WILMER CUTLER PICKERING
HALE AND DORR LLP
James M. Dowd (SBN: 259578)
james.dowd@wilmerhale.com
Bethany Stevens (SBN: 245672)
bethany.stevens@wilmerhale.com
350 South Grand Avenue, Suite 2100
Los Angeles, CA 90071
Telephone: (213) 443-5300
Facsimile: (213) 443-5400

WILMER CUTLER PICKERING
HALE AND DORR LLP
Elizabeth Rogers Brannen (SBN: 226234)
elizabeth.brannen@wilmerhale.com
1117 California Avenue
Palo Alto, CA 94304
Telephone: (650) 858-6000
Facsimile: (650) 858-6100

Attorneys for Plaintiff
TERADYNE, INC.

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

TERADYNE, INC.

Plaintiff,

vs.

XYRATEX LTD. and XYRATEX INTERNATIONAL, INC.

Defendants.

Case No. CV09-2580 MMM (CWx)

PLAINTIFF TERADYNE, INC.'S COMPLAINT FOR PATENT INFRINGEMENT;

DEMAND FOR JURY TRIAL

Plaintiff Teradyne, Inc. ("Teradyne") hereby alleges for its Complaint for Patent Infringement against the defendants, Xyratex Ltd. and Xyratex International, Inc. (collectively, "Defendants"), as follows:

JURISDICTION AND VENUE

1. This Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a) as this action arises under the patent laws of the United States, 35 U.S.C. § 1, *et seq.*

2. Venue is proper in this judicial district under 28 U.S.C. § 1400(b) as Defendants each have transacted business and committed acts of infringement in this district, and this action arises from the transaction of that business and that infringement.

3. Defendants are each subject to this Court's personal jurisdiction.

PARTIES

4. Plaintiff Teradyne is a Massachusetts corporation having a principal place of business at 600 Riverpark Drive, Reading, Massachusetts 01864. Teradyne is a supplier of automatic test equipment to major electronics, semiconductor and automotive companies worldwide.

5. Upon information and belief, defendant Xyratex Ltd. is a limited liability company organized under the laws of Bermuda with its principal place of business at Langstone Road, Havant PO9 1SA, United Kingdom.

6. Upon information and belief, defendant Xyratex International, Inc. is a California corporation having a principal place of business at 2031 Concourse Drive, San Jose, California 95131.

FACTUAL BACKGROUND

7. Teradyne is the owner by assignment of U.S. Patent No. 6,681,351 (the "'351 patent"), entitled "Easy to Program Automatic Test Equipment," which was duly and legally issued on January 20, 2004. A true and correct copy of the '351 patent is attached hereto as Exhibit A.

8. Teradyne is the owner by assignment of U.S. Patent No. 6,966,019 (the "'019 patent"), entitled "Instrument Initiated Communication For Automatic Test Equipment," which was duly and legally issued on November 15, 2005. A true and correct copy of the '019 patent is attached hereto as Exhibit B.

CLAIM FOR PATENT INFRINGEMENT

(Against All Defendants)

9. Teradyne restates and incorporates by reference its allegations in Paragraphs 1 through 8 as if fully set forth herein.

10. Upon information and belief, Defendants are infringing and have infringed and/or induced infringement of and/or contributed to the infringement of the '351 patent, including without limitation by making, using, selling and offering for sale products practicing the inventions claimed in the '351 patent, including but not limited to their XCalibre and Optimus products.

11. Upon information and belief, Defendants are infringing and have infringed and/or induced infringement of and/or contributed to the infringement of the '019 patent, including without limitation by making, using, selling and offering for sale products practicing the inventions claimed in the '019 patent, including but not limited to their XCalibre and Optimus products.

12. Upon information and belief, Defendants' infringement has been and continues to be willful and deliberate.

13. As a result of Defendants' infringement, Teradyne has suffered substantial damages and will suffer severe and irreparable harm, unless infringement is enjoined by this Court.

PRAYER FOR RELIEF

WHEREFORE, Teradyne prays for judgment as follows:

a. Adjudge that Teradyne is the owner of the '351 patent and all rights of recovery thereunder, and that the '351 patent is good and valid in law and enforceable;

b. Adjudge that Teradyne is the owner of the '019 patent and all rights of recovery thereunder, and that the '019 patent is good and valid in law and enforceable;

c. Adjudge that Defendants have infringed and, unless enjoined, will continue to infringe the '351 patent, and that such infringement has been willful and deliberate;

d. Adjudge that Defendants have infringed and, unless enjoined, will continue to infringe the '019 patent, and that such infringement has been willful and deliberate;

e. Permanently enjoin Defendants, their officers, directors, employees, agents, licensees, successors, assigns, and all persons in concert with them, from further infringement of the '351 patent;

f. Permanently enjoin Defendants, their officers, directors, employees, agents, licensees, successors, assigns, and all persons in concert with them, from further infringement of the '019 patent;

g. Award Teradyne compensatory damages caused by Defendants' infringement, plus pre-judgment and post-judgment interest accrued on such amounts as provided by law;

h. Treble the damages assessed against Defendants pursuant to 35 U.S.C. § 284;

i. Declare this case exceptional under 35 U.S.C. § 285 and award Teradyne its costs and attorneys' fees;

//

//

//

j. Award Teradyne such other and further relief as this Court deems just and proper.

Respectfully submitted,

WILMER CUTLER PICKERING
HALE AND DORR LLP

Dated: April 14, 2009

By: __________________
James M. Dowd

DEMAND FOR JURY TRIAL

Pursuant to Federal Rule of Civil Procedure 38, plaintiff Teradyne, Inc. hereby demands a jury trial on all issues so triable.

Respectfully submitted,

WILMER CUTLER PICKERING
HALE AND DORR LLP

Dated: April 14, 2009

By: ____________________
James M. Dowd

350 South Grand Avenue, Suite 2100
Los Angeles, CA 90071

EXHIBIT A

US006681351B1

(12) **United States Patent**
Kittross et al.

(10) **Patent No.: US 6,681,351 B1**
(45) **Date of Patent: Jan. 20, 2004**

(54) **EASY TO PROGRAM AUTOMATIC TEST EQUIPMENT**

(75) Inventors: **Andrew W. Kittross**, Beverly, MA (US); **Allan M. Ryan**, Billerica, MA (US)

(73) Assignee: **Teradyne, Inc.**, Boston, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/417,034**

(22) Filed: **Oct. 12, 1999**

(51) **Int. Cl.**^{7} **G11C 29/00**; G06F 19/00

(52) **U.S. Cl.** **714/724**; 702/119

(58) **Field of Search** 714/724, 742, 714/726, 741, 738; 395/704, 500, 183.01; 364/468.28, 580; 702/119; 371/22.1; 368/118

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,668,745 A	*	9/1997	Day	702/121
5,794,007 A	*	8/1998	Arrigotti et al.	714/720
5,828,674 A	*	10/1998	Proskauer	714/724
5,892,949 A	*	4/1999	Noble	717/125
5,910,895 A		6/1999	Proskauer et al.	364/468.28
5,913,022 A	*	6/1999	Tinaztepe et al.	714/25
5,968,192 A	*	10/1999	Kornachuk et al.	714/724
6,128,759 A	*	10/2000	Hansen	700/121
6,205,407 B1	*	3/2001	Testa et al.	702/119
6,249,893 B1	*	6/2001	Rajsuman et al.	714/738
6,353,904 B1	*	3/2002	Le	324/158.1
6,408,412 B1	*	6/2002	Rajsuman	714/724

FOREIGN PATENT DOCUMENTS

FR	2658933 A	8/1991	G06F/11/30

OTHER PUBLICATIONS

National Instruments Corporation, Austin, Texas, "Instrumentation Newsletter", vol. 11, No. 3, Third Quarter 1999.

R. Hutson: "Notes, Tutorial 8, Digital Test Applications," Oct. 18, 1998, IEEE Computer Society, Test Technology Technical Committee, Washington DC USA; XP002157245; figures 2–1.

Masciola J.A., et al: "A Software Architecture for Mixed Signal Funtional Testing" Proceedings of the International Test Conference, US, New York, IEEE, Oct. 2, 1994, pp. 580–586, XP000520019; ISBN: 0–7803–2103–0; abstract.

http://www.hp.com/pressrel/feb97/24feb97d.htm, HP Introduces Major Upgrade to HP VEE Visual Test Programming Language, Palo Alto, California, Feb. 24, 1997, 3 pages.

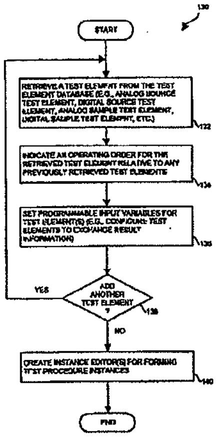
* cited by examiner

Primary Examiner—Phung M. Chung
Assistant Examiner—Anthony T. Whittington
(74) *Attorney, Agent, or Firm*—Lance Kreisman

(57) **ABSTRACT**

The invention is directed to techniques for providing a test procedure for testing a device using automatic test equipment (ATE). An ATE system includes memory having a test application stored therein, a test interface to connect to the device, and a processor coupled to the memory and the test interface. The processor is configured to operate in accordance with the test application to (i) provide a series of instructions based on a test procedure defining a device testing task, and (ii) control the test interface based on the provided series of instructions in order to test the device. The test procedure includes multiple test elements. Each test element defines instructions and programmable input variables that direct the processor to perform a particular test operation of the device testing task. The user of the ATE system combines test elements when creating the test procedure rather than write code from scratch, or modify code of prewritten templates. Accordingly, the user does not need to possess knowledge of a programming language or low-level ATE component details.

22 Claims, 9 Drawing Sheets


130
START
RETRIEVE A TEST ELEMENT FROM THE TEST ELEMENT DATABASE (E.G., ANALOG SOURCE TEST ELEMENT, DIGITAL SOURCE TEST ELEMENT, ANALOG SAMPLE TEST ELEMENT, DIGITAL SAMPLE TEST ELEMENT, ETC.)
132
INDICATE AN OPERATING ORDER FOR THE RETRIEVED TEST ELEMENT RELATIVE TO ANY PREVIOUSLY RETRIEVED TEST ELEMENTS
134
SET PROGRAMMABLE INPUT VARIABLES FOR TEST ELEMENT(S) (E.G., CONFIGURE TEST ELEMENTS TO EXCHANGE RESULT INFORMATION)
136
ADD ANOTHER TEST ELEMENT ?
138
YES
NO
CREATE INSTANCE EDITOR(S) FOR FORMING TEST PROCEDURE INSTANCES
140
END

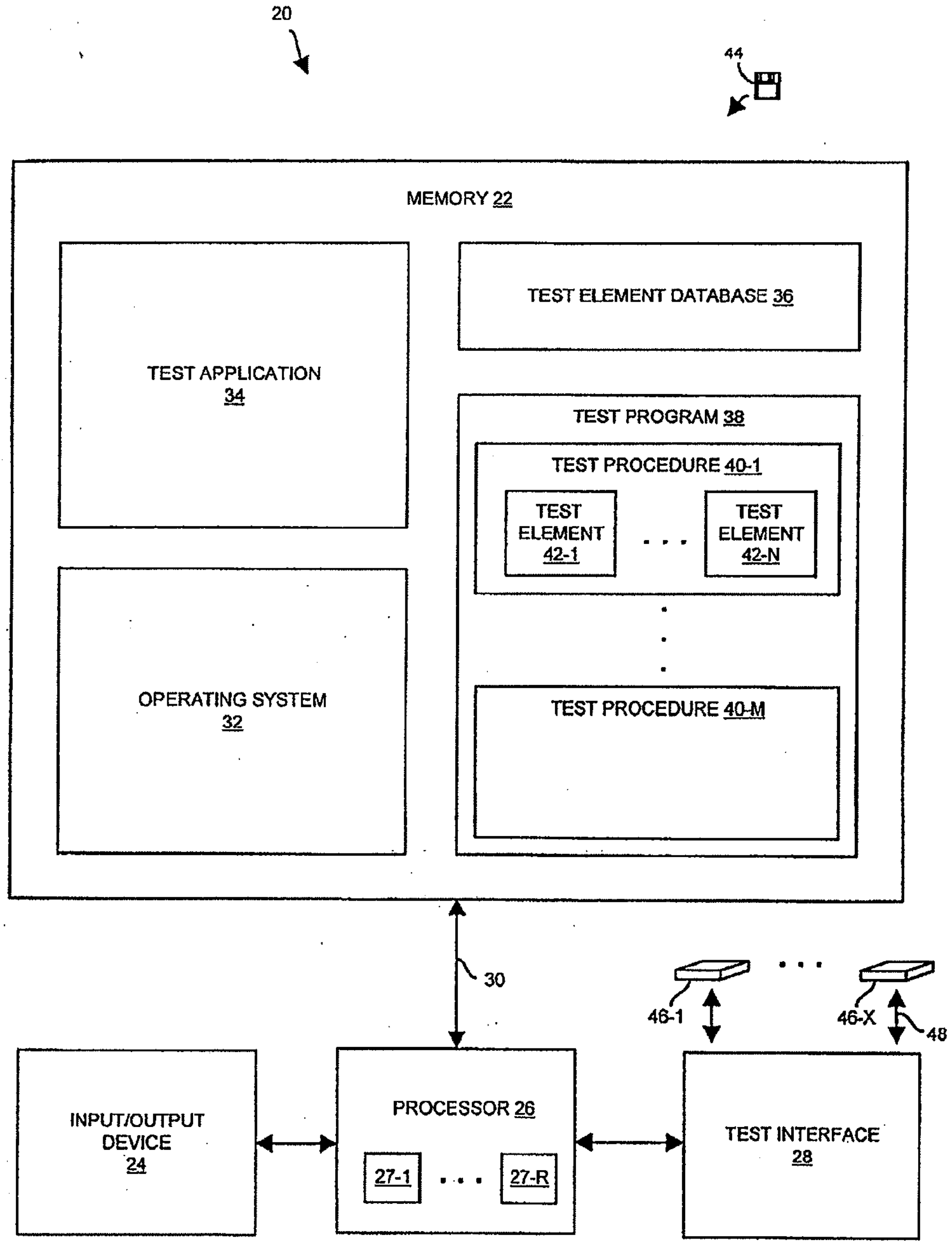


FIG. 1

EXHIBIT A PG. 6

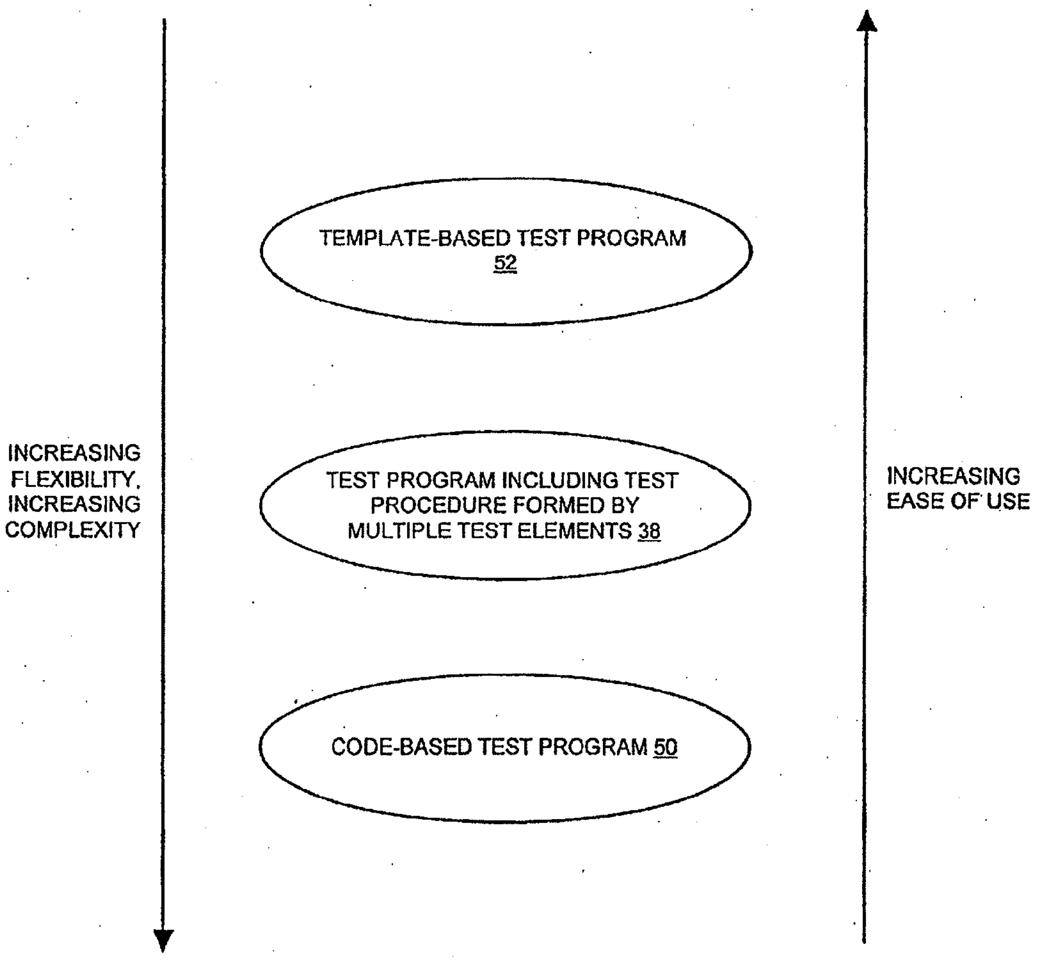

INCREASING FLEXIBILITY, INCREASING COMPLEXITY
TEMPLATE-BASED TEST PROGRAM 52
TEST PROGRAM INCLUDING TEST PROCEDURE FORMED BY MULTIPLE TEST ELEMENTS 38
CODE-BASED TEST PROGRAM 50
INCREASING EASE OF USE

