DUT will be evaluated. The comparator circuitry will compare the output voltage to a pre-defined logic 1 or logic 0 level. The test system will then make a pass/fail decision. Output sampling is also called the strobe.
Package Test	Wafers are cut into individual die and each die is then assembled into a package.

	The packaged device is then tested to insure that the assembly process was correctly performed and to verify that the device still meets it's design specifications. Package test is also called final test.
Pattern Memory	A high speed memory which stores test vector information. Vector memory may also be called vector memory.
Pin Electronics	Circuitry located in the test head that is used to supply input signals to the DUT and to receive output signals from the DUT. The pin electronics are also called PE cards or I/O cards.
PMU	The Precision Measurement Unit, also called the Parametric Measurement Unit is used make accurate DC measurements. The PMU is capable of forcing voltage and measuring current or forcing current and measuring voltage.
Power Pin	A device pin that is connected to a power supply or ground. VDD and VCC are typical examples of a power pins. VSS and ground are also identified as power pins. Power pins have a structure that is different from signal pins.
Positive current	Current flowing from the test system into the device.
Preconditioning	Setting a device into the proper logic state so that a test may then be performed. A functional vector sequence is often required in order to prepare the DUT for a DC test.
Pre/Post Burn-in	Pre/Post Burn-In is testing devices before and after they are "burned in" to verify that the process did not cause certain parameters to drift. This process weeds out infant mortality devices.
QA Test	Quality Assurance testing is performed on a sample basis to insure that the package test was performed correctly.
RVS	Reference Voltage Supplies are used to supply voltage references for logic 0 and logic 1 levels to the driver and comparator circuitry located on the pin electronic cards. These voltages are used to establish VIL, VIH, VOL and VOH.
Sense Line	The Sense line is the non-current carrying line of a four wire system such as a power supply. The sense line senses the voltage at the point where it is connected to the force line. The four wires of a four wire system are the High Force, High Sense, Low Force and Low Sense. In test systems the Device Power Supply and the Precision Measurement Unit are four wire systems.
Shared Resource	A test system resource that is limited in number and must be shared over multiple tester channels such as timing generators or a single PMU.
Signal Format	The wave shape of an input signal supplied by the pin electronics driver circuitry.
Signal Pins	Input, output, Tri-State® and bi-directional pins, not power or ground pins. Signal pins have a structure which is different from power pins.
Sink	A term used to describe current flow from the test system into a device output pin (positive current). When an output is in the logic 0 state it can accept current from the test system which will flow through the device to ground.
Source	A term used to describe current flow from a device output pin into the test system

	(negative current). When an output is in the logic 1 state it can supply current which will flow from the DUT into the test system.
Static	This term indicates that the DUT is in a non-active condition, no input or output signals are changing.
Test Cycle	The time duration of one test vector execution. The test cycle is based on the operating frequency of the DUT. The test cycle time can be determined by the formula: Cycle = 1/frequency. The test cycle is also known as the period.
Test Patterns	A representation of the states of inputs and outputs for the various logical functions that the device is designed to perform. Input data is supplied to the DUT by the test system. Output pattern data is compared against the response from the output pins of the DUT. During a functional test the test patterns are executed or applied to the DUT. In the event that the expected output data does not match the output data from the DUT a functional failure occurs. Test patterns are also called test vectors or truth tables. Test vectors are often represented as a sequence of ones and zeros or other characters which represent logical levels.
Test Philosophy	A consensus of opinion of what is the best method of testing within a given company. It is based on their particular requirements, the selling price of their devices and often it is influenced by past experiences.
Test Program	The test program is what controls the test hardware in a manner that will guarantee that the DUT meets or exceeds all of its design parameters.
Test System	The test system is electronic and mechanical hardware used to simulate the operating conditions that the DUT will experience as it is used in the intended application, and separate good devices from defective ones. The test system is often referred to as ATE or Automated Test Equipment.
Test Vectors	See Test Patterns. Test vectors are also called test patterns or truth tables.
Tester Channel	Circuitry on the pin electronics card which applies and/or processes voltage, current and timing for one DUT pin. Also called a tester pin.
Tester Per Pin	Each tester channel has its own resources, such as independent levels, timings and PMU for each tester pin.
Three-State Output	A device pin that functions as an output pin but has the added capability of turning off (going to a high impedance state) Also called Tri-State® output.
Truth Tables	See Test Vectors. Truth Tables are also called test patterns or test vectors.
UUT	The semiconductor device being tested is often referred to as the DUT (Device Under Test). It is also sometimes referred to as a UUT (Unit Under Test).
VCC	The supply voltage for a TTL device.
VDD	The supply voltage for an MOS device.
VDDmax	The lowest supply voltage at which a device is guaranteed to still operate correctly.
VDDmin	The highest supply voltage at which a device is guaranteed to still operate correctly.