FIG. 2

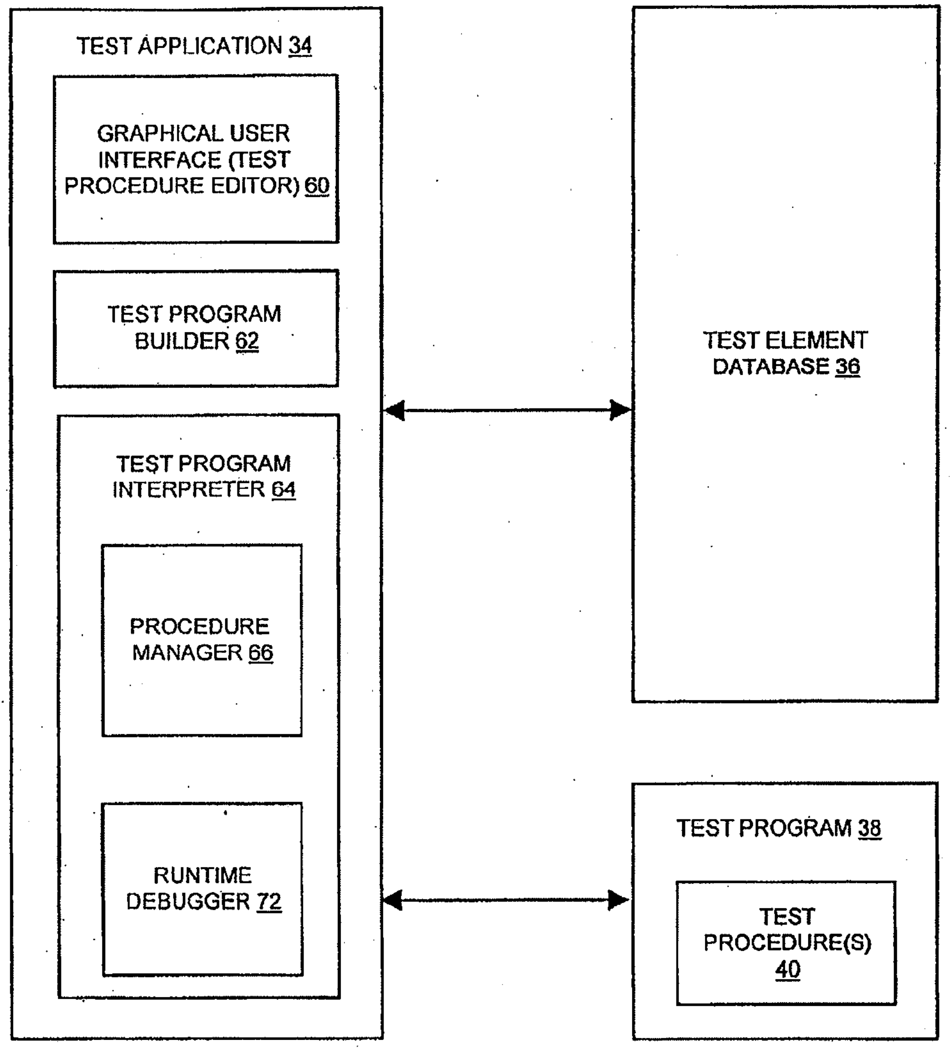
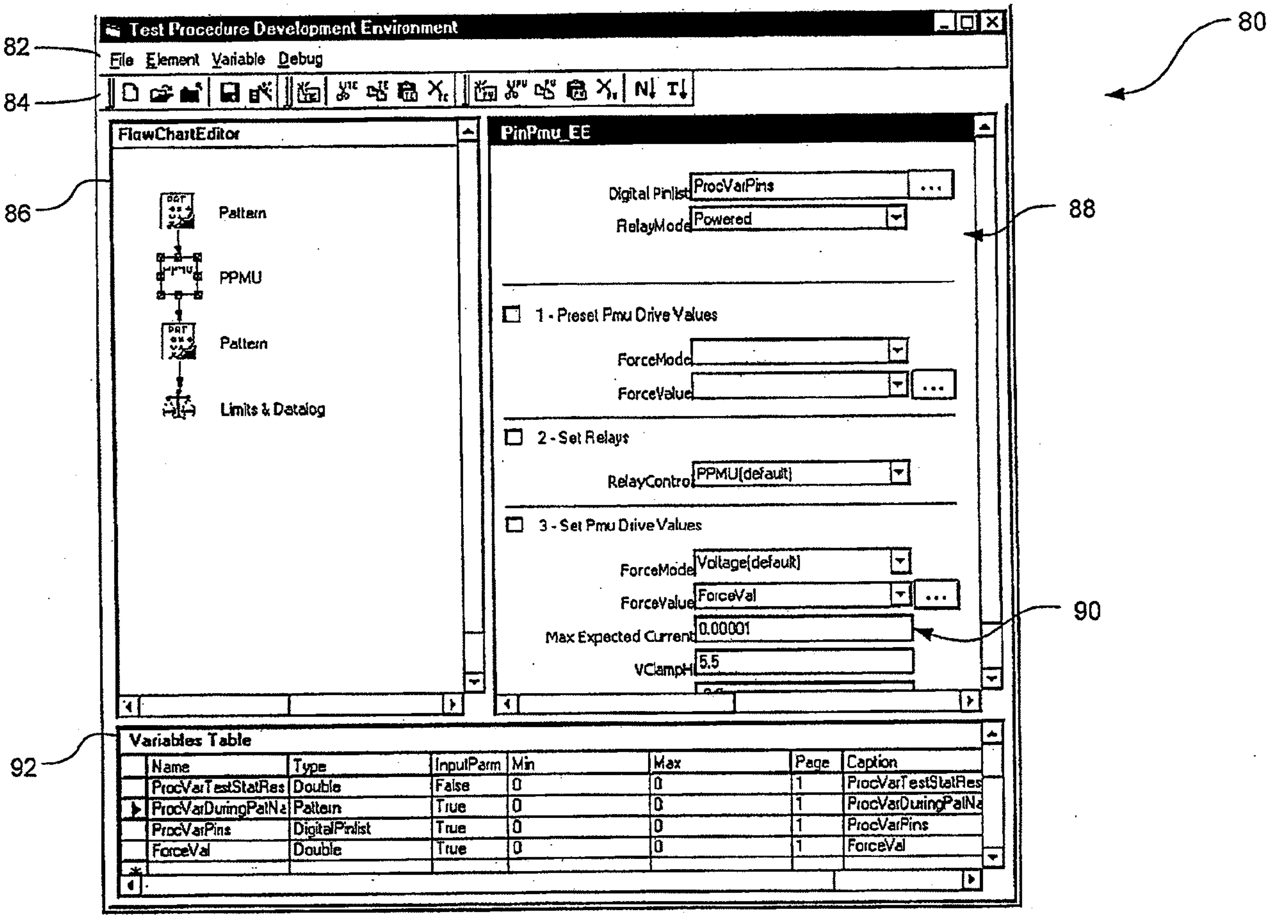

TEST APPLICATION 34
GRAPHICAL USER INTERFACE (TEST PROCEDURE EDITOR) 60
TEST PROGRAM BUILDER 62
TEST PROGRAM INTERPRETER 64
PROCEDURE MANAGER 66
RUNTIME DEBUGGER 72
TEST ELEMENT DATABASE 36
TEST PROGRAM 38
TEST PROCEDURE(S) 40

FIG. 3



Digital Pinlist ProcVarPins

RelayMode Powered

□ 1 - Preset Pmu Drive Values

ForceMode

ForceValue

□ 2 - Set Relays

RelayControl PPMU(default)

□ 3 - Set Pmu Drive Values

ForceMode Voltage(default)

ForceValue ForceVal

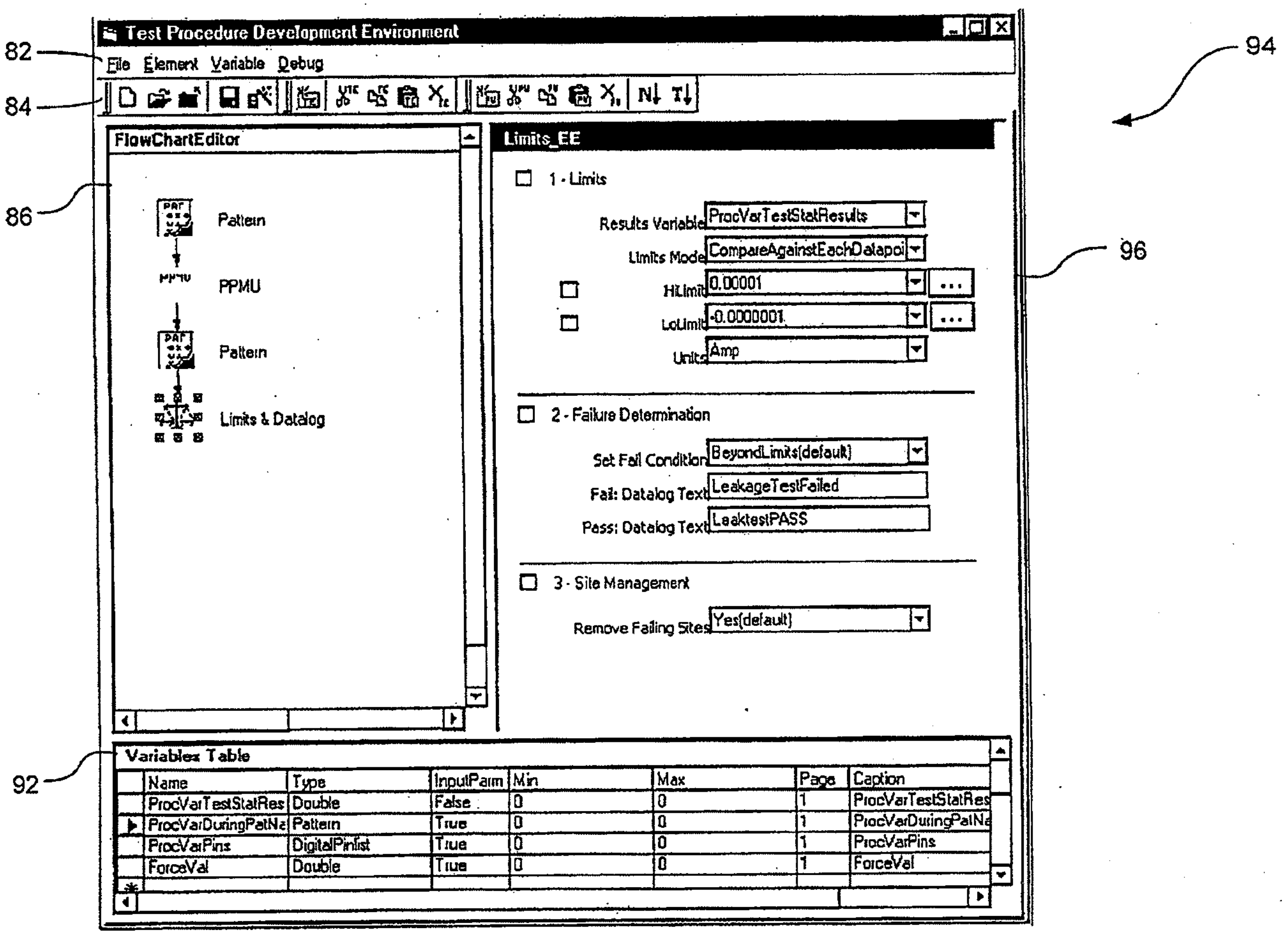
Max Expected Current 0.00001

VClampHi 5.5

Name	Type	InputParm	Min	Max	Page	Caption
ProcVarTestStatRes	Double	False	0	0	1	ProcVarTestStatRes
ProcVarDuringPatNa	Pattern	True	0	0	1	ProcVarDuringPatNa
ProcVarPins	DigitalPinlist	True	0	0	1	ProcVarPins
ForceVal	Double	True	0	0	1	ForceVal

FIG. 4

EXHIBIT A PG. 9



□ 1 - Limits

Results Variable ProcVarTestStatResults

Limits Mode CompareAgainstEachDatapoi

□ HiLimit 0.00001

□ LoLimit -0.0000001

Units Amp

□ 2 - Failure Determination

Set Fail Condition BeyondLimits(default)

Fail: Datalog Text LeakageTestFailed

Pass: Datalog Text LeaktestPASS

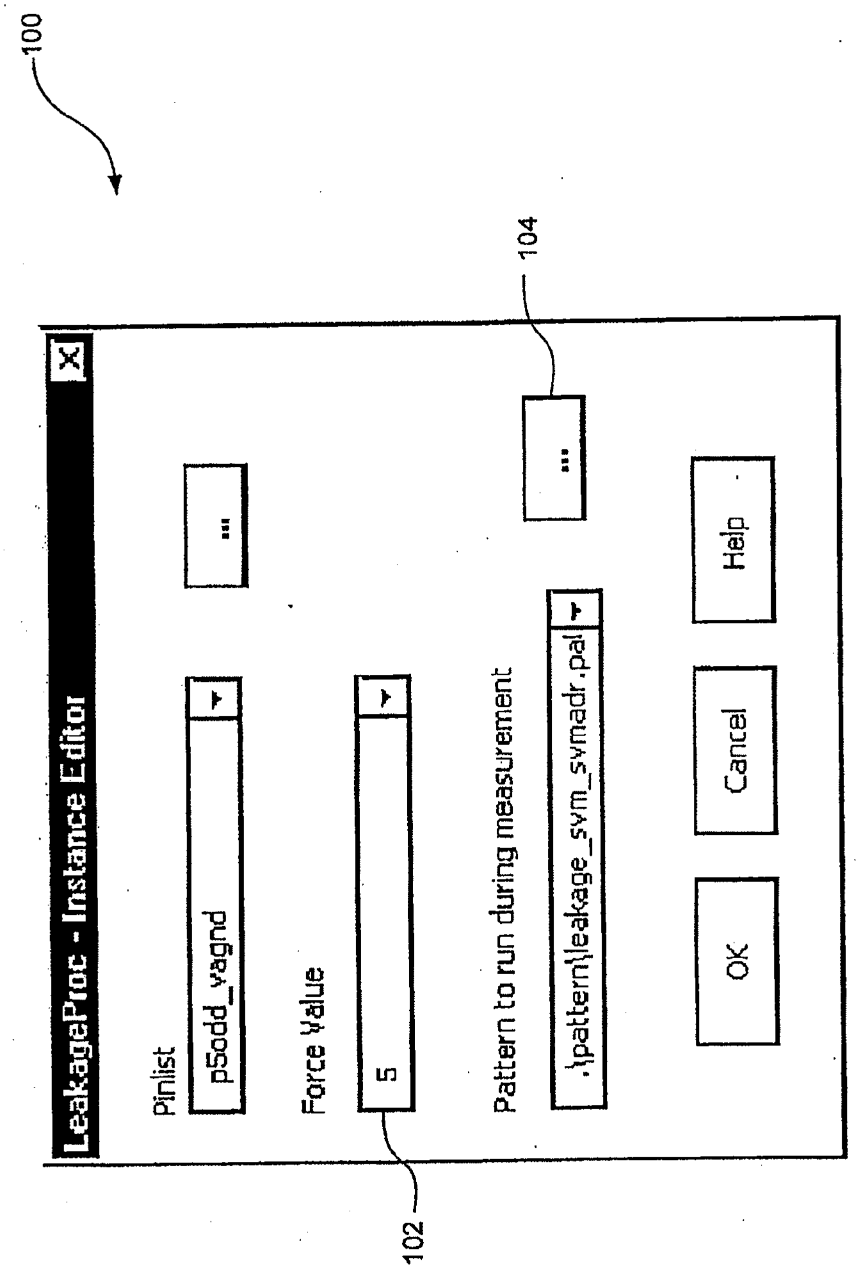
□ 3 - Site Management

Remove Failing Sites Yes(default)

Name	Type	InputParm	Min	Max	Page	Caption
ProcVarTestStatRes	Double	False	0	0	1	ProcVarTestStatRes
ProcVarDuringPatNa	Pattern	True	0	0	1	ProcVarDuringPatNa
ProcVarPins	DigitalPinlist	True	0	0	1	ProcVarPins
ForceVal	Double	True	0	0	1	ForceVal

FIG. 5

EXHIBIT A PG. 10



LeakageProc - Instance Editor

Pinlist

p5odd_vagnd

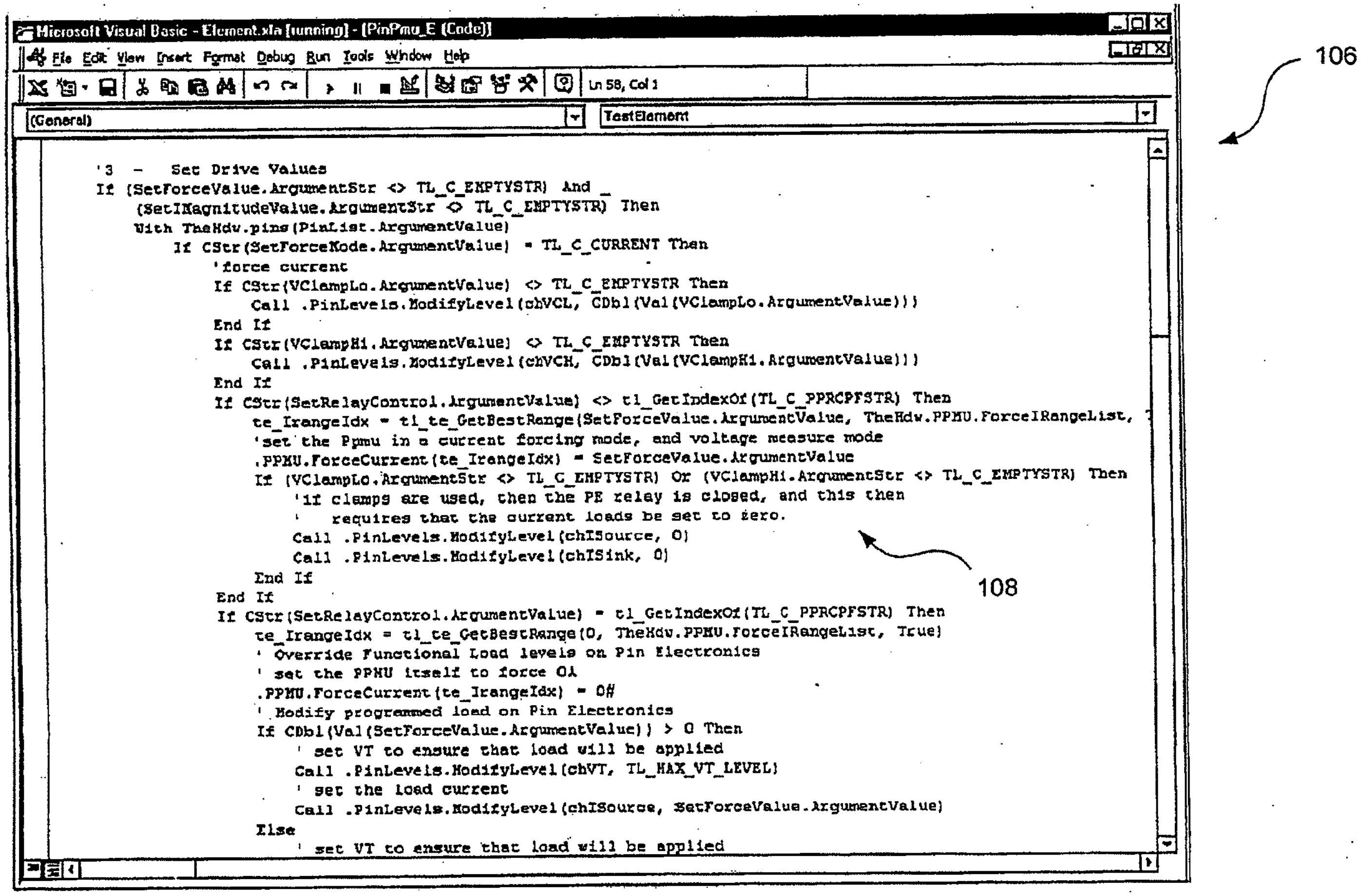
Force Value

5

Pattern to run during measurement

.\pattern\leakage_svm_svmadr.pal

FIG. 6


106
108
Microsoft Visual Basic - Element.xla [running] - [PinPmu_E (Code)]
File Edit View Insert Format Debug Run Tools Window Help
Ln 58, Col 1
(General)
TestElement

```
'3  -   Set Drive Values
If (SetForceValue.ArgumentStr <> TL_C_EMPTYSTR) And _
    (SetIMagnitudeValue.ArgumentStr <> TL_C_EMPTYSTR) Then
    With TheHdw.pins(PinList.ArgumentValue)
        If CStr(SetForceMode.ArgumentValue) = TL_C_CURRENT Then
            'force current
            If CStr(VClampLo.ArgumentValue) <> TL_C_EMPTYSTR Then
                Call .PinLevels.ModifyLevel(chVCL, CDbl(Val(VClampLo.ArgumentValue)))
            End If
            If CStr(VClampHi.ArgumentValue) <> TL_C_EMPTYSTR Then
                Call .PinLevels.ModifyLevel(chVCH, CDbl(Val(VClampHi.ArgumentValue)))
            End If
            If CStr(SetRelayControl.ArgumentValue) <> tl_GetIndexOf(TL_C_PPRCPFSTR) Then
                te_IrangeIdx = tl_te_GetBestRange(SetForceValue.ArgumentValue, TheHdw.PPMU.ForceIRangeList, T
                'set the Ppmu in a current forcing mode, and voltage measure mode
                .PPMU.ForceCurrent(te_IrangeIdx) = SetForceValue.ArgumentValue
                If (VClampLo.ArgumentStr <> TL_C_EMPTYSTR) Or (VClampHi.ArgumentStr <> TL_C_EMPTYSTR) Then
                    'if clamps are used, then the PE relay is closed, and this then
                    '   requires that the current loads be set to zero.
                    Call .PinLevels.ModifyLevel(chISource, 0)
                    Call .PinLevels.ModifyLevel(chISink, 0)
                End If
            End If
            If CStr(SetRelayControl.ArgumentValue) = tl_GetIndexOf(TL_C_PPRCPFSTR) Then
                te_IrangeIdx = tl_te_GetBestRange(0, TheHdw.PPMU.ForceIRangeList, True)
                ' Override Functional Load levels on Pin Electronics
                ' set the PPMU itself to force 0A
                .PPMU.ForceCurrent(te_IrangeIdx) = 0#
                ' Modify programmed load on Pin Electronics
                If CDbl(Val(SetForceValue.ArgumentValue)) > 0 Then
                    ' set VT to ensure that load will be applied
                    Call .PinLevels.ModifyLevel(chVT, TL_MAX_VT_LEVEL)
                    ' set the load current
                    Call .PinLevels.ModifyLevel(chISource, SetForceValue.ArgumentValue)
                Else
                    ' set VT to ensure that load will be applied
```

FIG. 7

EXHIBIT A PG. 12

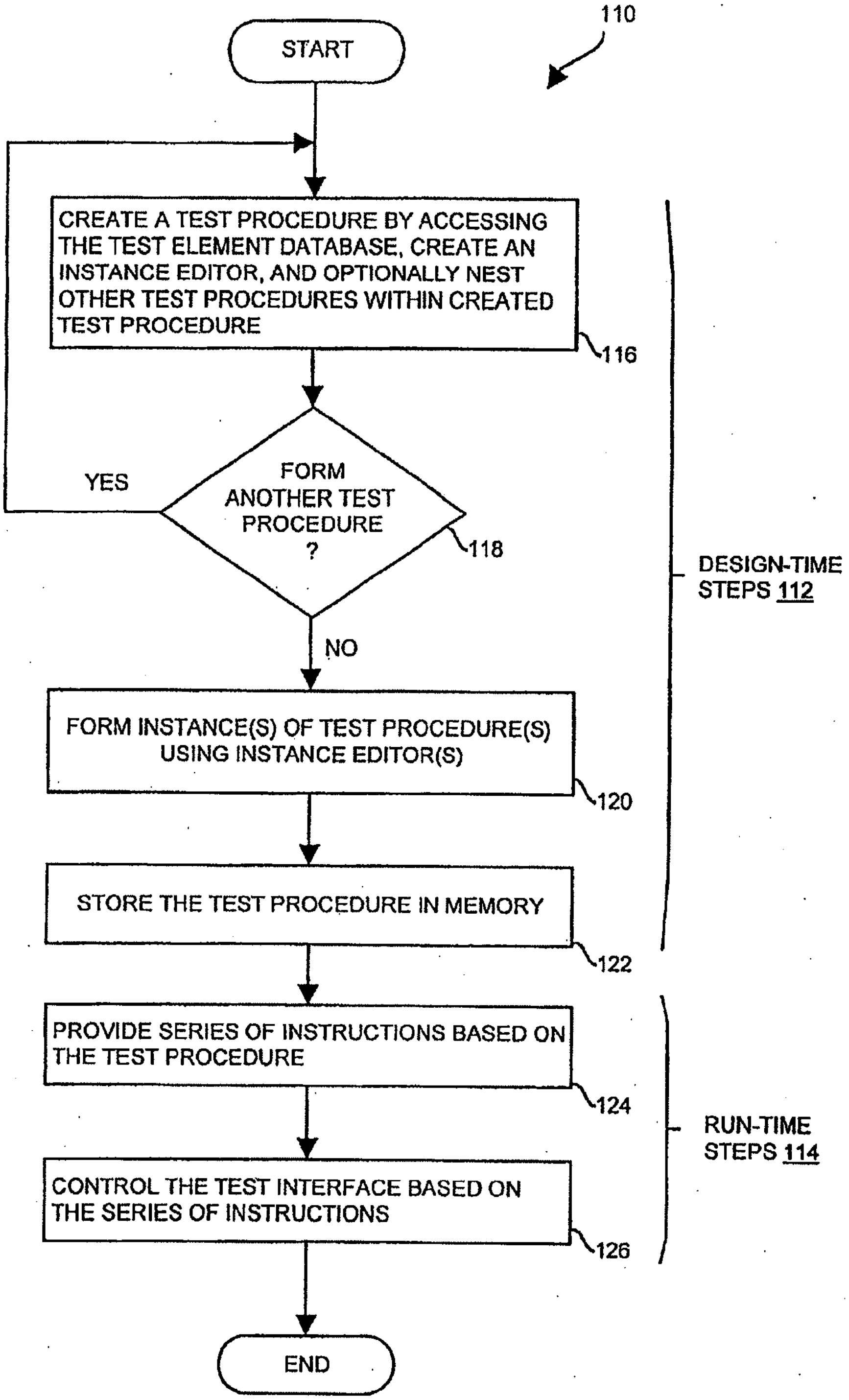

110
START
CREATE A TEST PROCEDURE BY ACCESSING THE TEST ELEMENT DATABASE, CREATE AN INSTANCE EDITOR, AND OPTIONALLY NEST OTHER TEST PROCEDURES WITHIN CREATED TEST PROCEDURE
116
YES
FORM ANOTHER TEST PROCEDURE ?
118
NO
DESIGN-TIME STEPS 112
FORM INSTANCE(S) OF TEST PROCEDURE(S) USING INSTANCE EDITOR(S)
120
STORE THE TEST PROCEDURE IN MEMORY
122
PROVIDE SERIES OF INSTRUCTIONS BASED ON THE TEST PROCEDURE
124
RUN-TIME STEPS 114
CONTROL THE TEST INTERFACE BASED ON THE SERIES OF INSTRUCTIONS
126
END

FIG. 8

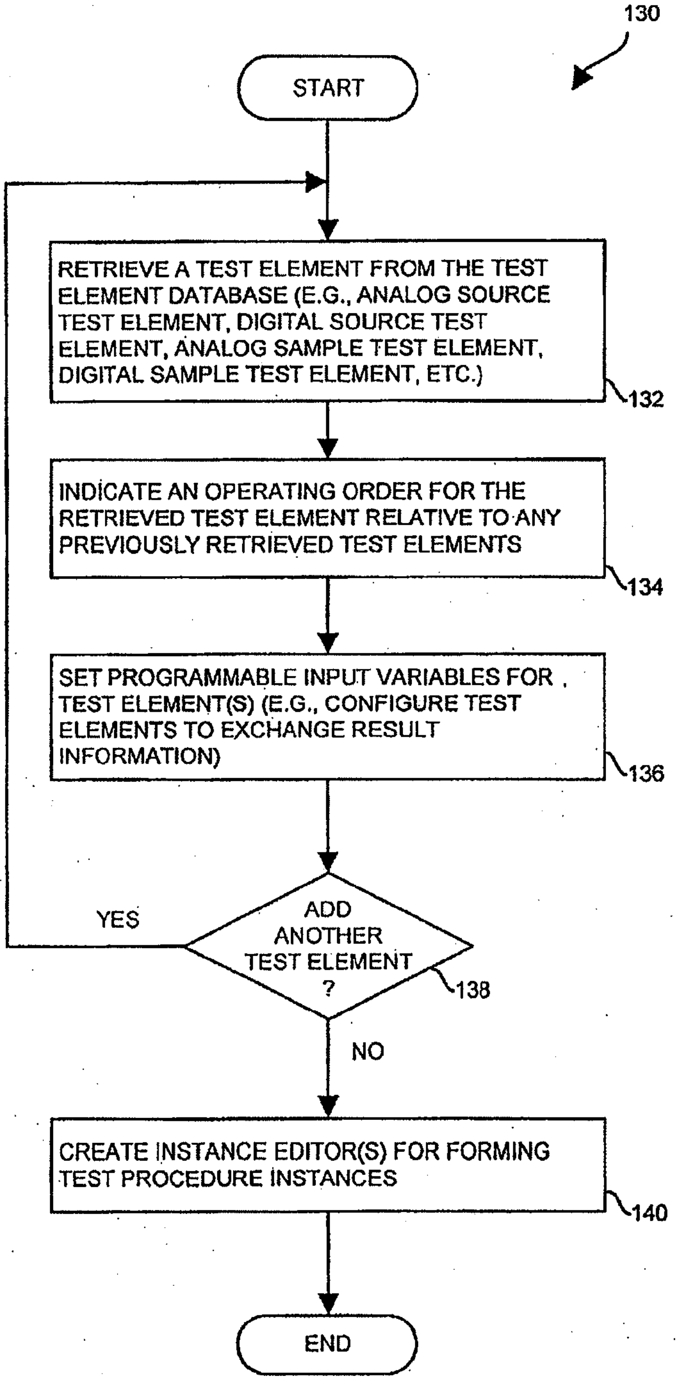

130
START
RETRIEVE A TEST ELEMENT FROM THE TEST ELEMENT DATABASE (E.G., ANALOG SOURCE TEST ELEMENT, DIGITAL SOURCE TEST ELEMENT, ANALOG SAMPLE TEST ELEMENT, DIGITAL SAMPLE TEST ELEMENT, ETC.)
132
INDICATE AN OPERATING ORDER FOR THE RETRIEVED TEST ELEMENT RELATIVE TO ANY PREVIOUSLY RETRIEVED TEST ELEMENTS
134
SET PROGRAMMABLE INPUT VARIABLES FOR TEST ELEMENT(S) (E.G., CONFIGURE TEST ELEMENTS TO EXCHANGE RESULT INFORMATION)
136
YES
ADD ANOTHER TEST ELEMENT ?
138
NO
CREATE INSTANCE EDITOR(S) FOR FORMING TEST PROCEDURE INSTANCES
140
END

FIG. 9

EASY TO PROGRAM AUTOMATIC TEST EQUIPMENT

FIELD OF THE INVENTION

The invention relates generally to automatic test equipment, and more particularly to techniques for controlling such equipment.

BACKGROUND OF THE INVENTION

In general, automatic test equipment (ATE) is equipment for testing devices (or complete systems) in an automated manner. Some ATE systems test electronic circuitry such as integrated circuits (ICs) or circuit boards. When a typical ATE system tests such a device (commonly referred to as the device under test or DUT), the ATE system applies stimuli (e.g., electrical signals) to the device and checks responses (e.g., currents and voltages) of the device. Typically, the end result of a test is either "pass" if the device successfully provides certain expected responses within pre-established tolerances, or "fail" if the device does not provide the expected responses within the pre-established tolerances. More sophisticated ATE systems are capable of evaluating a failed device to potentially determine one or more causes of the failure.

It is common for an ATE system to include a computer that directs the operation of the ATE system. Typically, the computer runs one or more specialized software programs to provide (i) a test development environment and (ii) a device testing environment. In the test development environment, a user typically creates a test program, i.e., a software-based construct of one or more files that controls various portions of the ATE system. In the device testing environment, the user typically provides the ATE system with one or more devices for testing, and directs the ATE system to test each device in accordance with the test program. The user can test additional devices by simply providing the additional devices to the ATE system, and directing the ATE system to test the additional devices in accordance with the test program. Accordingly, the ATE system enables the user to test many devices in a consistent and automated manner based on the test program.

In general, there are two conventional approaches to creating a test program using the test development environment of the ATE system: a code-based approach and a template-based approach. In the code-based approach, the user writes code in a programming language (e.g., the C or C++ programming languages) with extensions and libraries. In general, the code controls low-level operations of various components within the ATE system such as voltage and current sources, measuring devices, and data storage resources. Additionally, if multiple devices are to be tested in parallel, specific code must usually be added to control the additional ATE system components which are required to test the additional devices, substantially increasing program complexity. Accordingly, the code-based approach generally requires the user to posses extensive knowledge of a programming language and the various components of the ATE system. Typically, the developed code is several thousand lines long, and often requires significant time and effort to test and debug before formal device testing can begin.

The code-based approach is powerful and flexible because it allows the user to utilize the resources of the ATE system to their fullest potential. Since the user essentially develops the test program from scratch by writing the code, the user can customize and tailor the test program to take full advantage of the ATE system. For example, the user may include certain optimizations in the test program to operate the ATE system in a manner that provides the quickest test times attainable and thus maximize the capacity of the ATE system. Additionally, the user is not bound to industry testing methods or standards. Rather, the user has the capability to establish and implement user-specific testing methods and standards which, in some respects, may be superior to industry norms. Moreover, the user can test as many different types of devices as appropriate for the ATE system's capabilities. This is particularly important with regard to testing mixed-signal devices which utilize both analog and digital signals, and where device-specific configuration is typically required before devices are ready to accept stimulus signals and provide response signals.

In contrast to the code-based approach, the template-based approach is less burdensome on the user. In the template-based approach, the ATE system manufacturer supplies the user with a selection of test program templates. Each template typically includes code prewritten by the ATE manufacturer thus alleviating the need for the user to write code from scratch. Furthermore, such a template is generally customizable to a degree. For example, a particular test program template may allow the user to enter certain operating parameters that control particular actions performed by the ATE system when testing a device. It is also typical that a test program template can apply the customized operating parameters to multiple devices in parallel and make measurements from multiple devices in parallel. More extensive changes typically require the user to examine and edit the prewritten code if the ATE makes this underlying code accessible to the user. Accordingly, in the template-based approach, the task of the user is to select a template that is well-suited for the particular intended test, and to provide suitable operating parameters to the selected template in order to direct the ATE system to properly perform the intended test.

In general, from the perspective of the user, the template-based approach is relatively easier to use than the code-based approach. In the template-based approach, the user takes advantage of prewritten test programs thus avoiding having to write code from scratch. Accordingly, the user does not require extensive knowledge of a programming language or low-level details of the ATE system. Rather, the user simply selects a prewritten template and sets particular operating parameters for the selected template to customize the operation of the ATE system. As such, the user essentially plays the role of an ATE operator rather than a test program developer.

SUMMARY OF THE INVENTION

Unfortunately, the code-based and template-based approaches to creating a test program have certain drawbacks. In particular, the code-based approach, in which a user writes a test program from scratch, requires the user to posses extensive knowledge of a programming language and low-level details of the various components of the ATE system. Without such knowledge the user would not be able to create a test program that properly operates the various portions of the ATE system to test a device. Furthermore, even with such knowledge, the task of creating the test program is complex and error-prone. Typically, the user invests a significant amount of time developing and debugging a test program before the test program is ready for regular use in device testing.

The template-based approach, in which a user utilizes prewritten test program templates provided by the ATE manufacturer, suffers from certain drawbacks as well. First, this approach is relatively inflexible because the user can only choose from test program templates that the ATE manufacturer has conceived in advance. Such templates often cannot be customized to the extent necessary to test devices which have complex device-specific configuration requirements, such as mixed-signal devices.

Second, some ATE users prefer to establish and implement their own user-specific methods and standards rather than rely on those provided by the ATE manufacturer. In some situations, it is difficult or impossible for such users to implement their own methods and standards by modifying prewritten templates since these templates typically only accept operating parameter changes. More extensive modifications usually entail working at a lower level by searching and editing the underlying template instructions, and verifying that no undesired changes have been made. Modifying and debugging such code can, on occasion, be a more difficult task than writing code from scratch.

Third, as ATE manufacturers attempt to increase the flexibility and customizability of their templates, the templates tend to become more complex from the perspective of the user. In some situations, creating a test program by developing a proficiency in a prewritten template and then modifying that template may be a larger burden on the ATE user, than just writing a test program from scratch. In particular, the user may already possess some knowledge of a programming language and the components of the ATE system. Such a user would now have the burden of having to further learn the details of the manufacturer's test program template and then determine how to modify it.

To avoid making test program templates too complex, ATE manufacturers may attempt to provide a wider selection of test program templates. Unfortunately, this becomes extremely burdensome on the ATE manufacturer, and it is often impossible for the manufacturer to anticipate and satisfy every need of the ATE system user with a test program template. For example, it is unrealistic to expect an ATE manufacturer to provide a separate template for every possible mixed-signal test arrangement (a device test which involves both analog and digital signals) due to the large number of different signal combinations. The manufacturer would likely not be able to complete such an undertaking in a timely manner. Moreover, the ATE system user likely would be overwhelmed by the selection of test program templates, and perhaps may not choose the best-suited test program template for a particular situation. Nevertheless, if the ATE manufacturer does not provide a template which is well-suited for a desired device test (e.g., one that is capable of handling the correct number of analog and digital signals of the intended device under test), the ATE user likely will be unable to effectively test the devices unless the user modifies a less-suitable template. Often such modifications require extensive code changes.

In contrast to the above-described conventional code-based and template-based approaches to creating a test program, the invention is directed to techniques for providing a test procedure from multiple test elements. Each test element defines instructions and programmable input variables for a particular test operation. Preferably, each test element embodies the basic functionality of one of the ATE system's instruments or capabilities. The test procedure defines a device testing task and can operate as a test program, or be combined with other test procedures (or one or more instances of the same test procedure) to form a larger test program. Accordingly, the there is no need to write a test program code from scratch as in the conventional code-based approach. Furthermore, providing a test procedure from test elements generally provides greater flexibility than modifying prewritten test program templates since individual test operations defined by the test elements can be combined in a customized fashion to suit the particular needs of the ATE user.

In one arrangement, the ATE user combines multiple test elements from a test element database to form a test procedure which defines a device testing task. Each test element defines instructions and programmable input variables that direct a processor to perform a particular test operation of the device testing task. The ATE user sets at least a portion of the programmable input variables of each test element forming the test procedure to initial values. Additionally, the ATE user indicates an operating order for the test elements forming the test procedure. Furthermore, the ATE user stores the test procedure within a memory. Accordingly, the ATE user can create a customized test program suitable for testing a specific device such as an IC or a circuit board. Furthermore, the ATE user is not burdened with writing code from scratch and does not need to possess extensive knowledge of a programming language or low-level details of the ATE components as is necessary in the code-based approach.

Additionally, the ATE user has the capability to nest the test procedure within another test procedure. Such a nesting feature enables the ATE user to better organize the test program through modularization. Moreover, the ATE user may be able to reuse certain test procedures on other types of devices that are similar to the device for which the test procedure was originally created.

Preferably, the test element database includes analog signal test elements which define instructions that direct the processor to perform analog signal test operations, digital signal test elements which define instructions that direct the processor to perform digital signal test operations, and other test elements which define instructions for configuring the DUT to receive and transmit stimulus and response signals. The task of combining these analog signal test elements, digital signal test elements, and DUT configuration test elements to form a mixed-signal test procedure or test program is, in many respects, easier than trying to convert a mixed-signal template designed to handle a particular set of signals and DUT configuration requirements, as is sometimes necessary in the conventional template-based approach to effectively test a mixed-signal device.

After the ATE user creates a test procedure from test elements, an ATE system can use the test procedure to test a device. In particular, when the user begins a formal device test, the user enters commands on an input/output (I/O) device of the ATE system to begin a test. In response, a processor of the ATE system obtains the test procedure from memory, and provides a series of instructions based on the test procedure. In response, the processor controls a test interface, which connects (e.g., electrically or mechanically) with the device under test, based on the provided series of instructions in order to test the device.

Preferably, the ATE system provides a graphical user interface (GUI) through the I/O device. The user operates the GUI to create the test procedure from test elements of the test element database. In particular, the user preferably selects graphical icons corresponding to respective test elements, and arranges the icons in an editor window of the GUI to provide an operating order for the test elements. Additionally, the user sets programmable input variables of the test elements to initial values using dialog boxes which vary depending on the operation defined by the test element. The underlying instructions for the test operations which define the test elements is preferably prewritten by the ATE manufacture so that the user can simply manipulate icons and enter parameter values. Hence, the user has the power and flexibility of the code-based approach since the user creates a test program (or test procedure) from scratch, but is does not require extensive knowledge of a programming language or low-level details of ATE components since the user is not writing lines of code from scratch.

It should be understood that the test elements of the test procedure are preferably arranged so that information resulting from one test element of the test procedure can be used by another test element. That is, when the ATE system operates in accordance with the test procedure during a device test, the test operation results from one test element are used by the other test element. For example, one test element can direct the ATE to take measurements from a device under test. A subsequent test element can process these measurements to determine whether the device operates properly.

It should be further understood that the processor preferably includes multiple processing units that are associated with respective multiple devices. In this arrangement, controlling the test interface involves operating each of the multiple processing units based on the provided series of instructions to test each of the respective multiple devices in parallel. This arrangement enables the ATE system to be scaled by increasing the number of processing units within the processor. It should be farther understood that preferably a test procedure and its test elements automatically test as many devices in parallel as are presented to the test interface, in contrast to a conventional code-based test program which normally requires significant modification in order to test different numbers of devices in parallel.

Another arrangement of the invention is directed to a computer program product that includes a computer readable medium having instructions stored thereon for performing the above-described test procedure. The instructions, when processed by a data processing device, cause the data processing device to combine test elements from a test element database to form the test procedure in response to user commands such that the test procedure (i) defines a device testing task, and (ii) includes multiple test elements. The instructions further cause the data processing device to (i) set at least a portion of the programmable input variables of each test element forming the test procedure to initial values, (ii) indicate an operating order for the test elements forming the test procedure, and (iii) store the test procedure within a memory, in response to further user commands.

Another arrangement of the invention is directed to a computer program product that includes a computer readable medium having instructions stored thereon for testing a device. The instructions, when processed by a data processing device, cause the data processing device to obtain a test procedure which defines a device testing task. The test procedure includes multiple test elements defining instructions and programmable input variables that direct the processor to perform particular test operations of the device testing task. The instructions further cause the data processing device to provide a series of instructions based on the test procedure, and control the test interface based on the provided series of instructions in order to test the device.

The features of the invention, as described above, may be employed in an automatic test system and other related devices such as those manufactured by Teradyne, Inc. of Boston, Mass.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

FIG. 1 shows a block diagram of an automatic test equipment system which is suitable for use by the invention.

FIG. 2 shows a diagram illustrating how test programs of the invention and conventional approaches compare with respect to flexibility, complexity and ease of use from a perspective of a user.

FIG. 3 shows a detailed block diagram of a test application of FIG. 1 and its interaction with various components when operating within the automatic test system.

FIG. 4 shows, by way of example, a test program development window of a graphical user interface of FIG. 3.

FIG. 5 shows, by way of example, another test program development window of the graphical user interface of FIG. 3.

FIG. 6 shows, by way of example, yet another test program development window of the graphical user interface of FIG. 3.