Vector Memory	A high speed memory which stores test vector information. Vector memory may also be called pattern memory.
VIH	Voltage In High is the voltage value applied to an input when applying a logic 1. The VIH value represents the minimum guaranteed voltage value that can be applied to an input and still be recognized as a logic 1 by the DUT circuitry.
VIL	Voltage In Low is the voltage value applied to an input when applying a logic 0. The VIL value represents the maximum guaranteed voltage value that can be applied to an input and still be recognized as a logic 0 by the DUT circuitry.
VOH	Voltage Out High is the voltage value produced by an output when driving a logic 1. The VOH value represents the minimum guaranteed voltage value that will be produced by the output when driving out a logic 1.
VOL	Voltage Out Low is the voltage value produced by an output when driving a logic 0. The VOL value represents the maximum guaranteed voltage value that will be produced by the output when driving out a logic 0.
VREF	The reference voltage associated with the dynamic loads. VREF controls the switching point of IOL and IOH currents.
VSS	The power pin that provides a return path for the power supplied to the VDD or VCC pin.
Wafer Test	Testing of individual devices when they are still in wafer form. This is the first attempt at separating the good die from the bad. This activity is also referred to as wafer sort.

Answers to Review Questions

Section 3 - Introduction to Test

1. Digital test programs are often segmented into three separate sections. These sections are:

   AC. DC. Functional

2. Positive current flow is defined as:

   a) Current flowing from the test system into the DUT.

   b) Current flowing from the DUT into the test system

   c) Current flowing from a higher voltage to a lower voltage

3. A device pin that functions as both an input and output is called a:

   An I/O or bi-directional pin.

4. Which resource of the test system hardware is used to provide voltage and current to power the DUT?

   a) RVS.

   b) DPS.

   c) Driver.

5. Which resource of the test system hardware is used to perform very accurate voltage and current measurements?

   The PMU.

6. Which type of testing verifies correct operation of the various logical operations of the device?

   A Functional Test verifies the correct operation of the various logical operations of the DUT.

7. When using the PMU to force current, which clamp should be used?

   a) A current clamp

   b) A voltage clamp

   c) Neither, clamps are not used when forcing current

8. When using the PMU to force voltage, which clamp should be used?

   a) A current clamp

   b) A voltage clamp

   c) Neither, clamps are not used when forcing current

Section 4 - Workbook Exercise

VOH/IOH

1. VDD voltage will be set to 4.75V.
2. IOH current will be forced. The correct value of IOH is -5.2mA.
3. Voltage will be measured during the test. The test will fail if the measured voltage is less than 2.4V.

VOL/IOL

1. VDD voltage will be set to 4.75V.
2. IOL current will be forced. The correct value of IOL is 8.0mA.
3. Voltage will be measured during the test. The test will fail if the measured voltage is greater than 0.4V.

VIL/VIH

1. VDD voltage will be set to 4.75V and 5.25V.
2. The VIH (input high level) will be set to 2.2V.
3. The VIL (input low level) will be set to 0.8V.

IIL/IIH

1. VDD voltage will be set to 5.25V.
2. When forcing a high voltage on the inputs Vin will be set to 5.25V.
3. Current will be measured during the test. The test will fail if the measured current is greater than +10mA.
4. When forcing a low voltage on the inputs Vin will be set to 0.0V.
5. Current will be measured during the test. The test will fail if the measured current is less than -10mA.

IOZ

1. VDD voltage will be set to 5.25V.
2. When forcing a high voltage on the outputs Vout will be set to 5.25V.
3. Current will be measured during the test. The test will fail if the measured current is greater than +10mA.
4. When forcing a low voltage on the outputs Vout will be set to 0.0V.
5. Current will be measured during the test. The test will fail if the measured current is less than -10mA.

IDD Dynamic Current

1. VDD voltage will be set to 5.25V.
2. The output current loading (Iout) will be set to 0.0mA.
3. This test requires the execution of a functional test loop during the measurement of the IDD current. The frequency of the functional test will be 66MHz.
4. During this test the IDD current will be measured. The test will fail if the measured current is greater than +120mA.

Section 4 - Device Specification

1. Standard TTL input levels are:
   a) VIL = 1.5V (VDD* 0.3) VDD = 5.0
      VIH = 3.5V (VDD * 0.7) VDD = 5.0
   b) VIL = 0.4V
      VIH = 2.0V
   c) VIL = 0.8V
      VIH = 2.4V
   d) **None of the above.** — See page 2-9 for the answer. 0.8V and 2.0V.

2 Which DC test forces voltage onto input pins and then measures the current flow?

The IIH/IIL tests.

3. From the sample device specification in chapter two, find the commercial IDD current limit for the 256 x 4 device
   The maximum current flow limit is:

The commercial IDD limit is 120 ma.

4 When an output is in the logic 0 state, what test is performed to guarantee the correct output voltage level?

a) VIL
b) VOL.
c) IOZL
d) None of the above

5. From the sample 256 x 4 device specification, find the commercial operating temperature range:

The commercial operating temperature range is 0° c to 70° c.

6. In the sample 256 x 4 RAM device specification, what current limit values would be used in the IIL/IIH test?

The values used for limits during the ILL/IIH tests are ±10 ua.

5. In the sample 256 x 4 device specification, the values for IOL and IOH are equal,

a) True
b) False.

Section 5 - Opens and Shorts

1. Device pins must be preconditioned for the opens and shorts test. To be properly preconditioned, they are:
   a) Programmed to VDD
   b) Programmed to ground.
   c) Floating
   d) None of the above

2. When performing the opens and shorts test using the PMU:
   a) A positive current is forced
   b) A negative current is forced
   c) A or B - more correct, A and B.
   d) None of the above

3. In the datalog example of the opens and shorts test pin 2 is failing because it is:
   a) open
   b) shorted

4. In the datalog example of the opens and shorts test pin 4 is failing because it is:
   a) open
   b) shorted

5. In the datalog example of the opens and shorts test, pin 2 measures as -3.0V. This reading is caused by:
   a) The PMU voltage clamp
   b) The pin being shorted
   c) The maximum voltage range of the PMU
   d) None of the above

Section 6 - DC Tests

1. The program flow is most important to:
   a) Characterization testing.
   b) Engineering evaluation.
   c) Production testing.

2. What is the purpose of binning?

   To separate good devices from bad ones.

   Pass bins categorize good devices into different grades - good, better, best.

   Fail bins categorize bad devices based on the reason they failed. This allows the engineer to follow trends which help to identify and head off problems.

3. What are some of the items shown on a test summary?

   Total devices tested.

   Total devices that passed and that failed.

   Total devices in each bin.

   Lot or run numbers.

   Operator ID.

   Temperature or test type.

   Percentage of total devices tested for all the bins shown.

4. What is the information in a test summary used for and who uses it?

   Test summaries are used by product engineers to follow trends which help to identify and head off problems.

   Test summaries are used by product engineers to track yields. This information is also used by cost accountants in determining the cost of manufacturing a product.

   Test summaries are used by production to track quantities of tested devices and track inventory.

5. What is the meaning of "TBD" in a test specification?

   "To Be Determined" - This is a way of not having to specify limits on a data sheet.

6. The current flow for a static IDD test is
   a) A positive current.
   b) A negative current.

c) AorB.

d))None of the above.

7. In most cases, a dynamic current test will result in:

a) Less current than a static IDD test.

b) More current than a static IDD test.

c) The same amount of current as the static IDD test.

d) None of the above.

8. To perform a static IOL test, the outputs are first preconditioned to the logic 1 state:

a) True.

b) False.

9. When performing an IOH measurement, the current is:

a) A positive current.

b) **A negative current.**

c) A or B.

d) None of the above.

10. When performing a VOL/IOL measurement, the specified current is 4 ma and the specified VOL is 0.4V. What is the maximum value of resistance that the output may contain and still pass the test?

a) 50Ω.

b) I00Ω

c) It depends on the value of VDD.

d) None of the above.

11. The input leakage test (IIL/IIH) verifies:

a) Input capacitance.

b) Input threshold levels.

c) Input resistance.

d) None of the above.

12. When performing the IIL/IIH test, what is the advantage of making a parallel measurement?

The advantage of making a parallel measurement is speed - The parallel test method is many fanes faster.