FIG. 7 shows, by way of example, yet another test program development window of the graphical user interface of FIG. 3.

FIG. 8 shows a flow diagram of a procedure performed by the automatic test system of FIG. 1 when operating in accordance with the invention.

FIG. 9 shows a flow diagram of a procedure performed by a user of the automatic test equipment system of FIG. 1.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The invention is directed to techniques for providing a test procedure from multiple test elements for use in automatic test equipment (ATE). The test procedure defines a device testing task, and each test element defines instructions and programmable input variables for a particular test operation of the device testing task., Each test element preferably embodies the basic functionality of one of the ATE system's instruments or capabilities. The test procedure may be used as an ATE test program or combined with other test procedures (or instances of the same test procedure) to form a larger ATE test program. Accordingly, such techniques alleviate the need to create ATE test programs by writing code from scratch as in a conventional code-based approach, or by modifying templates as in a conventional template-based approach. Such techniques may be employed in ATE systems and other related devices such as those manufactured by Teradyne, Inc. of Boston, Mass.

FIG. 1 shows an ATE system 20 which is suitable for use by the invention. The ATE system 20 includes memory 22, an input/output (I/O) device 24, a processor 26 having multiple processing units 27, and a test interface 28. The arrows 30 of FIG. 1 represent information (i.e., signals) exchanged between the memory 22, the I/O device 24, the processor 26 and the test interface 28. Another ATE system which is suitable for use by the invention, and which is similar to the ATE system 20 of FIG. 1, is described in U.S. Pat. No. 5,910,895 (Proskauer et. al.), the teachings of which are hereby incorporated by reference in their entirety.

As shown in FIG. 1, the memory 22 stores various constructs including, among other things, an operating system 32, a test application 34, a test element database 36, and a test program 38. The test program 38 includes test procedures 40-1, . . . , 40-M (collectively, test procedures 40). Each of the test procedures 40 includes multiple test elements 42. For example, test procedure 40-1 includes test elements 42-1, . . . , 42-N.

One or more of the above-listed constructs in the memory 22 can be provided to the memory 22 through a computer program product 44 (e.g., one or more diskettes or tapes). Preferably, the test application 34 and the test element database 36 are provided by the manufacturer of the ATE system 20 to an ATE end-user via such a computer program product 44. Alternatively, these constructs can be provided to the memory 22 via other mechanisms (e.g., downloaded through a network interface, not shown) or created by an ATE user operating the ATE system 20.

The ATE system 20 operates in accordance with the operating system 32 and the test application 34 (i.e., one or more software programs coordinated by the operating system 32) to provide a user with a test development environment and a device testing environment. The test application 34 provides a graphical user interface (GUI) on the I/O device 24 enabling a user (i) to create the test program 38 (i.e., one or more test procedures 40) when operating within the test development environment, and (ii) to test devices 46-1, . . . , 46-X (collectively, devices 46) in accordance with the test program 38 when operating within the device testing environment.

As described above, each test procedure 40 includes multiple test elements 42. Preferably, each test element 42 is a modularized, prewritten grouping of instructions and variables provided by the manufacturer of the ATE system 20. Initially, these groupings are centrally stored in the test element database 36.

In one arrangement, when a user creates a test procedure 40 from multiple test elements 42, the user copies the actual groupings of instructions and variables for the multiple test elements 42 from the test element database 36 into a location of the memory 22 designated for storing the test procedure 40. In this arrangement, the test procedures 40 are essentially self-contained in that each test procedure 40 includes all of the instructions and variables necessary for directing the ATE system 20 to perform a particular device testing task.

In another arrangement, when the user creates a test procedure 40 from multiple test elements, the user adds test element references and variable values (represented as test elements 42 shown in FIG. 1) to the test procedure 40. The test element references are essentially pointers to the test element instructions and variables which remain centrally stored in the test element database 36. In this arrangement, at device testing time, the ATE system 20 retrieves the instructions and variables for the test elements 42 based on the test element references, and uses the earlier-provided variable values as initial values for the variables. This arrangement uses less memory space than the earlier-described self-contained arrangement. Furthermore, when revisions and improvements are made to a particular test element 42, i.e., when a change is made to the instructions of variables of the particular test element 42, each test procedure 40 referencing that test 42 is automatically capable of accessing the change.

It should be understood that, in either the self-contained arrangement or the test element reference arrangement, the test procedures 40 are considered to include the test elements 42 since the groupings of instructions and variables for each test element 42 is retrieved from the test element database 36 at some point in both arrangements. In the self-contained arrangement, the groupings are retrieved at test procedure creation time. In the test element reference arrangement, the groupings are retrieved at device testing time.

It should be further understood that the invention overcomes drawbacks of the conventional code-based and template-based approaches to creating a test program. In particular, the user of the code-based approach needs to write code from scratch to create a test program. Such code writing is unnecessary with the invention. Rather, the user of the invention combines test elements (i.e., prewritten groupings of instructions and variables) to create a test program 38. The test program 38 can be just as powerful and flexible as one created using the code-based approach since the user of the invention can select test elements 42 best-suited for performing particular test operations to form a custom-tailored test procedure 40 for performing a particular device testing task. Furthermore, multiple test procedures 40 can be combined to perform multiple device testing tasks for a device test.

Furthermore, it should be understood that the user of the template-based approach often needs to edit underlying prewritten code to customize a template for an effective device test. Such an undertaking is often difficult if the user is not knowledgeable in the programming language of the template. Such a modification is unnecessary with the invention. Rather, the user of the invention can combine suitable test elements 42 (preferably, by arranging graphical icons on a GUI, as will be explained shortly) and providing variable values to create a test program 38. Each of these test elements 42 can test different numbers of devices in parallel without modification. Accordingly, the user does not need to wrestle with any underlying code.

FIG. 2 illustrates a general comparison, from the perspective of a user, of the test program 38 of the invention, a typical test program 50 created using the conventional code-based approach, and a typical test program 52 created using the conventional template-based approach. As shown in FIG. 2, the test program 38 of the invention is generally more flexible than the template-based test program 54. In particular, with the ATE system 20 of the invention, the user can test as many different types of devices as appropriate for the ATE system's capabilities. The user is not limited by the prewritten temple choices provided by the ATE manufacturer as in the template-based approach.

Furthermore, a user creating the test program 38 of the invention can simply combine test elements 42 from the test element database 36. No code modifications are necessary to create the test program 38. In contrast, in the template-based approach, code modifications are often required for an effective device test if no well-suited templates are provided by the ATE manufacturer. Unfortunately, extensive code modifications are generally difficult and require significant testing and debugging before regular testing can begin since the user has to locate and rewrite portions of the template without corrupting the operation of other related or adjacent template portions.

Additionally, as shown in FIG. 2, from the perspective of the user, the test program 38 of the invention is generally less complicated to create than the code-based test program 50. To create the test program 38 of the invention, the user works at a level above actual code, and combines test elements 42 (prewritten groupings of instructions and variables) and provides initial values to form a test procedure 40 or multiple test procedures 40 for the test program 38. Knowledge of a programming language and the low-level details of the ATE components are not required to create the test program 38. On the other hand, in the code-based approach, the user must possess knowledge of a programming language and low-level details of the various components of the ATE system. Additionally, once a user creates the test program 50 using the code-based approach, the user generally must spend considerable time and effort testing and debugging code statements within the test program 50.

Furthermore, it is likely that the code-based test program 50 is not as well organized as the test program 38 of the invention since any organizational controls over a user-written test program result from self-imposed controls. That is, there is no guarantee that the code-based test program 50 will be modularized, well-organized or easy to debug. On the other hand, the test program 38 of the invention naturally takes a modular structure due to the arrangements of test elements 42 forming test procedures 40, and one or more test procedures 40 forming a test program 38. Moreover, if the test program 38 uses references to test elements 42 within the test element database 36 and if a revision or improvement is made to a particular test element 42, the ATE system 20 propagates that change to the test program 38 of the invention and every other test program 38 and test procedure 40 using that test element 42. Hence, in general, the test program 38 of the invention is less complex from the user's perspective than the code-based test program 50.

Referring again to FIG. 2, from the perspective of the user, the test program 38 of the invention is generally easier to use than the code-based test program 50. The test program 38 of the invention utilizes a standardized interface (e.g., a GUI) regardless of the particular device test. Moreover, in one arrangement, the test program 38 is controlled and launched from a common standard spreadsheet program. One spreadsheet program that is suitable for such use is Microsoft Excel, which is manufactured by Microsoft Corporation of Redmond, Wash. Accordingly, an operator of the ATE system 20 does not need to learn a new interface for every different device test.

On the other hand, the code-based test program 50 typically uses a unique user interface since the code-based test program 50 is handwritten exclusively for testing a particular device. Accordingly, an ATE test operator must learn a new interface for each new code-based test program 50. Hence, the code-based test program 50 is generally less easy to use than the test program 38 of the invention.

Further details of the invention will now be provided with reference to FIG. 3. The test application 34 has a number of components including, among others, a graphical user interface (GUI) 60, a test program builder 62 and a test program interpreter 64. The test program interpreter 64 includes, among other things, a program manager 66 and a runtime debugger 72.

Although the following description of the test application components may describe the components as performing particular operations for simplicity, it should be understood that the processor 26 actually performs these operations under direction of the components. That is, each test application component directs the processor 26, perhaps in conjunction with one or more other portions of the ATE system 20, to perform such operations.

The GUI 60 operates (through the I/O device 24) as a front-end for the test application 34. That is, the GUI 60 prompts the user of the ATE 20 for commands and data, and provides the commands and data to the test program builder 62 and the test program interpreter 64. Additionally, the GUI 60 receives responses to the user commands and data from the test program builder 62 and the test program interpreter 64, and displays the responses as feedback to the user on the I/O device 24.

The test program builder 62 is responsible for building the test program 38 in the test development environment. As shown in FIG. 3, the test program builder 62 accesses the test element database 36 to build the test program 38. In particular, commands provided by the user through the GUI 60 direct the test program builder 62 to combine and arrange test elements to create one or more test procedures 40. An instance of each test procedure 40 can operate as a test program 38 itself, or combine with other test procedures 40 or instances of the same test procedure 40 to form a larger test program 38.

The test program interpreter 64 is responsible for device testing based on the test program 38 in the device testing environment. In particular, the program manager 66 interprets test element instructions of the test elements forming the test procedures 40 of the test program 38, and performs the low-level operations needed to perform the device test. In particular, the test program interpreter 64 transforms the test element parameters into small sequences of test actions and coordinates operation of the processing units 27. The user can remain ignorant of these low-level details. Further details of the operation of the various test application components will be provided shortly.

FIG. 4 shows, by way of example, a window 80 of the test development environment that is suitable for use by the ATE system 20. The window 80 includes menu 82 and toolbar 84 which operate in a conventional manner to enable a user to enter commands to perform particular tasks (e.g., to create or store a test procedure).

The window 80 further has a number of subwindows. In particular, the window 80 includes a test procedure editor subwindow 86 that graphically displays (e.g., in a top-down iconic format) an arrangement order for multiple test elements 42 of the current test procedure 40. In the example, the current test procedure 40 begins with a "Pattern" test element as shown by the "Pattern" icon, which is followed by a "PPMU" test element as shown the "PPMU" icon, and so on. The test element order can be modified by adding, inserting, or deleting test element icons. Furthermore, the test element order can be modified by changing arrows connecting the test element icons to provide loops or branches in the flow. Accordingly, the user can construct complex, conditional procedure flows to accomplish complex testing tasks.

The window 80 further includes an input value subwindow 88 which prompts the user to input values and/or variable names required by a particular test element 42. As shown in FIG. 4, by way of example, the "PPMU" test element is selected in the test procedure editor subwindow 86 as shown by the boxed outline around the "PPMU" icon, and the subwindow 88 prompts the user to enter required values for "PPMU" test element. By way of example only, the user has entered (e.g., via a keyboard or mouse) a value of "0.00001" in a field 90 which corresponds to a "Max Expected Current" variable of the "PPMU" test element.

The window 80 further includes a variable table subwindow 92 which provides the user with an overview of the variables used by the test procedure 40, and their properties. The user can change particular property values used by the test procedure 40 by simply selecting a field of the variable table subwindow 92 (e.g., using a mouse of the I/O device 22), and entering a new property value in that field.

FIG. 5 shows, by way of example only, how the window 80 of the test development environment can change based on input from the user. In particular, suppose that the user selects another test element of the flow within the test procedure editor subwindow 86 of the window 80 (also see FIG. 4). In response, the user will see window 94 on the I/O device 22, as shown in FIG. 5. Window 94 is similar to window 80, except that window 94 includes a modified test procedure editor subwindow 86' which shows the icon of the newly selected test element highlighted or boxed (i.e., the "Limits & Datalog" icon), rather than the earlier selected icon. Furthermore, the subwindow 88 of window 80 is replaced with a new subwindow 96 corresponding to the newly selected test element.

The above-described window-based presentation arrangement makes the use of test elements superior to general purpose programming tools. For example, the test element specific windows 88 and 96 indicate to the user what information is needed. Accordingly, the user does not need to consult reference material or posses extensive experience with the ATE system 20.

When the user is finished adding test elements 42 to the test procedure 40, the user may wish to set certain variables to initial values. For example, the user may wish to perform one test an a device using a voltage supply of 3.5 volts, and then repeat the test on the same device using a voltage supply of 5.0 volts. To accommodate the user, the ATE system 20 allows the user to create multiple instances of the test procedure 40 (one for each test). To this end, the user enters commands into the GUI 60, and the ATE system 20 prompts the user to create multiple instances of the test procedure 40, distinguishing each instance by a unique identifier (e.g., an instance name or number). FIG. 6 shows, by way of example only, an instance editor window 100 which prompts the user to enter values for a particular instance of the test procedure 40. Accordingly, the user can develop the test program 38 so that it performs variations of the same test on the same device using multiple instances of the same test procedure 40.

It should be understood that the ATE system 20 of the invention is easy to use when creating the test procedure 40 since the user simply retrieves test elements (i.e., test element icons representing the desired test elements) from the test element database 36, and arranges them and initializes their input variables. The user does not need to write or modify any code as in conventional approaches to creating a test program. Nevertheless, the ATE system 20 has the capability to show the instructions and variables defined by a test element 42 if the user wishes to see them. For example, the user can direct the ATE system 20 to display the instructions and variables using the menu 82 or toolbar 84. As shown in FIG. 7, the ATE system 20 responds by showing the user a window 106 having the instructions and variables 108 defined by a selected test element 42.

Further details of the operation of the ATE system 20 will now be provided with reference to FIGS. 8 and 9. FIG. 8 shows a flow diagram of a procedure 110 performed by the ATE system 20. The procedure includes design-time steps 112 which are performed when the ATE system 20 operates in the test development environment, and run-time steps 114 which are performed when the ATE system operates in the device testing environment.

In step 116, the ATE system 20 accesses the test element database 36 to create a test procedure 40. In particular, as shown in FIGS. 4 through 6, the GUI 60 and the test procedure builder 62 enables the user to combine test elements 42 from the test element database 36, set values for their variables, and order the test elements relative to each other to create the test procedure 40. Optionally, the user may include one or more previously created test procedures 40 in the test procedure 40 presently being created.

In step 118, the ATE system 20 (i.e., the GUI 60 and test procedure builder 62) provides the user with the choice of creating additional test procedures. If the user directs the ATE system 20 to create another test procedure 40, step 118 proceeds back to step 116. Otherwise, step 118 proceeds to step 120.

In step 120, the ATE system 20 forms an instance of the earlier-created test procedure 40. The user provides input into the instance editor window 100 (see FIG. 6) to set initial values for use during device testing.

In step 122, the ATE system 20 stores, in the memory 22, the test procedure instance 40. Hereinafter, the test procedure instance 40 will be referred to as the test procedure 40 for simplicity. It should be understood that the ATE system 20 can handle multiple instances of the test procedure 40. Step 122 then proceeds to step 124 (the beginning of the device testing environment).

In step 124, the ATE system 20 provides a series of instructions based on the test procedure 40. In one arrangement where the test procedure 40 includes references to test element instructions stored within the test element database 36, the program manager 66 of the test procedure interpreter 64 retrieves the instructions from the test element database 36. In another arrangement where the test procedure 40 includes actual instructions and variables for each test element 42 forming the test procedure 40, the program manager 66 obtains the instructions by simply reading portions of the test procedure 40 from the memory 22. In both arrangements, the program manager 66 transforms the series of instruction into a sequence of operations which are performed by each of the processing units 27 of the processor 26.

In step 126, each processing unit 27 of the processor controls the test interface 28 to test a respective device 46. For example, the processing units 27 may apply or measure signals 48 of the devices 46. The simultaneous operation of the processing units 27 enables the ATE system 20 to test multiple devices in parallel automatically.

Further details of the activities of the user, when creating the test procedure 40, will now be provided with reference to FIG. 9. When the user operates the ATE system 20 to create the test procedure 40, the user performs a procedure 130 using the GUI 60 and the test program builder 62 of the test application 34.

In step 132, the user retrieves a test element from the test element database 36. For example, for a mixed-signal device, the user may select one of an analog source test element, a digital source test element, an analog sample test element, a digital sample test element, etc.

In step 134, the user provides information about the order of operation of the retrieved test element relative to any other test elements that have been added to the test procedure 40. Preferably, the user arranges test element icons within a test procedure editor window of the GUI 60 (e.g., see subwindow 6 of FIG. 4). For example, the user may order the most recently retrieved test element after a previously retrieved test element.

In step 136, the user sets programmable input variables for any of the test elements 42 of the test procedure 40. For example, the user may enter variable names for the most recently retrieved test element 42 using a subwindow of the GUI 60 (e.g., see subwindow 88 of FIG. 4). Furthermore, the user may configure a previously retrieved test element 42 to provide results into the most recently retrieved test element 42, for example by using the procedure variables defined in the variable table subwindow 92. This configures the test elements 42 to pass information among themselves automatically. Preferably, this is done with procedure variables which hold multiple values, one for each DUT to be tested in parallel.

In step 138, the user determines whether to add another test element 42 to the test procedure 40. If the user wishes to add another test element 42, the user repeats steps 132 through 136. Otherwise, the user proceeds to step 140.

In step 140, the user creates an instance editor for each test procedure 40. In particular, in response to user input through the GUI 60, the test program builder 62 analyzes the test procedure 40 to determine which programmable input variables of the test elements 42 require initial values prior to device testing. In one arrangement, the test program builder 62 examines information used within the variable table subwindow 92 (see FIGS. 4 and 5) to identify the programmable input variables that require initial values. The ATE system 20 then creates the instance editor window 100 to prompt the user to provide the required initial values (see FIG. 6). The user can then create one or more instances of the test procedure 40 for device testing (see step 120 in FIG. 8). Since the instance editor prompts the user to provide the initial values needed by the test procedure 40 for device testing, the user does not need to look up such requirements in manuals or rely on prior experience using the test procedure 40.

It should be understood from the above-provided description that the techniques of the invention enable a user to create a test procedure 40 for use as a test program 38 itself, or for use with other test procedures 40 (or multiple instances of the same test procedure) to form a larger test program 38 without drawbacks of conventional approaches to creating a test program. In particular, the user combines test elements which define groupings of instructions and variables into an ordered arrangement which directs the ATE system 20 to perform a device test on one or more devices 46. The ATE system 20 is easy to use, does not require expert knowledge of the low-level ATE instruments and components or a programming language, and is flexible so that the user can direct how he or she wants to use the ATE system's resources. The features of the invention may be particularly useful in computerized devices manufactured by Teradyne, Inc. of Boston, Mass.

EQUIVALENTS

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.

For example, the I/O device 22, the memory 22 and the processor 24 may form portions of a general purpose computer (e.g., mainframe, workstation, distributed computing system, etc.), or form a specialized ATE device. The memory 22 may include both primary memory (e.g., semiconductor memory), secondary memory (e.g., disk or tape memory), other storage media (e.g., CDROM, network servers, etc.), or a combination different memory types. Additionally, the I/O device 24 may include multiple input and output devices such as one or more displays, keyboards, printers, mice, etc.

Furthermore, although the processor 26 is shown to include multiple processing units 27, the processor 26 may be implemented as a single processor that multiplexes time to test each device 46 connected to the test interface 28 at any given time. Alternatively, the processor 26 may have a multiprocessor configuration or a distributed processing configuration which controls the multiple processing units 27 respectively associated with the multiple devices 46. Various interconnection mechanisms (e.g., a computer bus, backplane, etc.) are suitable for handling the information exchange 30 between the various components of the ATE system 20.

In the context of testing ICs or circuit boards, the test interface 28 may include probes or pins for making electrical contact with the devices 46. In other contexts, the test interface 28 may include a variety of devices specific to those contexts. For example, to mechanically test a device, the test interface may include force application and measuring devices.

Additionally, it should be understood that the ATE system 20 does not necessarily require each test procedure 40 to include multiple test elements 42. Rather, a test procedure 40 can include a single test element 42. However, the test procedures 40 preferably include multiple test elements 42 due to their naturally provided organizational benefits (i.e., modularizing instructions and variables into groupings which perform a particular test operation).

Furthermore, it should be understood that the user can disable one or more of the processing units 27 of the processor when the ATE system 20 operates at less than full capacity. For example, if the user wishes to test a single device 46, the user can configure the ATE system 20 to operate only one processing unit 27 to test that device 46.

What is claimed is:

1. A system for testing a device, comprising:

memory having a test application stored therein;

a test interface to connect to a device; and

a processor coupled to the memory and the test interface, the processor being configured to operate in accordance with the test application to:

i) provide a series of instructions based on a test procedure defining a device testing task, the test procedure including multiple test elements, each test element defining instructions and programmable input variables that direct the processor to perform a particular test operation of the device testing task, and

ii) control the test interface based on the provided series of instructions in order to test the device,

wherein each test element defines instructions for only a partial device testing task such that the multiple test elements together form a complete device testing task for testing the device.

2. The system of claim 1, further comprising:

an input/output device coupled to the processor, wherein the processor is further configured to operate in accordance with the test application to provide a graphical user interface on the input/output device through which a user directs the processor to:

(i) combine test elements from a test element database to form the test procedure;

(ii) set at least a portion of the programmable input variables of each test element forming the test procedure to initial values;

(iii) indicate an operating order for the test elements forming the test procedure; and

(iv) store the test procedure within the memory.

3. A system for testing a device, comprising:

memory having a test application stored therein;

a test interface to connect to a device; and

a processor coupled to the memory and the test interface, the processor being configured to operate in accordance with the test application to:

i) provide a series of instructions based on a test procedure defining a device testing task, the test procedure including multiple test elements, each test element defining instructions and programmable input variables that direct the processor to perform a particular test operation of the device testing task, and

ii) control the test interface based on the provided series of instructions in order to test the device,

wherein the test procedure includes:

a first test element which defines instructions directing the processor to perform a first test operation that provides a first result; and

a second test element which defines a second set of instructions directing the processor to perform a second test operation that provides a second result which is based on the first result.

4. The system of claim 1 wherein the test procedure is a nested test procedure that is nested within another test procedure such that the processor provides instructions to perform the device testing task when providing instructions based on the other test procedure.

5. The system of claim 1 wherein the processor includes multiple processing units that are associated with respective multiple devices, and wherein each of the multiple processing units is configured to control a respective portion of the test interface based on the provided series of instructions to test the respective multiple devices in parallel.

6. The system of claim 1 wherein the processor includes multiple processing units that are associated with respective multiple devices, and wherein the multiple processing units are configured to control respective portions of the test interface based on instructions defined by a single instance of the test procedure to test the respective multiple devices in parallel.

7. The system of claim 1 wherein the device is a mixed signal device, and wherein the test procedure includes:

a first test element which defines instructions directing the processor to perform an analog signal test operation; and

a second test element which defines instructions directing the processor to perform a digital signal test operation.

8. The system of claim 1 wherein the processor is further configured to:

analyze the test procedure to identify which programmable input variables of the test elements of the test procedure require initial values; and

create a graphical user interface component which prompts a user to provide the required initial values to initialize the identified programmable input variables.

9. A method for testing a device, comprising the steps of:

obtaining a test procedure which defines a device testing task, the test procedure including multiple test elements, each test element defining instructions and programmable input variables that direct a processor to perform a particular test operation of the device testing task;

providing a series of instructions based on the test procedure; and

controlling a test interface based on the provided series of instructions in order to test the device, wherein the step of controlling the test interface includes the steps of:

based on a first test element, performing a first test operation that generates a first test result, and

based on a second test element, performing a second test operation that (i) obtains the first test result generated by the first test operation performed by the processor under direction of the first test element, and (ii) generates a second test result based on the first test result.

10. The method of claim 9, further comprising the steps of:

providing a graphical user interface on an input/output device through which a user directs the processor to:

(i) combine test elements from a test element database to form the test procedure;

(ii) set at least a portion of the programmable input variables of each test element forming the test procedure to initial values;

(iii) indicate an operating order for the test elements forming the test procedure; and

(iv) store the test procedure within a memory.

11. The method of claim 9 wherein the step of providing includes the steps of:

providing instructions directing the processor to perform the first test operation that generates the first test result based on the first test element of the test procedure; and

providing instructions directing the processor to perform the second test operation that generates the second test result based on the second test element of the test procedure, the second test result being based on the first test result.

12. The method of claim 9 wherein the test procedure is a nested test procedure that is nested within another test procedure, and wherein the step of providing includes the step of:

providing instructions directing the processor to perform the device testing task when providing instructions based on the other test procedure.

13. The method of claim 9 wherein the test interface includes multiple processing units that are associated with respective multiple devices, and wherein the step of controlling the test interface includes the steps of:

operating each of the multiple processing units of the test interface based on the provided series of instructions to test each of the respective multiple devices in parallel.

14. The method of claim 9 wherein the test interface includes multiple processing units that are associated with respective multiple devices, and wherein the step of controlling the test interface includes the steps of:

operating the multiple processing units of the test interface based on instructions defined by a single instance of the test procedure to test each of the respective multiple devices in parallel.

15. The method of claim 9 wherein the device is a mixed signal device, and wherein the step of providing includes the steps of:

providing instructions directing the processor to perform an analog signal test operation based on one of the first and second test elements of the test procedure; and

providing instructions directing the processor to perform a digital signal test operation based on another of the first and second test elements of the test procedure.

16. The method of claim 9, further comprising the steps of:

analyzing the test procedure to identify which programmable input variables of the test elements of the test procedure require initial values; and

creating a graphical user interface component which prompts a user to provide the required initial values to initialize the identified programmable input variables.

17. A method for providing a test procedure for testing a device, comprising the steps of:

combining test elements from a test element database to form a test procedure such that (i) the test procedure defines a device testing task, (ii) the test procedure includes multiple test elements, and (iii) each test element defines instructions and programmable input variables that direct a processor to perform a particular test operation of the device testing task;

setting at least a portion of the programmable input variables of each test element forming the test procedure to initial values;

indicating an operating order for the test elements forming the test procedure; and

storing the test procedure within a memory,

wherein the step of indicating the operating order includes the step of:

designating a first test element that directs the processor to perform a first test operation that generates a first test result ahead of a second test element that directs the processor to perform a second operation that (i) obtains the first test result generated by the first test operation performed by the processor under direction of the first test element, and (ii) generates a second test result based on the first test result.

18. The method of claim 13 further comprising the step of:

nesting the test procedure within another test procedure.

19. The method of claim 13 wherein the device is a mixed signal device, and wherein the step of combining includes the steps of:

incorporating one of the first and second test elements into the test procedure, the one of the first and second test elements defining instructions which direct the processor to perform an analog signal test operation; and

incorporating another of the first and second test elements into the test procedure, the other of the first and second test elements defining instructions which direct the processor to perform a digital signal test operation.

20. The system of claim 1 wherein each test element of a set of test elements within the test procedure corresponds to a particular signal used by the device.

21. A system for testing a device, comprising:

memory having a test application stored therein;

a test interface to connect to a device; and

a processor coupled to the memory and the test interface, the processor being configured to operate in accordance with the test application to:

i) provide a series of instructions based on a test procedure defining a device testing task, the test procedure including multiple test elements, each test element defining instructions and programmable input variables that direct the processor to perform a particular test operation of the device testing task, and

ii) control the test interface based on the provided series of instructions in order to test the device,

wherein the test procedure includes:

a first test element that directs the processor to perform a first test operation that generates a first test result; and

a second test element that directs the processor to perform a second operation that (i) obtains the first test result generated by the first test operation performed by the processor under direction of the first test element, and (ii) generates a second test result based on the first test result.

22. A system for testing a device, comprising:

memory having a test application stored therein;

a test interface to connect to a device;

a processor coupled to the memory and the test interface, the processor being configured to operate in accordance with the test application to:

i) provide a series of instructions based on a test procedure defining a device testing task, the test procedure including multiple test elements, each test element defining instructions and programmable input variables that direct the processor to perform a particular test operation of the device testing task, and

ii) control the test interface based on the provided series of instructions in order to test the device; and

a test element database which stores a first test element having instructions that direct the processor to perform an analog signal operation, and a second test element having instructions that direct the processor to perform a digital signal operation, wherein the test procedure is configured to direct the processor to the first and second test elements to perform a mixed signal test based on the first and second test elements.

* * * * *

EXHIBIT B

US006966019B2

(12) **United States Patent**
Viens et al.

(10) **Patent No.: US 6,966,019 B2**
(45) **Date of Patent: Nov. 15, 2005**

(54) **INSTRUMENT INITIATED COMMUNICATION FOR AUTOMATIC TEST EQUIPMENT**

(75) Inventors: **Dominic Viens**, Milford, MA (US); **Gregory P. Brown**, Cambridge, MA (US); **Larry Klein**, Newton, MA (US); **Francois Labonte**, Palo Alto, CA (US)

(73) Assignee: **Teradyne, Inc.**, Boston, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 639 days.

(21) Appl. No.: **10/186,195**

(22) Filed: **Jun. 28, 2002**

(65) **Prior Publication Data**

US 2004/0003328 A1 Jan. 1, 2004

(51) **Int. Cl.**^{7} **G01R 31/317**
(52) **U.S. Cl.** 714/724; 716/4
(58) **Field of Search** 714/724; 716/4; G01R 31/317

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,402,055 A		8/1983	Lloyd et al.	....	364/579
5,361,336 A		11/1994	Atchison	....	395/275
5,673,272 A		9/1997	Proskauer et al.	....	371/22.1
6,031,349 A	*	2/2000	Hard et al.	....	318/466
6,107,818 A		8/2000	Czamara	....	324/765
6,265,842 B1	*	7/2001	Hard et al.	....	318/466
6,400,173 B1	*	6/2002	Shimizu et al.	....	324/765
6,727,723 B2	*	4/2004	Shimizu et al.	....	324/765

FOREIGN PATENT DOCUMENTS

EP	0 939 321	9/1999	.... G01R/31/319

OTHER PUBLICATIONS

Mehtani, et al. "MixTest: A Mixed–Signal Extension to a Digital Test Sytem" IEEE, Paper 42.4, pp. 945–953, 1993.

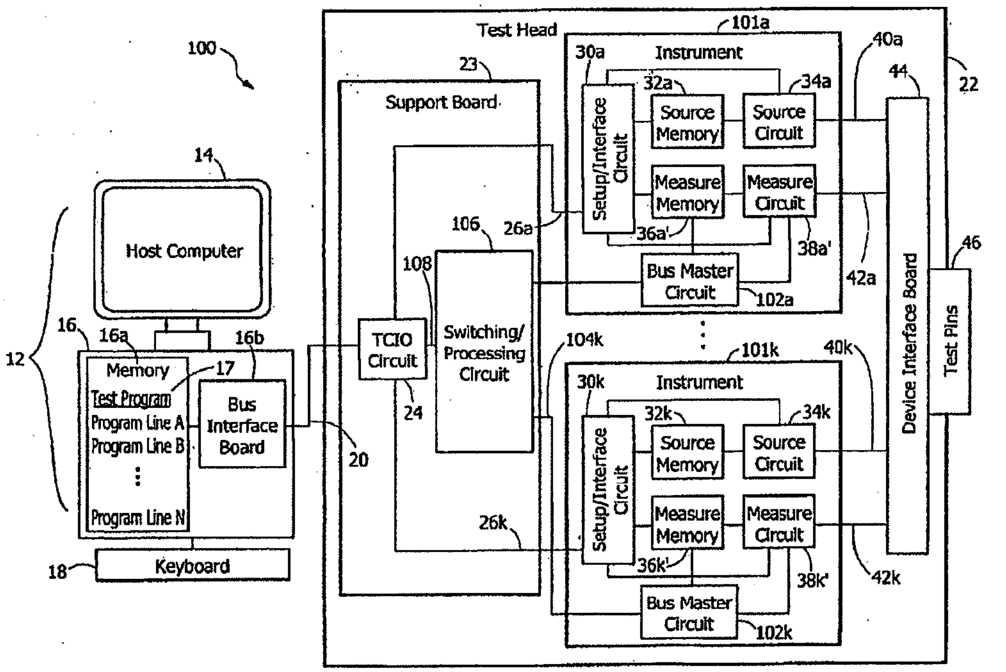
* cited by examiner

Primary Examiner—R. Stephen Dildine

(57) **ABSTRACT**

An automatic test system transfers measure data from one or more test instruments to a processor and processes the measure data, the packaging, transfer, and the processing of the measure data initiated by the one or more test instruments. A method for testing a device under test includes capturing measure data with a test instrument, and initiating, with the test instrument, operation upon the measure data. The operations include packaging the measure data to provide packaged data, and transferring the packaged data to a switching/processing circuit for processing. A method of manufacturing an electronic circuit includes fabricating the electronic circuit and testing the electronic circuit with the aforementioned method.

20 Claims, 5 Drawing Sheets



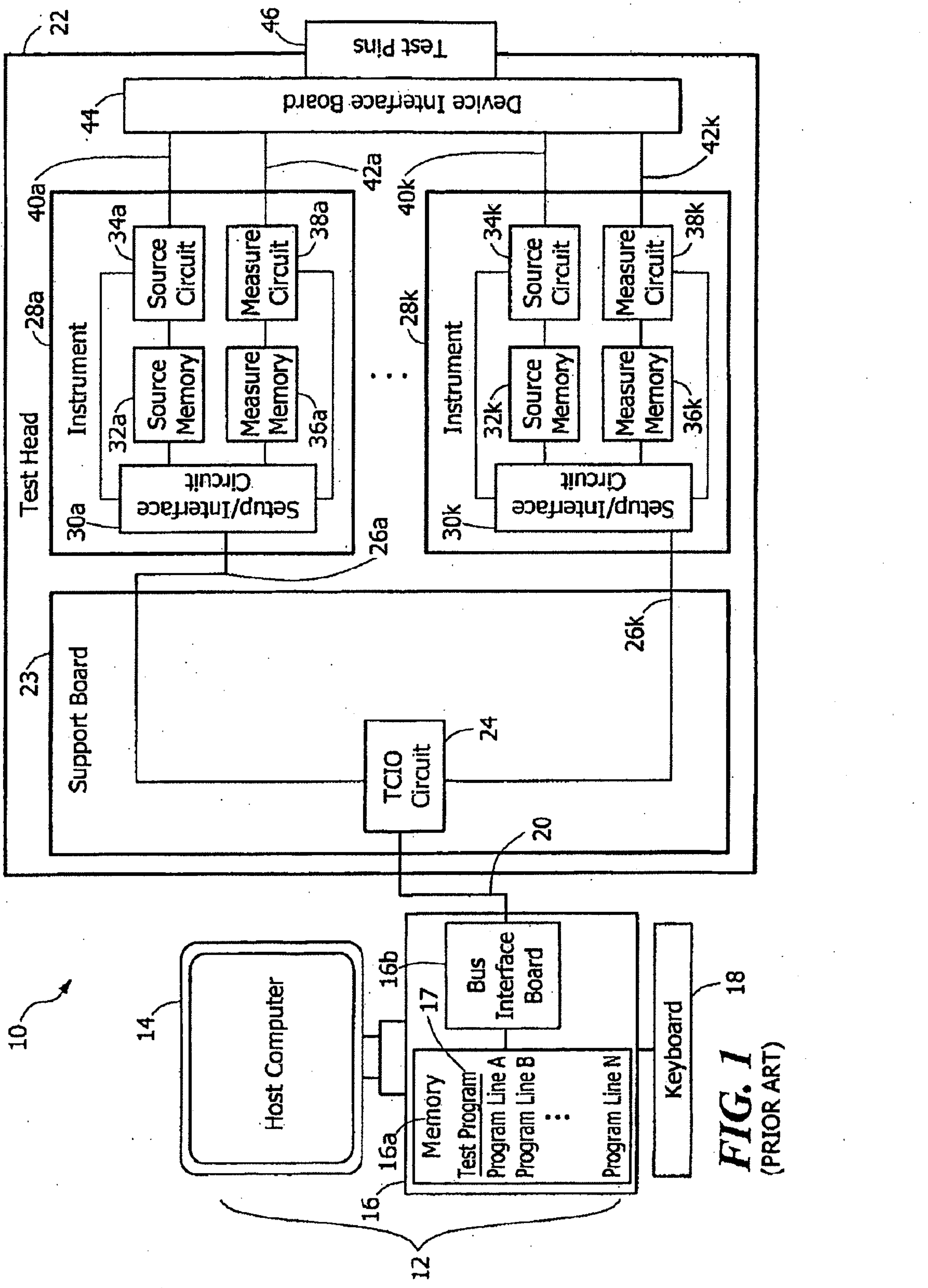


FIG. 1
(PRIOR ART)

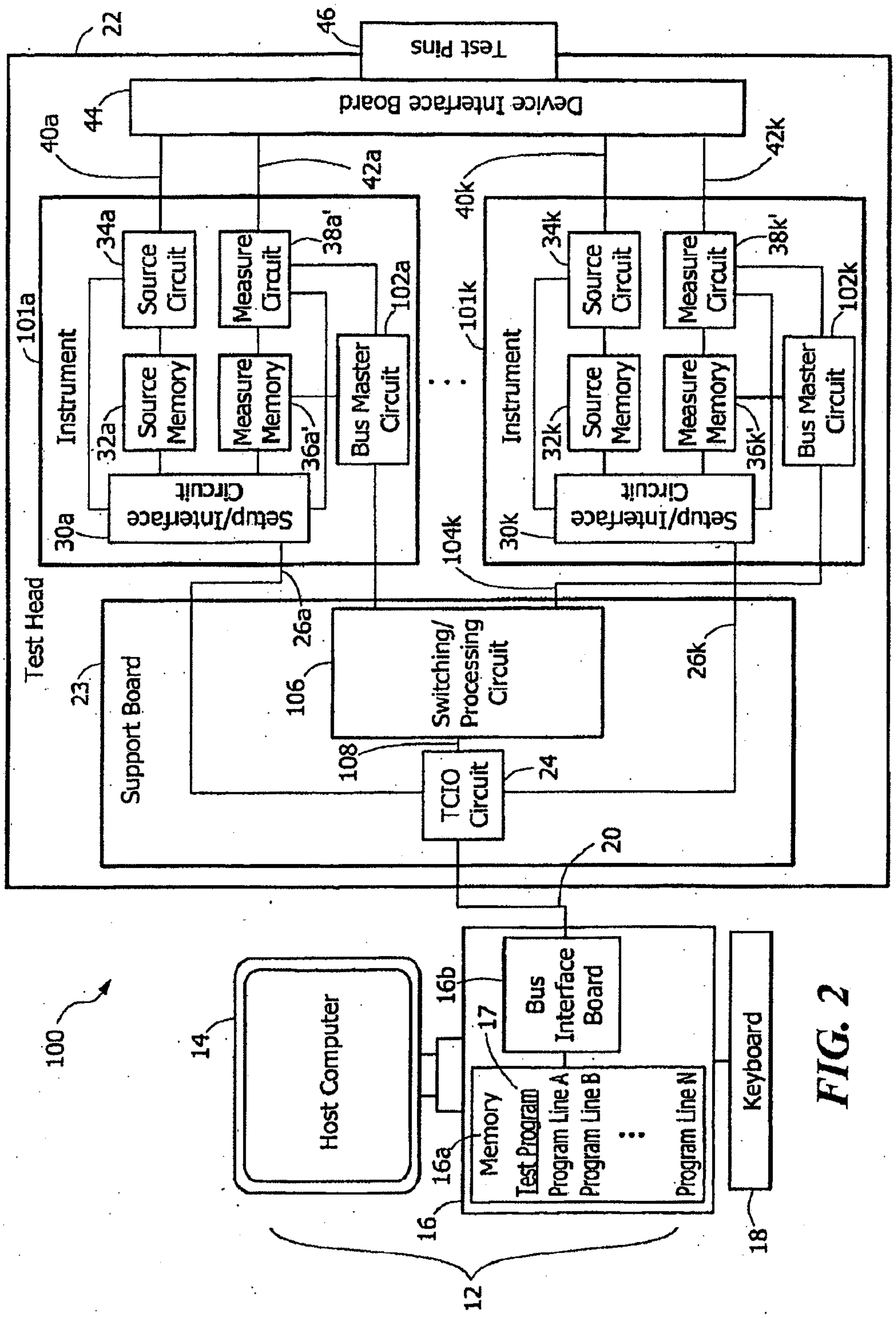

22
46
Test Pins
Device Interface Board
44
40a
42a
40k
42k
Instrument
101a
34a
Source Circuit
Measure Circuit
38a'
Source Memory
Measure Memory
32a
36a'
Bus Master Circuit
102a
Setup/Interface Circuit
30a
Instrument
101k
34k
Source Circuit
Measure Circuit
38k'
102k
Source Memory
Measure Memory
32k
36k'
Bus Master Circuit
Setup/Interface Circuit
30k
104k
Test Head
23
Support Board
26a
106
Switching/ Processing Circuit
26k
108
TCIO Circuit
24
20
100
14
Host Computer
16b
Bus Interface Board
17
Memory
Test Program
Program Line A
Program Line B
Program Line N
16a
16
Keyboard
18
12