13. Which test method should not be used when performing the IIL/IIH test on resistive inputs (inputs with pull-ups or pull-downs)?

a) Serial Method.

b) Ganged method.

14. The IOZ (high impedance current) test is performed on (circle all correct answers):
    a) Pure input pins.
    b) Pure output pins.
    c) Three-state output pins.
    d) Bi-directional output pins.
    e) Pins defined under the IOZ test in the test specification.

15. Before the IOZ test is performed, the pins to be tested must first be preconditioned to:
    a) A logic 1 state.
    b) A logic 0 state.
    c) An off state (High Z state).
    d) A bi-directional state.

16. To perform the VI (input clamp) test:
    a) A negative voltage is forced.
    b) A positive voltage is forced.
    c) A negative current is measured.
    d) None of the above.

17. What preconditioning is required to make the IOS (output short circuit) test?
    Set outputs to logic 1 (high).

Section 7 - Functional Test

1. If a test specification defines VDD as 5.0V ± 10%, what would be the value of VDDmin?

   VDDmin would be 4.5 Volts.

2. If a test specification defines VDD as 5.0V ± 5%, the functional tests should be executed at what voltage?

   a) 5.00V
   b) 4.75V and 5.52V.
   c) 4.50V and 5.50V.
   d) 4.57V and 5.25V.
   e) None of the above. — The correct values are 4.75 & 5.25.

3. The purpose of an AC output load is to simulate the environment the output would encounter when connected to
   one or more device inputs.

   a) True
   b) False

4. The VIL/VIH specifications are verified by executing:

   a) A DC test.
   b) A Functional test.
   c) An AC test.
   d) None of the above.

5. When a test is made to guarantee the VIL/VIH specification and a failure occurs, the failing pin (as seen on the
   failing datalog) will be:

   a) An input pin.
   b) An output pin.
   c) A power pin.
   d) None of the above.

6. If a VIH test is made with VIH set at 2.0V and the result of the test is a failure, how could the voltage be changed in
   order to make the test pass?

   a) Lower the VIH voltage by 0.4V.
   b) Raise the VIH voltage to VDD.
   c) Raise the VDD voltage by 0.4V.
   d) None of the above.

7. When performing the VOL/IOL VOH/IOH test functionally, the comparators are set to the specified VOL/VOH levels and the dynamic (or resistive) current loads are used to supply IOL/IOH. When this test results in a pass condition, all four parameters (VOL/IOL VOH/IOH) are guaranteed to have met the test specification.

   a) True.

   b) False.

8. For debugging, if the functional VOL/IOL VOH/IOH test fails, it may be necessary to increase the test period (slow the test frequency) and repeat the test in an attempt to verify correct operation of VOL and VOH?

   a) True.

   b) False.

9. When performing the Z-State (high impedance) test as a functional test, some method of output loading must be used to provide an intermediate voltage level. This voltage level will be sensed by test comparators when the output pin enters a high impedance state. A correct value for the intermediate voltage level would be:?

   a) 5.0V.

   b) 2.0V

   c) 3.0V.

   c) 0.4V.

   d) None of the above.

10. The purpose of performing a gross functional test is:

    To verify that the device performs all of its basic functions correctly.

11. What is the advantage of using equation based timing?

    Changing the value of one variable can change all timing by the scale factor while keeping all timing relationships correct. This eliminates the need to calculate the new timing information and hard coding it into the program.

12. Open drain outputs do not have the ability to:

    a) Drive a logic 0.

    b) Drive a logic 1.

    c) Source current.

    d) B and C. — Can't drive a logic one and can't source current.

13. What is the meaning of RO in the statement "The input control signal is in RO format."?

    RQ — Return to One — Starts high, goes low, returns to high.

14. What signal format must be used to guarantee both setup and hold times?

    SBC — Surround by complement — Starts at the complement, goes to active data, returns to the complement.

15. When data changes only at TO, what is the signal format called?
    a) DNRZ.
    b) RZ.
    c) RO.
    d) None of the above — It is NRZ.

16. What would the signal look like for an SBC/XOR signal when the functional data is a logic 0? Draw the wave shape:
    Starts high, goes low, returns to high

17. What signal format switches between a drive condition and a high impedance state?
    ZD/DZ - High impedance to drive or Drive to High Impedance.

18. What is the formula used to find the test cycle time when the operational frequency is known?
    Cycle =l/freq.

19. What is the advantage of performing a functional opens and shorts test rather than a DC opens and shorts test?
    Speed which reduces cost.