FIG. 2

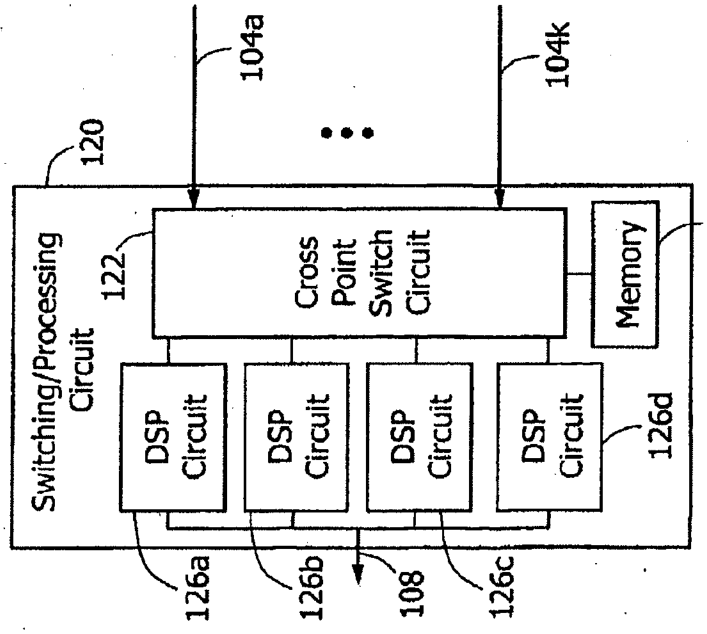

120
Switching/Processing Circuit
104a
104k
122
Cross Point Switch Circuit
Memory
124
DSP Circuit
DSP Circuit
DSP Circuit
DSP Circuit
126a
126b
108
126c
126d

FIG. 3

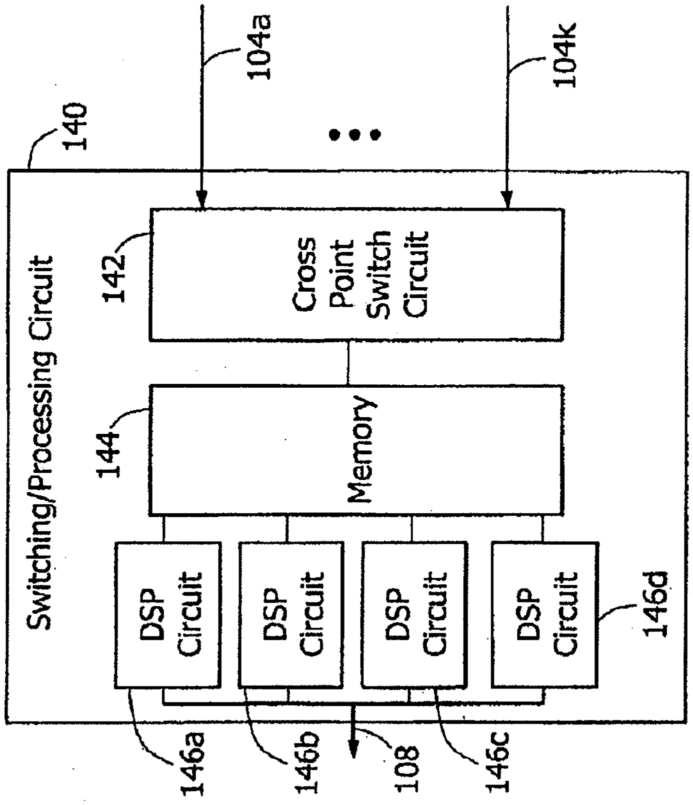

140
Switching/Processing Circuit
104a
104k
142
Cross Point Switch Circuit
144
Memory
DSP Circuit
DSP Circuit
DSP Circuit
DSP Circuit
146a
146b
108
146c
146d

FIG. 3A

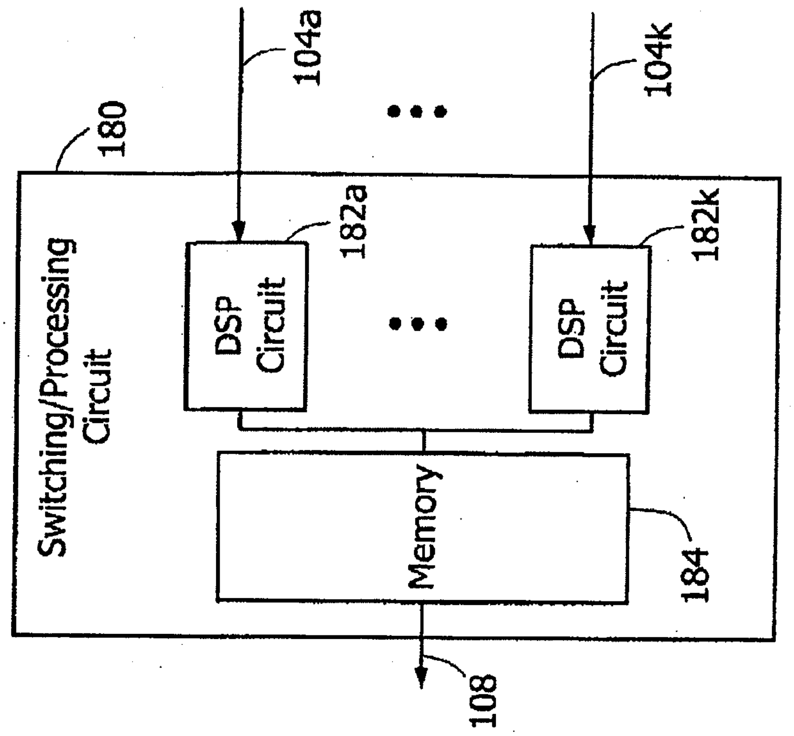

180
Switching/Processing Circuit
104a
104k
DSP Circuit
182a
DSP Circuit
182k
Memory
184
108

FIG. 3C

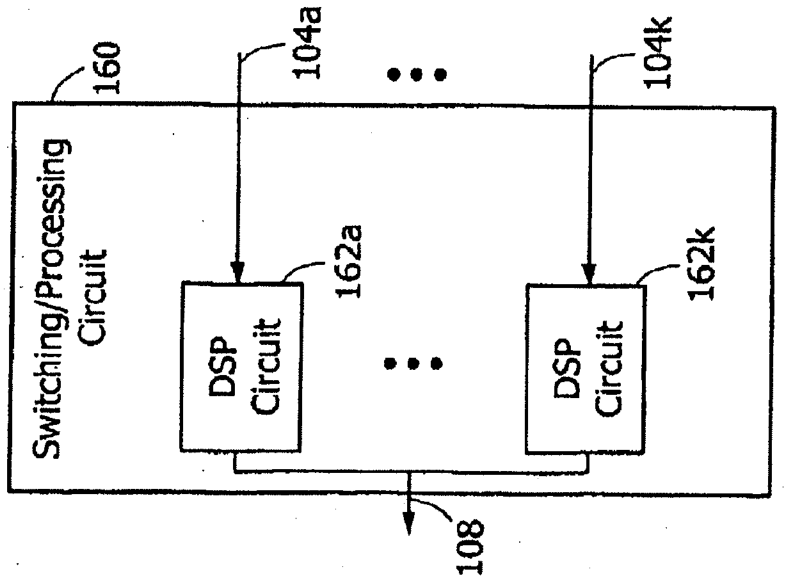

160
Switching/Processing Circuit
104a
104k
DSP Circuit
162a
DSP Circuit
162k
108

FIG. 3B

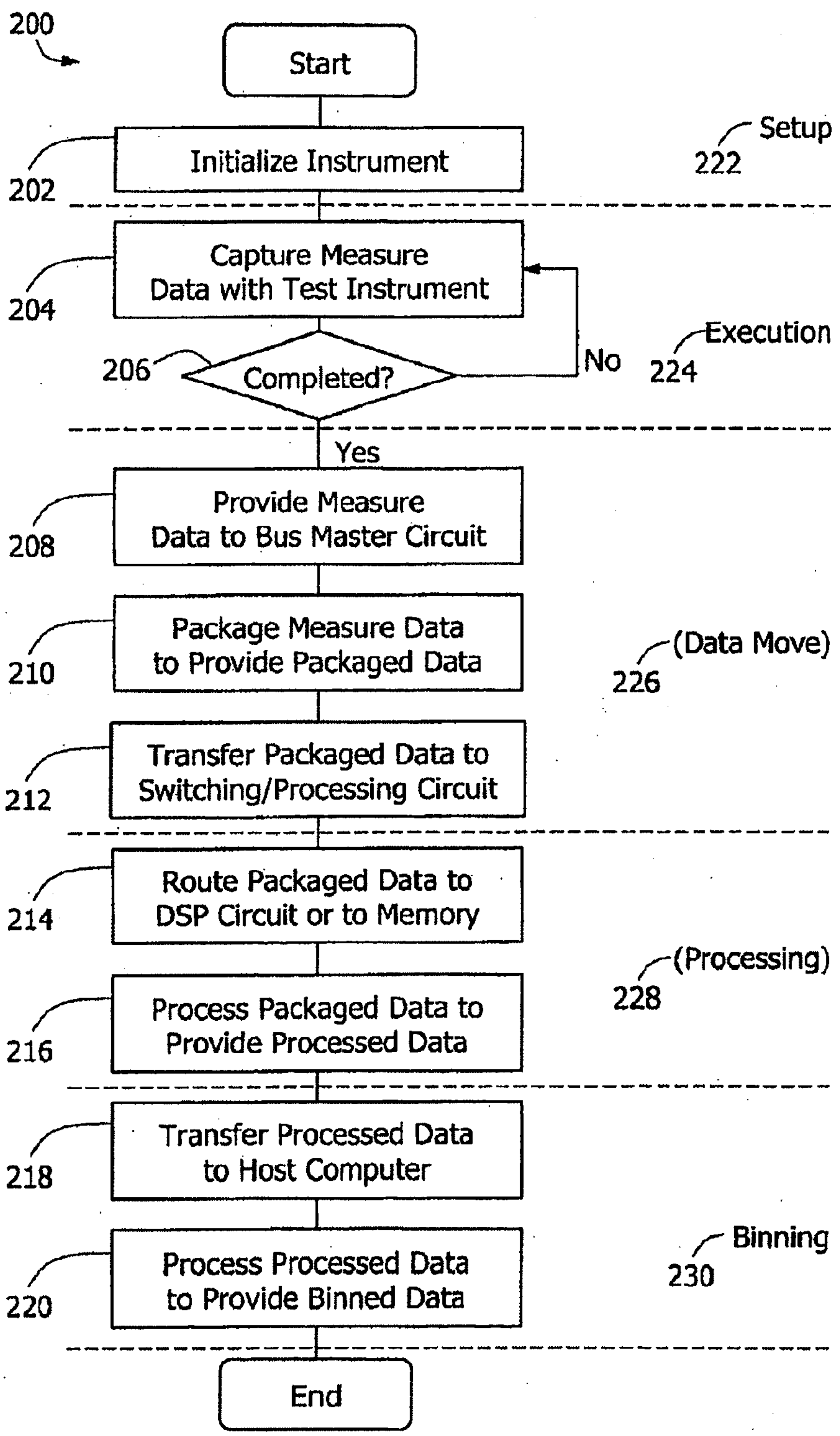

200
Start
Initialize Instrument
202
Setup
222
Capture Measure Data with Test Instrument
204
206
Completed?
No
Execution
224
Yes
Provide Measure Data to Bus Master Circuit
208
Package Measure Data to Provide Packaged Data
210
(Data Move)
226
Transfer Packaged Data to Switching/Processing Circuit
212
Route Packaged Data to DSP Circuit or to Memory
214
(Processing)
228
Process Packaged Data to Provide Processed Data
216
Transfer Processed Data to Host Computer
218
Process Processed Data to Provide Binned Data
220
Binning
230
End

FIG. 4

INSTRUMENT INITIATED COMMUNICATION FOR AUTOMATIC TEST EQUIPMENT

CROSS REFERENCE TO RELATED APPLICATIONS

Not Applicable.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH

Not Applicable.

FIELD OF THE INVENTION

This invention relates generally to automatic test equipment (ATE) used to test electronic circuits.

BACKGROUND OF THE INVENTION

As is know in the art, some automatic test equipment (ATE) is adapted to electrically test integrated circuits at the wafer stage of integrated circuit (IC) fabrication and/or at the packaged IC stage. The circuit being tested is often referred to as a device under test (DUT). So-called, "handling equipment" physically manipulates the DUTs to provide a coupling between the DUTs and the ATE. For example, in one type of system, handling equipment provides a rapid feed of packaged ICs, sequentially coupling each IC to the ATE. In another system, the handling equipment utilizes step and repeat miniature probes that couple the ATE to individual IC dies on a wafer, thereby allowing the dies to be sequentially tested by the ATE.

As is also known, to keep the cost of IC's low and to meet the supply requirements of IC users, there has been a trend to reduce the amount of time required by the ATE to test an IC. To help address the IC cost and supply issues, state-of-the-art ATE systems typically include a host computer coupled to one or more test instruments. Each of the one or more test instruments is typically coupled to a moveable test head. The host computer can be a personal computer, a workstation, or any digital computing platform. The one or more test instruments, responsive to the host computer can include both generic test equipment, for example a conventional voltmeter, and also specialized test equipment integrated with the ATE, for example a digital pattern generator. Under control of the host computer and a test program associated therewith, the one or more test instruments can receive signals from and provide signals to a DUT through THE test head. The test head includes a group of test pins typically referred to as a "pin field." The pin field includes one or more pins, (e.g. so-called pogo pins) through which the one or more test instruments are coupled to one or more DUTs provided by the handling equipment.

The one or more test instruments can provide one or more "source signals" to respective pins within the pin field. The source signals may include, for example, one or more analog signals or one or more digital signals or a combination of one or more analog signals and one or more digital signals. The source signals are this coupled through the pin field to the DUT.

The one or more test instruments can also receive "measure signals" from the DUTs through respective pins within the pin field. The measure signals may include, for example, one or more analog signals or one or more digital signals or a combination of one or more analog signals and one or more digital signals. The measure signals are coupled to the one or more test instruments from the DUT through the pin field. Thus, the ATE, in conjunction with the handling equipment, can provide signals to and receive signals from either dies upon a wafer or packaged ICs.

A source signal, whether analog or digital, can be represented as "source data." In providing a source signal, it is often necessary to transfer the "source data" that corresponds to the source signal from the host computer to the one or more test instruments. The source data can correspond, for example, to digital samples of a sine wave. The source data is transferred to one or more of the test instruments and used by the test instruments to provide the source signals. It should be appreciated that there can be a large amount of source data.

Similarly, a measure signal, whether analog or digital, can be represented as digital samples referred to as "measure data." The measure signals are typically converted to "measure data" by the test instrument. It is often necessary to transfer the measure data from the test instrument to the host computer. It should be appreciated that there can be a relatively large amount of measure data.

The host computer executes a test program that includes instructions for setup and control of the test instruments, instructions for generation of the source data, instructions for transfer of the source data to the test instrument, instructions for capture of the measure data received from the DUT, instructions for transfer of the measure data to the host computer, instructions for processing the measure data to generate "processed data," and instructions to process the processed data to generate "binned data." Processed and binned data are described below.

Each of the above instructions can be categorized according to function categories associated with the test program, the function categories including a) setup, b) execution, c) data move, d) processing, and e) binning. It should be noted that a conventional test program can include instructions corresponding to each of these function categories. It will also be noted that each different test program takes a different amount of time to setup, execute, data move, process and bin.

Of the above instructions, the instructions for setup and control of the test instruments, the instructions for generation of the source data, and the instructions for transfer of the source data to the test instrument, each correspond to the setup function category. Each of these instructions are generally provided only once at the beginning of the test program.

The instructions for capture of the measure signals are associated with acquisition of data by the test instrument, the time duration of which is determined by the characteristics of the test instrument. The instructions for capture of measure signals, associated with the execution function category, are generally repeated a number of times within the test program.

The instructions for transfer of the measure data to the host computer, associated with the data move function category, also are generally repeated a number of times within the test program.

The instructions for processing of the measure signals to generate the processed data, associated with the processing function category, also are generally repeated a number of times within the test program.

The instructions for processing the processed data to generate binned data, associated with the binning function category, are generally provided only once at the end of the test program. The data processing and the binning operations are often performed by the host computer.

The execution time (or "run time") of a test program which implements one individual test on an IC is typically about several seconds. It is often necessary, however, to perform hundreds or thousands of individual tests on each IC. For a particular test program, any one or more of the function categories described above can dominate the total amount of testing time associated with the test program. However, the lengthiest time durations are often those associated with functions that must be repeated a number of times within the test program, rather than with those functions that are performed only one time per test program. As described above, the instructions associated with the execution, data move, and processing function categories are typically repeated a number of times within the test program. Thus, the processes carried out in these function categories are often associated with the longest time durations of the test program.

The time durations associated with the above-mentioned execution function category are influenced by a variety of factors, including but not limited to, the test instrument acquisition speed and the number of measure samples required for a particular test. For example, the test instrument can be a high speed digital signal capture instrument that can capture 8 digital states (one byte) in parallel at a rate of 100 Mbytes per second having a total number of captured bytes of 256,000. This capture would take 0.256 milliseconds. If this data capture were repeated a number of times in the test program, a large amount of time could be devoted to the data capture.

The time durations associated with the above-mentioned data move function category are also influenced by a variety of factors, including, but not limited to, the amount of measure data to be transferred and the speed of the data bus upon which the transfer is done. For example, a transfer of one thousand digital bytes can be associated with an instruction for transfer of measure data, and the measure data can be transferred on a one bit wide serial bus having a bus speed of 10 Mbits per second. In this example, the data transfer would take 0.8 milliseconds. If this data transfer were repeated a number of times in the test program, a large amount of time could be devoted to the transfer of measure data.

The time durations associated with the above-mentioned processing function category are also influenced by a variety of factors, including, but not limited to, the type of processing, the processing speed of the processor that performs the processing, and the amount of data to process. For example, the processing can be a fast Fourier transform performed on one thousand measure samples with a personal computer.

With regard to the above-mentioned data move function category, ATE is often configured having a single data bus coupled between the host computer and the one or more test instruments. Often, the single data bus is a relatively slow data bus. Thus, in many ATE, the time duration associated with the instructions for transfer of measure data from the one or more test instruments to the host computer often has the longest duration. Furthermore, the test program, upon reaching an instruction requesting the transfer of the measure data, must wait until the measure data transfer is completed before proceeding to the next test program instruction.

With regard to the above-mentioned processing function category, the measure data is often processed with functions, for example FFTs, which are most rapidly performed using floating-point arithmetic. Many ATE systems are provided having a host computer that is not optimized to perform floating-point arithmetic on the measure data.

It would, therefore, be desirable to increase the speed with which an ATE system can test a DUT. It would also be desirable to reduce the amount of time required to transfer the measure data from the DUT to the host computer. It would also be desirable to provide an ATE system that can rapidly process measure data.

SUMMARY OF THE INVENTION

In accordance with an embodiment of the present invention, a circuit testing apparatus includes at least one test instrument and a bus master circuit coupled to the at least one test instrument. With this particular arrangement, an ATE system is provided which automatically transfers measure data from the one or more test instruments to a processor. One or more bus master circuits are adapted to receive measure data from respective ones of the at least one test instruments. Upon initiation from one of the at least one test instruments, the respective bus master circuit packages the measure data in accordance with a predetermined protocol to provide packaged data and transfers the packaged data to a switching/processing circuit. The switching/processing circuit is adapted to receive the packaged data and to process the packaged data to provide processed data and to communicate the processed data to a host computer. In this manner, the test program need not provide instructions for the transfer of measure data nor wait for the transfer to complete before proceeding. Thus the circuit testing apparatus can test circuits relatively rapidly.

In accordance with another aspect of the present invention, a circuit testing method includes capturing measure data in a test instrument, the measure data generated in response to the test instrument implementing one or more tests to an electronic circuit. The method also includes initiating, with the test instrument, operations upon the measure data. The operations include packaging the measure data with a bus master circuit to provide packaged data having a data packet and a data header. The operations also include transferring the packaged data to a switching/processing circuit, routing the packaged data with the switching/processing circuit to a processor, and processing the packaged data with the processor to provide processed data. The processing is responsive to the data header.

With this particular arrangement, a technique for reducing the amount of time required to test a circuit is provided. By initiating, within the test instrument, the packaging and the transfer of the measure data to a processor, the time required to transfer and to process the measure data is eliminated from the host computer flow of operations.

In accordance with another aspect of the present invention, a method of manufacturing an electronic circuit includes fabricating an integrated circuit and testing the integrated circuit with a circuit testing apparatus by capturing, in a test instrument, measure data provided by the electronic circuit, and initiating, with the test instrument, operations upon the measure data. The operations include packaging the measure data with a bus master circuit to provide packaged data having a data packet and a data header. The operations also include transferring the packaged data to a switching/processing circuit, routing the packaged data with the switching/processing circuit to a processor, and processing the packaged data with the processor to provide processed data. The processing is responsive to the data header.

With this particular arrangement, a method of rapidly manufacturing an electronic circuit is provided. By initiating, within the test instrument, the packaging and the transfer of the measure data to a processor, the time required to manufacture the electronic circuit is reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing features of the invention, as well as the invention itself may be more fully understood from the following detailed description of the drawings, in which:

FIG. 1 is a block diagram of a prior art automatic test equipment (ATE);

FIG. 2 is a block diagram of an exemplary ATE in accordance with the present invention;

FIG. 3 is a block diagram of an exemplary embodiment of a switching/processing circuit provided as part of the ATE shown in FIG. 2;

FIG. 3A is a block diagram of another exemplary embodiment of a switching/processing circuit provided as part of the exemplary ATE shown in FIG. 2;

FIG. 3B is a block diagram of yet another exemplary embodiment of a switching/processing circuit provided as part of the exemplary ATE shown in FIG. 2;

FIG. 3C is a block diagram of yet another exemplary embodiment of a switching/processing circuit provided as part of the exemplary ATE shown in FIG. 2; and

FIG. 4 is a flow chart of an exemplary process for measure data capture and measure data transfer provided in association with the exemplary ATE shown in FIG. 2.

DETAILED DESCRIPTION OF THE INVENTION

Before describing the instrument initiated moves and instrument initiated processing in accordance with the present invention, some introductory concepts and terminology are explained. The terms "ATE" and "tester," as used herein, are used to describe automatic test equipment that can be used to electrically test a device under test (DUT). The DUT can be an electronic system, an integrated circuit (IC), a discrete circuit, for example a transistor, a populated circuit board, or a bare circuit board. Many examples used herein refer to an ATE that is used to test an IC. However, it will be understood that the concepts described herein apply equally well to an ATE used to test any of the aforementioned DUTs.

The term "source signal" as used herein, refers to a signal applied to the DUT. The same signal may be provided as any type of signal including but not limited to an electrical signal, an electronic signal, a radio frequency (RF) signal or an optical signal. The source signal can be either an analog source signal or a digital source signal. The source signal is often a sampled signal constructed from "source data." The source data may be provided, for example, as digital samples of a sine wave. Through a digital-to-analog (D/A) converter, the source data can be used to generate a source signal sine wave. For another example, the source data can be digital bits and, through a buffer and/or a level translator, the source data can be used to generate a source digital waveform.

The term "measure signal," as used herein, refers to a signal provided by the DUT in response to one or more source signals. The measure signal can be either an analog measure signal or a digital measure signal. An analog measure signal can be digitally sampled with an analog-to-digital (A/D) converter or the like and converted into "measure data." A digital measure signal can be constructed into digital bytes, the digital bytes corresponding to the measure data.

Referring now to FIG. 1, an exemplary embodiment of a prior art tester **10** includes a host computer **12**. The host computer **12** can be a personal computer (PC), a workstation, or any computing platform. The host computer **12** includes a display **14**, a processing portion **16** and a keyboard **18**. The processing portion **16** includes a program memory **16*a*** having stored therein test program code **17** executable by the processing portion **16**. The host computer **12** also includes a bus interface circuit board **16*b***.

Through the bus interface circuit board **16*b***, the test program **17** communicates with a test head **22** through a test computer input/output (TCIO) bus **20**. The test head **22** includes a support circuit board **23** including a TCIO circuit **24**. The TCIO circuit **24** is coupled to the TCIO bus **20**. The TCIO circuit **24** is also coupled to one or more test instruments **28*a*–28*k*** through respective test instrument busses **26*a*–26*k***. While eleven test instruments **28*a*–28*k***, associated with eleven respective test instrument busses **26*a*–26*k*** are here shown, it should be appreciated that any number of test instruments and associated test instrument busses can be used.

Taking the test instrument **28*a*** as representative of all such test instruments **28*a*–28*k***, the test instrument **28*a*** includes a setup/interface circuit **30*a*** coupled to both a source memory **32*a*** and to a source circuit **34*a***. The source memory **32*a*** is also coupled to the source circuit **34*a***. The source memory **32*a*** is a storage resource in which source data generated by the test program **17** can be stored. The source data, described above, is provided to the source circuit **34*a*** and is used to generate a source signal **40*a***. The source signal **40*a*** is coupled to a device interface board **44** that is further coupled to a group of test pins **46**, also called a pin field **46**. The device interface board **44** provides connection switching of the source signal **40*a*** to one or more of the test pins **46**. The test pins **46** are coupled to a DUT (not shown). The source signal **40*a*** can be an analog or digital source signal, or a combination of an analog and a digital source signal.

The test pins **46** and the device interface board **44** coupled thereto are also coupled to a measure circuit **38*a***. The measure circuit **38*a*** is coupled to a measure memory **36*a***, and both the measure circuit **38*a*** and the measure memory **36*a*** are coupled to the setup/interface circuit **30*a***. In response to a source signal, the DUT provides one or more measure signals **42*a*** through the test pins **46** to the measure circuit **38*a***. The one or more measure signals **42*a*** can be analog or digital measure signals, or a combination of an analog and a digital measure signals. The measure circuit **38*a*** converts the one or more measure signals **42*a*** to measure data which are stored in the measure memory **36*a***.

As described above, the host computer **12** runs the test program **17** that includes instructions for setup and control of the test instruments **28*a*–28*k***, instructions for generation of the source data, instructions for transfer of the source data to the test instruments **28*a*–28*k***, instructions for capture of the measure data in the measure memory **36*a*–36*k***, instructions for transfer of the measure data to the host computer **12**, instructions for processing the measure data in the host computer **12** to generate "processed data," and instructions to process the processed data in the host computer **12** to generate "binned data." The processed data and the binned data are further described below.

Signals generated by the test program **17** can be transmitted over the TCIO bus **20** and also the test instrument bus **26*a***. For example, signals corresponding to instructions for setup and control of the test instrument **28*a*** including setup

EXHIBIT B PG. 32

of the source circuit 34a and the measure circuit 38a can be transmitted over the TCIO bus 20 and the test instrument bus 26a. Also, signals corresponding to instructions for transfer of the source data to the source memory 32a, instructions for capture of the measure data in the measure memory 36a, and instructions for transfer of the measure data from the measure memory 36a to the host computer 12, can also be transmitted over the TCIO bus 20 and the test instrument bus 26a. Thus, the TCIO bus 20 and the test instrument bus 26a are bi-directional busses (or functionally equivalent transmission paths) which allow signals to be transmitted in two directions.

In operation, the host computer 12 provides signals corresponding to the instructions for setup and control of the test instrument 28a to the setup/interface circuit 30a. The test instrument 28a is adapted to receive the signals corresponding to instructions for setup and control of the test instrument 28a and to act upon the instructions in a variety of ways, including, but not limited to, setup of D/A converters within the source circuit 34a, setup of amplifier gains within the source circuit 34a, setup of filters within the source circuit 34a, setup of A/D converters within the measure circuit 38a, setup of amplifier gains within the measure circuit 38a, and setup of filters within the measure circuit 38a. The setup can also include setup of timing edge placement of digital source signals.

The signals corresponding to the instructions for transfer of the source data to the test instrument 28a are also provided by the host computer 12 to the setup/interface circuit 30a. These signals can include transfer of source data that can be reconstructed into a sampled analog waveform. For example, the source data can be samples of a sine wave that are converted to a sampled sine wave at the test instrument 28a by a D/A converter or the like. The signals corresponding to the instructions for transfer of the source data to the test instrument 28a can also include instructions for transfer of a source pattern, the source pattern corresponding to a pattern of digital signals that are provided by the test program and stored in the source memory 32a. The source pattern corresponds to a particular desired pattern of digital source signal 40a.

In response to a particular source signal 40a, the DUT (not shown) is expected to provide a particular measure signal 42a associated with processed data having particular pre-determined processed data thresholds. The pre-determined processed data thresholds can include a variety of thresholds, including, but not limited to, magnitude thresholds, timing thresholds, and measured digital pattern thresholds.

Upon receiving the measure signal 42a from the DUT, the measure circuit 38a converts the measure signals to measure data as required for storage in the measure memory 36a. Upon receipt of an instruction from the test program 17 corresponding to a transfer of the measure data from the measure memory 36a, the measure data is transferred on the test instrument bus 26a to the TCIO circuit 24 and further transferred to the host computer 12 on the TCIO bus 20. It should be understood that the measure data is transferred from the measure memory 36a upon initiation from the test program 17. The measure data is then processed by the host computer 12 to provide the processed data. The processed data is then processed by the host computer to provide the binned data.

The host computer 12, by processing the measure data to provide the processed data, and then associating the processed data with the processed data thresholds, generates a pass/fail determination associated with each particular test of the test program 17. The pass/fail determination is referred to herein as "binned data." Since the test program can provide a plurality of tests, a plurality of binned data can be associated with the test program. For example, the measured data can be 1024 measured data values and the measured data can be processed via a fast Fourier transform (FFT), for example, to generate processed data having 512 FFT processed data points. The processed data can be associated with magnitude data thresholds to provide binned data having a single pass or fail value. From this example, it should be appreciated that the amount of processed data is often less than the amount of measured data, and the amount of binned data is often less than the amount of processed data.

In one particular embodiment, the TCIO bus 20 is provided as a serial bus having a bit rate of 100 Mbits per second and the test instrument data bus 26a is provided as a serial data bus having a bit rate of 100 Mbits per second. Thus, 12.5 million 8-bit bytes per second can be carried on each of the two data busses. It will be appreciated that where a test program must transfer millions of measure data words from the test instrument 28a to the host computer 12, the test time can be unacceptably large.

As described above, the host computer 12 executes the test program 17 having a variety of function categories. Also, as described above, a long time duration is often associated with the data move function category, specifically the time duration associated with the instructions for transfer of the measure data to the host computer. In the ATE 10, the measure data is transferred from the measure memory 36a, through the setup/interface circuit 30a, to the test instrument bus 26a, through the TCIO circuit portion 24, to the TCIO bus 20, to the host computer. The test program 17, upon reaching an instruction requesting the transfer of the measure data, must wait until the measure data is transfer is completed before proceeding to the next test program instruction.

While test instruments 28a–28k are shown each having respective source memories 32a–32k source circuits 34a–34k, measure memories 36a–36k, and measure circuits 38a–38k, in an alternate embodiment, one or more of the test instruments have only source memories and source circuits or only measure memories and measure circuits.

Referring now to FIG. 2, in which like elements of FIG. 1 are provided having like reference designations, an exemplary ATE 100 in accordance with the present invention includes one or more test instruments 101a–101k, each of the test instruments 101a–101k including a bus master circuit 102a–102k. Each of the respective bus master circuits 102a–102k is coupled to a corresponding one of measure memories 36a'–36k' and measure circuits 38a'–38k'. The measure circuits 38a'–38k' receive measure signals 42a–42k from the DUT, converts the measure signals to measure data, and provides the measure data to the measure memories 36a'–36k'. Here, the measure circuits 38a'–38k' and the measure memories 36a'–36k' are distinguished from the measure circuits 38a–38k and the measure memories 36a–36k shown in FIG. 1, since they are provided each having a coupling to respective ones of the bus master circuits 102a–120k mentioned above.

Taking the test instrument 101a as representative of all such test instruments 101a–101k, the bus master circuit 102a receives the measure data from the measure memory 36a'. The bus master circuit formats and transfers the measure data from the measure memory 36a' upon initiation by the measure circuit **38***a*'. The measure circuit **38***a*' initiates a transfer once a particular measurement has completed.

In contrast to the prior art ATE of FIG. **1**, where the measure data is transferred from the measure memory **36***a* upon initiation by the test program **17**, in accordance with the present invention, the measure circuit **38***a*' initiates the transfer of data from the memory **36***a*' to the bus master circuit **102***a* as well as the transfer of data from the bus master circuit **102***a* to a switching/processing circuit **106**. Thus, since the measure circuit **38***a*' recognizes when testing is complete, the measure circuit **38***a*' is able to initiate all further processing of the data.

As described above, in the prior art system of FIG. **1**, the test program must wait until the measure data is transferred. In accordance with the present invention, however, the measure data is formatted and transferred by the bus master circuit **102***a* upon initiation from the measure circuit **38***a*', and no instruction for the transfer of measure data need be written into the test program. Thus, the test program need not wait for the transfer to complete prior to executing its next instruction.

The bus master circuit **102***a* formats the measure data to provide formatted data. The formatted data includes a data header and a data packet. The data packet corresponds to the measure data. The data header includes a variety of descriptors including but not limited to descriptors which identify the particular test instrument **101***a*, the amount of measure data, and the type of processing that is to be performed upon the measure data. Hereafter, the header information corresponding to the type of processing that is to be performed upon the data packet will be referred to as an "extended header descriptor." The particular content of the header is selected in accordance with a variety of factors, including, but not limited to, a desire to minimize the header size so that the formatted data can subsequently be rapidly transferred, as well as the particular application in which the ATE is being used.

While the bus master circuits **102***a*–**102***k* are shown in this exemplary embodiment to be part of the test instruments **101***a*–**101***k*, it should be apparent that all or part of the bus master circuits **102***a*–**102***k* can be provided apart from the test instruments **101***a*–**101***k*.

The formatted data is transferred by the bus master circuit **102***a* through a move data bus **104***a* to the switching/processing circuit **106**. The switching/processing circuit will be described in more detail in FIGS. **3**–**3**C. However, let it suffice here to say that the switching/processing circuit **106** routes the formatted data to a processor within the switching/processing circuit **106**. The formatted data is then processed within the switching/processing circuit **106** in response to the processing specified in the extended header descriptor. The output of the switching/processing circuit **106** provides the processed data described above. The circuit **106** may be provided as a switching fabric which may be implemented using any one of a variety of techniques including but not limited to bus architecture, crossbar switch, or a cross point switch.

While the switching/processing circuit **106** is shown to be separate from the test instruments **101***a*–**101***k*, it should be appreciated that in some embodiments it may be desirable or even necessary that all or part of the switching/processing circuit **106** be provided as part of the test instruments **101***a*–**101***k*. Also, while the switching/processing circuit **106** is shown as part of the support board **23**, in another alternate embodiment, the switching/processing circuit is on another circuit board (not shown).

The processed data is transferred on a switching/processing circuit bus **108** to the TCIO circuit **24**, whereupon the processed data is transferred on the TCIO bus **20** to the host computer **12**. The host computer processes the processed data to provide binned data. In an alternate embodiment, part or all of the binning can be performed by the switching/processing circuit **106**, and binned data is transferred to the host computer **12** on the TCIO bus **20**.

As described above, the processed data and the binned data often correspond to less data than the measured data. Thus, comparing the exemplary ATE **100** in accordance with the present invention to the prior art ATE **10** of FIG. **1**, less data is transferred on the TCIO bus **20** in the direction of the host computer by the ATE **100**.

As also described above, a conventional test program used with the prior art ATE **10** of FIG. **1** includes a variety of instructions that can be categorized according to function categories within the test program, the function categories including a) setup, b) execution, c) data move, d) processing, and e) binning.

It should be recognized, however, that in accordance with the present invention, no instructions need be provided that correspond to the data move category. It should also be recognized that no instructions need be provided that correspond to the processing function category. The exemplary ATE **100** provides transfer of the measure data to the switching/processing circuit **106** upon initiation by the measure circuit **38***a* in combination with the bus master circuit **102***a*, rather than initiation by the test program **17**. With this particular arrangement, it is also not necessary that the test program stop execution to wait for the measure data transfer.

The processing is provided by the switching/processing circuit **106** in response to the extended header descriptor. Information for the extended header descriptor is provided to the test instrument by the test program as part of the setup instructions. It should therefore be recognized that the test program need not provide instructions to process the measure data. Thus, the exemplary ATE **100** can include instructions that can be categorized into fewer function categories within the test program, the function categories including a) setup, b) execution, and c) binning.

It will also be recognized that each of the test instruments, here the test instruments **101***a*–**101***k*, can transfer at the same time (i.e. simultaneously or concurrently transfer) their respective measure data on respective more data busses **104***a*–**104***k* to the switching processing circuit **106** for processing.

In one particular embodiment, the move data busses **104***a*–**104***k* can be provided as high speed parallel data busses each having seven conductors, four of the conductors having data bits, two of the conductors having a differential clock signal, and one of the conductors having a synchronization signal. The formatted data generated by each of the bus master circuits **102***a*–**102***k* can include a 32 bit by 32-bit header and a 32-bit wide data packet, the data packet corresponding to the measure data. In an alternate embodiment, the move data busses **104***a*–**104***k* are provided as Ethernet busses and the formatted data is provided as Ethernet format data, both known to one of ordinary skill in the art.

The switching/processing bus **108** can be provided having any high-speed bus architecture.

While test instruments **101***a*–**101***k* are shown each having respective source memories **32***a*–**32***k*, source circuits **34***a*–**34***k* measure memories **36***a*'–**36***k*', measure circuits **38***a*'–**38***k*', and bus master circuits **102***a*–**102***k*, in an alternate

EXHIBIT B PG. 34

embodiment, one or more of the test instruments have only source memories and source circuits or only measure memories, measure circuits, and bus master circuits.

In another alternate embodiment, the measure data provided by the measure circuit, for example, the measure circuit **38***a*', is provided directly to the bus master circuit **102***a* without first being stored in the measure memory **36***a*'. For this alternate embodiment, no measure memory **36***a*' need be provided.

Referring now to FIG. **3**, in which like elements of FIG. **2** are provided having like reference designations, a first exemplary embodiment of a switching/processing circuit **120** includes a cross point switch circuit **122** coupled to the move data busses **104***a*–**104***k*. The cross point switch **122** is also coupled to a memory **124** and to four DSP circuits **126***a*–**126***d*. The four DSP circuits **126***a*–**126***d* are also coupled to the switching/processing bus **108**.

In operation, the cross point switch **122** switches the formatted data carried on one or more of the move data busses **104***a*–**104***k* to either the memory **124** or to one or more respective DSP circuits **126***a*–**126***d*, (also referred to herein as DSP processors) or to both the memory **124** and to the one or more respective DSP processor **126***a*–**126***d*. A DSP processor, for example the DSP processor **126***a*, generates the processed data and provides the processed data on the switching/processor bus **108** to the TCIO circuit **24**. The processed data is then transferred to the host computer **12** on the TCIO bus **20** (FIG. **2**).

While four DSP processors **126***a*–**126***d* are shown in the exemplary embodiment of FIG. **3**, it should be recognized that fewer or more than four DSP processor can be used. Having multiple DSP processors, for example, the four DSP processors **124***a*–**124***d*, the ATE **100** (FIG. **2**) can do parallel processing to enhance the processing speed. The number of DSP processors to use in circuit **120** is selected in accordance with a variety of factors, including, but not limited to, cost, speed, function to be performed, and the application in which this circuit is used. The size, speed and other characteristics of the memory **124** are selected in accordance with a variety of factors including but not limited to cost, speed, and bandwidth.

In an alternate embodiment, in addition to the processing function, the DSP processors **126***a*–**126***d* can also provide the binning function, and binned data is provided to the host computer **12**.

Referring now to FIG. **3A**, in which like elements of FIG. **2** are again provided having like reference designations, a second exemplary embodiment of a switching/processing circuit **140** includes a cross point switch circuit **142** coupled to the move data busses **104***a*–**104***k*. The cross point switch **142** is coupled to a memory **144**. The memory **144** is coupled to four DSP processors **146***a*–**146***d*, which are coupled to the switching/processing bus **108**.