20. When performing a functional opens and shorts test current is forced by:
    a) The PMU.
    b) The programmable (dynamic) current loads.
    c) Voltage is forced, not current.
    d) None of the above.

21. A functional opens and shorts test uses the comparators on the Pin Electronics cards to detect open shorted conditions.
    a) True.
    b) False.

Section 8 - AC Testing

1. The amount of time data must be present after a reference signal reaches a certain voltage point is:
   a) Setup time.
   b) Propagation time.
   c) Minimum pulse width time.
   d) **None of the above — Hold time.**

2. If the test specification defines the setup time as 12ns but the device functions property with a setup time of 10ns, the device:
   a) Fails to meet the test specification.
   b) **Exceeds the test specification.**
   c) I do not understand the question.
   d) None of the above.
   e) All of the above.

3. It is not possible (or at least not correct) to measure negative propagation delay time.
   a) True
   b) False

4. Propagation delay measurements are only made on output signals.
   a) True
   b) False

5. The output enable test is a measurement of an output going from driving valid data to a high impedance state.
   a) True
   b) False — It is the measurement of an output going from a high impedance state to driving valid data.

6. The output enable and output disable tests require an external reference voltage to indicate when the output has entered a high impedance state.
   a) True
   b) False

7. Read and record is a method of testing often used in production testing because it is the fastest test method.
   a) a)True
   b) b)False — It is used during characterization.

8. The Go/No-go testing method is often used when the exact performance of a device must be recorded, as in military testing.
   a) True
   b) False — It dose not produce a measured result.

Section 9 - Device Characterization

1. The binary search method is also known by another name. What is the other name?

   Successive approximation.

2. Standard test system utilities such as Shmoo Plots and Level Search routines are effective tools for device characterization.

   a) True

   b) False

3. It is often necessary to develop test vector patterns specifically for device characterization.

   a) True

   b) False

4. Shmoo Plots are used because they operate faster than search routines.

   a) True

   b) False — Shmoo Plots are slower. They also display more information graphically which is easier to read than a stack of measured data.

5. 5.A linear search typically executes slower than a binary search.

   a) True

   b) False

6. What hold time formula is used when making a binary search?

   Hold Time = ("the result of the binary search" - "reference signal")

Section 10 - Test Vector Development

1. 1 .When should the test engineer meet with the design engineer to discuss test vector development?
   a) Fairly early in the project
   b) Early in the design stage of the product.

2. The majority of test vectors are being developed from simulation data.
   a) True.
   b) False

3. When method of simulation results in one line of simulation data representing one test cycle?
   Cycle based simulation.

4. Where are test vector pattern stored within the test system?
   Vector memory or Pattern memory.

5. Does the number of test vectors (test cycles) affect the total test time?
   a) Yes — The time require to execute a functional test sequence is the total number of vector cycles multiplied by the cycle time..
   b) No

6. What is the purpose of a sequence of test vectors?
   Test vectors are to stimulate the DUT in such a way as to produce a specific response from the DUT and to verify the correct response of the DUT outputs

Section 11 - Test Program Development issues

1. What are the requirements for an engineering test program?

   An engineering program must verify the functionality of the device and may collect data. It should be structured to allow ease in modifying parameters (voltages, currents and timing) and in debugging. See page 10-1

2. What is the purpose of the production test program and how does it differ from the engineering program?

   A production program separates devices into categories or bins based on test results. A production program is also stream lined to execute a quickly as possible.

   The biggest difference between an engineering program and a production program is that the engineering program is more flexible and can be easily modified, this generally results in the use of variables which cause the program to run slower, while the production program executes faster 1

3. What test system routines are likely to be used within a characterization program?

   Level and timing search routines and Shmoo plots.

4. There are many issues which should be considered before test program development begins, name at least three:

   What is the purpose of the program — Is it an engineering, characterization or production program.

   Speed or throughput. The test flow or sequence of the tests can have a dramatic effect on throughput. Higher failure rate tests should be ahead of lower failure rate tests.

   Data collection. How much data needs to be collected during the testing.

   Compromises. Will the program include "at speed" testing? Or are there any compromises required to improve throughput or compromises due to test system limitations.

   Types of tests. Will the program include "at speed" testing? Is temperature testing required? Are drift tests required over temperature. Are there multiple pass and/or fail bins? Will testing include burn-in?

5. Why is a loadboard test important?

   Loadboard tests verify that the correct loadboard is installed.

   Loadboard tests can verify the functionality of the loadboard.

6. If the power-up or power-down sequence is performed incorrectly, what may occur?

   If the power-up sequence is incorrect the device may latch-up. This can cause damage to the device and test hardware.

   If the power-down sequence is incorrect it is possible to damage the device by incorrectly removing power from the device or by leaving signals applied while disconnecting the device from the hardware.

7. During production testing, what action is normally taken when a failure occurs?
   a) Testing stops.
   b) The test summary is updated.
   c) A fail bin is selected.
   d) **All of the above.**
   e) None of the above.

Section 12 - Creating a Test Program

1. What is the very first test that should be run when debugging a new test program?
   a) The opens and shorts test
   The loadboard test

   b) The gross IDD test
   c) The gross functional test
   d) None of the above

2. What is the purpose of the Gross IDD current test?
   To determine if it is safe to continue testing the device.

3. What steps should be taken to verify the functional test setup before testing the first device?
   Start by looping on the functional test. Then use **an oscilloscope to observe each tester channel, looking for the correct signal timing, format and level. Also, verify that power and ground are at the correct levels.**

4. When executing functional tests for the very first time, what should be done if the test result is a pass?
   a) Go out for beer and pizza
   b) Remove the device from the socket and run the test again
   c) Test another device
   d) None of the above

5. If the functional test fails and the input leakage test also fails, which test should be debugged first?
   a) Functional
   b) Input leakage

Section 13 - Troubleshooting

1. How do you begin to debug a failure?

   a) Turn on the datalog and examine the results.

   b) Either remove the device or short pins on the device, whichever will produce a meaningful result, or the replace device with a "Golden" part.

2. If all the devices being tested appear to have the same failure you should:

   a) Discuss the problem with a colleague.

   b) **Get a known good (standard) device and test it — Also. Verify the program and the test setup.**

   c) Call maintenance.

   d) None of the above.

3. If every test fails, what test should you debug first?

   If every test fails, start with the Opens and Shorts Test and then proceed to IIL and IIH.

4. If the VIL/VIH functional test fails and all other tests pass what is the most likely problem?

   If the VIL/VIH tests fail, it is most likely an input levels problem. This could be due to excessive test noise (check the noise on VDD) or it could simply be a defective device. Try raising VIH by .4v and lowering VIL by .4 volts, this should make the device pass the test.

5. When debugging a test problem is it important to understand the entire test program flow.

   a) **Yes**

   b) No

6. Explain why you selected yes/no in the previous question:

   If YOU don't understand the sequence of the tests, you can not determine which test to debug first.

7. If the functional test is failing what could be changed in an attempt to make it pass?

   a) Relax the input levels

   b) Relax the output levels

   c) Relax the timing

   d) **All of the above**

8. What is the scope tool used for?

   The scope tool is used to view the functional patterns as they execute.

   It is also used to generate a sync signal for synchronizing an oscilloscope with specific patterns. This allows the user to examine input and output levels, and timing.

9. The search utility will generally provide two methods of performing a search. What are they?

   Search utilities generally can perform both a Binary and a Linear search.

10. Shmoo plots provide valuable information but they may require considerable time to execute.

    a) True

    b) False

11. Learning to manipulate the test hardware will greatly increase your productivity when developing or debugging test programs.

    a) True

    b) False

12. Most test systems provide a means of modifying test vector data stored in vector memory directly from the keyboard.

    a) True

    b) False

Section 14 - Qualifying and Documenting the Test Program

1. Why is it essential to save sample devices?

   They are used to correlate test results between various testers. Also, later sample devices are used to verify test hardware and programs.

2. To what does the term "margins" refer?

   Margins refers to the amount by which a parameter can be varied before a failure occurs.

4. If a test vector pattern is working correctly, a Shmoo plot of the VDD vs. VIL will show no "holes."

   a) True

   b) False — It might still have holes. The holes in a Shmoo plot may be caused by device characteristics.

4. List at least four items which must be included in a test program documentation package:

   A summary of all the test development activity.

   Note any deviations from the test specification and reference any supportive documentation such as Shmoo plots, datalogs and memos.

   State the test time-State the details of any special requirements such as external hardware or non-standard auto-handling.

   Include all correspondence between the test engineer, design engineer, product engineer, marketing and customers.