In operation, the cross point switch **142** allows formatted data provided by the move data busses **104***a*–**104***k* to first be stored in the memory **144**. The formatted data stored within the memory **144** is withdrawn from the memory **144** by the DSP processors **146***a*–**146***d*. In particular, formatted data corresponding to one or more test instruments (not shown) is withdrawn from the memory **144** by one or more DSP processors. The DSP processors **146***a*–**146***d* provide the processed data on the switching/processing bus **108** to the TCIO circuit **24**. The size, speed and other characteristics of the memory **144** are selected in accordance with a variety of factors including but not limited to cost, speed, and bandwidth. While four DSP processors **146***a*–**146***d* are shown in the exemplary embodiment of FIG. **3A**, it should be recognized that fewer or more than four DSP processor can be used. The number of DSP processors to use is selected in accordance with a variety of factors including but not limited to cost, speed, function to be performed, and the application in which this circuit is used.

In an alternate embodiment, in addition to the processing function, the DSP processors **146***a*–**146***d* can also provide the binning function, and binned data is provided to the host computer **12**.

Referring now to FIG. **3B**, in which like elements of FIG. **2** are again provided having like reference designations, a third exemplary embodiment of a switching/processing circuit **160** includes DSP processors **162***a*–**162***k* coupled directly to respective move data busses **104***a*–**104***k*. The DSP processors **162***a*–**162***k* are also coupled to the switching/processing bus **108**.

In operation, the DSP processors **162***a*–**162***k* receive measure data directly from the move data busses **104***a*–**104***k* and provide the processed data to the switching processing bus **108**.

The number of the DSP processors **162***a*–**162***k* is selected in accordance with the number of move data busses **104***a*–**104***k* coupled thereto, i.e., the number of test instruments. While eleven DSP processors **162***a*–**162***k* are shown in the exemplary embodiment of FIG. **3B**, it will be appreciated that fewer than or more than eleven DSP processors can be used. The particular DSP characteristics are selected in accordance with a variety of factors including but not limited to cost, speed, function to be performed, and application in which this circuit is used.

In an alternate embodiment, in addition to the processing function, the DSP processors **162***a*–**162***k* can also provide the binning function, and binned data is provided to the host computer **12**.

Referring now to FIG. **3C**, in which like elements of FIG. **2** are again provided having like reference designations, a fourth exemplary embodiment of a switching/processing circuit **180** includes DSP processors **182***a*–**182***k*. The DSP processors **182***a*–**182***k* are coupled to a memory **184**. The memory **184** is coupled to the switching/processing bus **108**.

In operation, the DSP processors **182***a*–**182***k* receive measure data directly from the move data busses **104***a*–**104***k* and provide the processed data to the memory **108**. The memory **184** serves as a buffer. The memory **184** then provides the processed data to the switching/processing bus **108**.

The number of the DSP processors **182***a*–**182***k* is selected in accordance with the number of move data busses **104***a*–**104***k* coupled thereto, i.e., the number of test instruments. While eleven DSP processors **182***a*–**182***k* are shown in the exemplary embodiment of FIG. **3C**, it will be appreciated that fewer than or more than eleven DSP processors can be used. The particular DSP characteristics are selected in accordance with a variety of factors including but not limited to cost, speed, function to be performed, and the application in which this circuit is used. The size, speed and other characteristics of the memory **184** are selected in accordance with a variety of factors including but not limited to cost, speed and bandwidth.

In an alternate embodiment, in addition to the processing function, the DSP processors **182***a*–**182***k* can also provide the binning function, and binned data is provided to the host computer **12**.

Referring now to FIG. **4**, an exemplary process **200** for testing an electronic circuit includes a first step **202** at which one or more test instruments are initialized. As described above, the initialization can include a variety of setup functions, including, but not limited to setup of D/A converters within the source circuit (e.g., **34***a*, FIG. **2**), setup of amplifier gains within the source circuit, setup of filters within the source circuit, setup of A/D converters within the measure circuit (e.g., **38***a*', FIG. **2**), setup of amplifier gains within the measure circuit, and setup of filters within the measure circuit. The step **202** can also include providing source data to the one or more test instruments, the source data converted to one or more source signals by the one or more test instruments as described above.

At step **204**, measure data associated with a DUT is captured by a test instrument. As described above, the DUT provides one or more measure signals (e.g., **42***a*–**42***d*, FIG. **2**) in response to one or more source signals (e.g., **40***a*–**40***k*, FIG. **2**). The one or more measure signals are converted by the test instrument (e.g., by the measure circuit **34***a*'–**34***k*', FIG. **2**) to measure data and stored accordingly in one or more measure memories (e.g., the measure memories, **36***a*'–**36***k*' of FIG. **2**).

At step **204**, the process waits until all of the measure data associated with the test instrument has been completed. When the measure data has been completely captured, the process continues at step **208**.

At step **208**, the measure data is provided to a bus master circuit associated with the test instrument that has completed measurement, for example, the bus master circuit **102***a* of FIG. **2**. As described above, the data is provided to the bus master circuit upon initiation by the test instrument, the initiation commencing when the test instrument is ready. It should be recognized that each of the bus master circuits (e.g., **102***a*–**102***k*, FIG. **2**) can be initiated at different times.

At step **210**, the measure data is packaged by the bus master circuit, thereby providing packaged data. As described above, the packaged data includes, but is not limited to, a header and a data packet corresponding to the measured data. The header can include a variety of information associated with the data packet, including, but not limited to an extended header descriptor that describes the type of processing that is to be performed upon the data package.

In an alternate embodiment, the packaged data includes only a data packet corresponding to the measure data.

At step **212**, the packaged data is automatically transferred by the bus master circuit to a switching/processing circuit. The switching processing circuit is described above in FIGS. **3**–**3**C.

At step **214**, the packaged data is routed by the switching/processor circuit either to a DSP processor or to a memory for temporary storage until a processor is free to operate upon the packaged data.

At step **216**, the switching/processing circuit processes the packaged data to provide processed data, the processing responsive to the header. As described above, the amount of processed data can be less than the amount of measured data corresponding to the data packet. For example, the measured data can correspond to 1024 samples of a time domain analog signal, the processing can be an FFT, and the output of the processing can correspond to 512 samples of a frequency domain FFT output.

In an alternate embodiment, the type of processing is not responsive to the header, instead being pre-determined. In another alternate embodiment, the type of processing is communicated to the switching processor circuit by a host computer.

At step **218**, the processed data is transferred to the host computer. The host computer can be of a type described as the host computer **12** in FIGS. **1** and **2**.

At step **220**, the host computer processes the processed data to provided binned data. The binned data is provided by a comparison of the processed data to pre-determined thresholds. As described above, the amount of the binned data can be less than the amount of the processed data. For example the processed data can correspond to 512 samples of a frequency domain FFT output, the thresholds can correspond to FFT magnitude thresholds, and the binned data can correspond to one FFT frequency bin value that exceed the pre-determined thresholds.

In another alternate embodiment, the binned data provided at step **220** is generated along with the processed data at step **216** and the binned data is transferred to the host computer at step **218**.

As described above, the test program in accordance with this method, and each of the functional steps **202**–**220**, can be categorized according to the function categories of setup **222**, execution **224**, and binning **230**. The function categories of data move **226** and processing **228** are shown parenthetically to indicate that these functions, generated by the test program for a conventional ATE, are instead automatically provided with this invention. It should be appreciated that a subjective decision is made with respect to which steps to include in each of the categories. Thus, each of the categories may include different steps that those shown in FIG. **4**. For example, it is possible to consider step **214** as part of the data move category **226** rather than the processing category **228**. Similarly, steps **208** and **210** could be considered as part of the execution category **224** rather than as part of the data move category **226**.

All references cited herein are hereby incorporated herein by reference in their entirety.

Having described preferred embodiments of the invention, it will now become apparent to one of ordinary skill in the art that other embodiments incorporating their concepts may be used. It is felt therefore that these embodiments should not be limited to disclosed embodiments, but rather should be limited only by the spirit and scope of the appended claims.

What is claimed is:

1. A circuit testing apparatus, comprising:

at least one test instrument;

a bus master circuit coupled to said at least one test instrument, said bus master circuit adapted to receive measure data from said at least one test instrument and, upon initiation from said at least one test instrument, to package the measure data to provide packaged data; and

a switching/processing circuit coupled to receive the packaged data from said bus master circuit, to process the packaged data to provide processed data, and adapted to communicate the processed data to a host computer.

2. The circuit testing apparatus of claim 1, wherein the packaged data includes:

a data packet corresponding to the measure data; and

an extended header descriptor portion of the header associated with the data packet, the extended header descriptor including a processing identifier corresponding to a type of processing to be performed on the data packet.

3. The circuit testing apparatus of claim 1, wherein said switching/processing circuit comprises:

a switching circuit coupled to said bus master circuit and adapted to route the packaged data; and

at least one processor coupled to the switching circuit, the at least one processor adapted to receive the packaged data and to process the packaged data to provide the processed data.

4. The circuit testing apparatus of claim 3, wherein the at least one processor is adapted to receive the packaged data and to separate a data packet and an extended header descriptor from the packaged data, the at least one processor further adapted to process the data packet in response to the extended header descriptor to provide the processed data.

5. The circuit testing apparatus of claim 3, wherein said switching/processing circuit further includes a first buffer memory coupled to the switching circuit.

6. The circuit testing apparatus of claim 5, wherein the first buffer memory is coupled to the at least one processor.

7. The circuit testing apparatus of claim 1, wherein said bus master circuit is coupled to a measure memory, the measure memory adapted to receive the measure data from said at least one test instrument and to provide the measure data to said bus master circuit.

8. The circuit testing apparatus of claim 1, wherein the processed data includes binned data.

9. The circuit testing apparatus of claim 1, wherein said bus master circuit and said switching/processing circuit are on separate circuit boards.

10. The circuit testing apparatus of claim 1, wherein said bus master circuit and said switching/processing circuit are on one or more circuit boards included in the at least one test instrument.

11. A method for testing a device under test, comprising:

capturing measure data in a test instrument, the measure data corresponding to one or more functional tests of the device under test performed by the test instrument;

initiating, with the test instrument, operations upon the measure data, the operations including:

packaging the measure data with a bus master circuit as packaged data; and

transferring the packaged data to a switching/processing circuit.

12. The method of claim 11, wherein the initiating further includes:

routing the packaged data to a processor within the switching/processing circuit; and

processing the packaged data with the processor to provide processed data.

13. The method of claim 12, wherein the packaging includes packaging the measure data with the bus master circuit as packaged data, the packaged data including a data header, and wherein the processing includes processing the packaged data with the processor, the processing responsive to the header.

14. The method of claim 12, wherein the processing comprises:

pre-determining pass and fail thresholds associated with the one or more functional tests; and

associating the data packet with the pass and fail thresholds to provide binned data.

15. The method of claim 12, wherein the packaging includes:

generating a data packet corresponding to the measure data; and

generating an extended header descriptor associated with the data packet, the extended header descriptor including a processing identifier corresponding to a type of processing to be performed on the data packet.

16. An method of manufacturing an electronic circuit, comprising:

fabricating an electronic circuit; and

testing the electronic circuit with a circuit testing apparatus, wherein the testing comprises:

capturing measure data in a test instrument, the measure data corresponding to one or more functional tests of the electronic circuit performed by the test instrument;

initiating, with the test instrument, operations upon the measure data, the operations including:

packaging the measure data with a bus master circuit as packaged data; and

transferring the packaged data to a switching/processing circuit.

17. The method of claim 16, wherein the initiating further includes:

routing the packaged data to a processor within the switching/processing circuit; and

processing the packaged data with the processor to provide processed data.

18. The method of claim 17, wherein the packaging includes packaging the measure data with the bus master circuit as packaged data, the packaged data including a data header, and wherein the processing includes processing the packaged data with the processor, the processing responsive to the header.

19. The method of claim 17, wherein the processing comprises:

pre-determining pass and fail thresholds associated with the one or more functional tests; and

associating the data packet with the pass and fail thresholds to provide binned data.

20. The method of claim 17, wherein the packaging includes:

generating a data packet corresponding to the measure data; and

generating an extended header descriptor associated with the data packet, the extended header descriptor including a processing identifier corresponding to a type of processing to be performed on the data packet.

* * * * *

UNITED STATES . TRICT COURT, CENTRAL DISTRICT OF .LIFORNIA
CIVIL COVER SHEET

I (a) PLAINTIFFS (Check box if you are representing yourself □)
TERADYNE, INC.

DEFENDANTS
XYRATEX LTD. and XYRATEX INTERNATIONAL, INC.

(b) Attorneys (Firm Name, Address and Telephone Number. If you are representing yourself, provide same.)

James M. Dowd, WILMER CUTLER PICKERING HALE AND DORR, LLP
350 S. Grand Ave., Suite 2100, Los Angeles, CA 90071
Telephone: (213) 443-5300

Attorneys (If Known)

II. BASIS OF JURISDICTION (Place an X in one box only.)

□ 1 U.S. Government Plaintiff
☑ 3 Federal Question (U.S. Government Not a Party)
□ 2 U.S. Government Defendant
□ 4 Diversity (Indicate Citizenship of Parties in Item III)

III. CITIZENSHIP OF PRINCIPAL PARTIES - For Diversity Cases Only
(Place an X in one box for plaintiff and one for defendant.)

	PTF	DEF		PTF	DEF
Citizen of This State	□ 1	□ 1	Incorporated or Principal Place of Business in this State	□ 4	□ 4
Citizen of Another State	□ 2	□ 2	Incorporated and Principal Place of Business in Another State	□ 5	□ 5
Citizen or Subject of a Foreign Country	□ 3	□ 3	Foreign Nation	□ 6	□ 6

IV. ORIGIN (Place an X in one box only.)

☑ 1 Original Proceeding
□ 2 Removed from State Court
□ 3 Remanded from Appellate Court
□ 4 Reinstated or Reopened
□ 5 Transferred from another district (specify):
□ 6 Multi-District Litigation
□ 7 Appeal to District Judge from Magistrate Judge

V. REQUESTED IN COMPLAINT: JURY DEMAND: ☑ Yes □ No (Check 'Yes' only if demanded in complaint.)

CLASS ACTION under F.R.C.P. 23: □ Yes ☑ No

☑ **MONEY DEMANDED IN COMPLAINT: $** To be determined at trial.

VI. CAUSE OF ACTION (Cite the U.S. Civil Statute under which you are filing and write a brief statement of cause. Do not cite jurisdictional statutes unless diversity.)
Patent Infringement

VII. NATURE OF SUIT (Place an X in one box only.)

OTHER STATUTES

□ 400 State Reapportionment
□ 410 Antitrust
□ 430 Banks and Banking
□ 450 Commerce/ICC Rates/etc.
□ 460 Deportation
□ 470 Racketeer Influenced and Corrupt Organizations
□ 480 Consumer Credit
□ 490 Cable/Sat TV
□ 810 Selective Service
□ 850 Securities/Commodities/Exchange
□ 875 Customer Challenge 12 USC 3410
□ 890 Other Statutory Actions
□ 891 Agricultural Act
□ 892 Economic Stabilization Act
□ 893 Environmental Matters
□ 894 Energy Allocation Act
□ 895 Freedom of Info. Act
□ 900 Appeal of Fee Determination Under Equal Access to Justice
□ 950 Constitutionality of State Statutes

CONTRACT

□ 110 Insurance
□ 120 Marine
□ 130 Miller Act
□ 140 Negotiable Instrument
□ 150 Recovery of Overpayment & Enforcement of Judgment
□ 151 Medicare Act
□ 152 Recovery of Defaulted Student Loan (Excl. Veterans)
□ 153 Recovery of Overpayment of Veteran's Benefits
□ 160 Stockholders' Suits
□ 190 Other Contract
□ 195 Contract Product Liability
□ 196 Franchise

REAL PROPERTY

□ 210 Land Condemnation
□ 220 Foreclosure
□ 230 Rent Lease & Ejectment
□ 240 Torts to Land
□ 245 Tort Product Liability
□ 290 All Other Real Property

TORTS
PERSONAL INJURY

□ 310 Airplane
□ 315 Airplane Product Liability
□ 320 Assault, Libel & Slander
□ 330 Fed. Employers' Liability
□ 340 Marine
□ 345 Marine Product Liability
□ 350 Motor Vehicle
□ 355 Motor Vehicle Product Liability
□ 360 Other Personal Injury
□ 362 Personal Injury-Med Malpractice
□ 365 Personal Injury-Product Liability
□ 368 Asbestos Personal Injury Product Liability

IMMIGRATION

□ 462 Naturalization Application
□ 463 Habeas Corpus-Alien Detainee
□ 465 Other Immigration Actions

TORTS
PERSONAL PROPERTY

□ 370 Other Fraud
□ 371 Truth in Lending
□ 380 Other Personal Property Damage
□ 385 Property Damage Product Liability

BANKRUPTCY

□ 422 Appeal 28 USC 158
□ 423 Withdrawal 28 USC 157

CIVIL RIGHTS

□ 441 Voting
□ 442 Employment
□ 443 Housing/Accommodations
□ 444 Welfare
□ 445 American with Disabilities - Employment
□ 446 American with Disabilities - Other
□ 440 Other Civil Rights

PRISONER PETITIONS

□ 510 Motions to Vacate Sentence Habeas Corpus
□ 530 General
□ 535 Death Penalty
□ 540 Mandamus/ Other
□ 550 Civil Rights
□ 555 Prison Condition

FORFEITURE / PENALTY

□ 610 Agriculture
□ 620 Other Food & Drug
□ 625 Drug Related Seizure of Property 21 USC 881
□ 630 Liquor Laws
□ 640 R.R. & Truck
□ 650 Airline Regs
□ 660 Occupational Safety /Health
□ 690 Other

LABOR

□ 710 Fair Labor Standards Act
□ 720 Labor/Mgmt. Relations
□ 730 Labor/Mgmt. Reporting & Disclosure Act
□ 740 Railway Labor Act
□ 790 Other Labor Litigation
□ 791 Empl. Ret. Inc. Security Act

PROPERTY RIGHTS

□ 820 Copyrights
☑ 830 Patent
□ 840 Trademark

SOCIAL SECURITY

□ 861 HIA (1395ff)
□ 862 Black Lung (923)
□ 863 DIWC/DIWW (405(g))
□ 864 SSID Title XVI
□ 865 RSI (405(g))

FEDERAL TAX SUITS

□ 870 Taxes (U.S. Plaintiff or Defendant)
□ 871 IRS-Third Party 26 USC 7609

CV09-2580

FOR OFFICE USE ONLY: Case Number: ____________

AFTER COMPLETING THE FRONT SIDE OF FORM CV-71, COMPLETE THE INFORMATION REQUESTED BELOW.

UNITED STATES DISTRICT COURT, CENTRAL DISTRICT OF CALIFORNIA
CIVIL COVER SHEET

VIII(a). IDENTICAL CASES: Has this action been previously filed in this court and dismissed, remanded or closed? ☑ No □ Yes
If yes, list case number(s):

VIII(b). RELATED CASES: Have any cases been previously filed in this court that are related to the present case? □ No ☑ Yes
If yes, list case number(s): Potentially related to Case No. CV08-04545-SJO (PLAx) (See Accompanying Notice of Potential Related Case)

Civil cases are deemed related if a previously filed case and the present case:

(Check all boxes that apply)
□ A. Arise from the same or closely related transactions, happenings, or events; or
□ B. Call for determination of the same or substantially related or similar questions of law and fact; or
☑ C. For other reasons would entail substantial duplication of labor if heard by different judges; or
□ D. Involve the same patent, trademark or copyright, and one of the factors identified above in a, b or c also is present.

IX. VENUE: (When completing the following information, use an additional sheet if necessary.)

(a) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** named plaintiff resides.
□ Check here if the government, its agencies or employees is a named plaintiff. If this box is checked, go to item (b).

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
	Massachusetts

(b) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** named defendant resides.
□ Check here if the government, its agencies or employees is a named defendant. If this box is checked, go to item (c).

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
	Xyratex Ltd. resides in the United Kingdom; Xyratex International, Inc. resides in Santa Clara County

(c) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** claim arose.
Note: In land condemnation cases, use the location of the tract of land involved.

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
Los Angeles County	

*** Los Angeles, Orange, San Bernardino, Riverside, Ventura, Santa Barbara, or San Luis Obispo Counties**
Note: In land condemnation cases, use the location of the tract of land involved

X. SIGNATURE OF ATTORNEY (OR PRO PER): **Date** April 14, 2009

Notice to Counsel/Parties: The CV-71 (JS-44) Civil Cover Sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law. This form, approved by the Judicial Conference of the United States in September 1974, is required pursuant to Local Rule 3-1 is not filed but is used by the Clerk of the Court for the purpose of statistics, venue and initiating the civil docket sheet. (For more detailed instructions, see separate instructions sheet.)

Key to Statistical codes relating to Social Security Cases:

Nature of Suit Code	Abbreviation	Substantive Statement of Cause of Action
861	HIA	All claims for health insurance benefits (Medicare) under Title 18, Part A, of the Social Security Act, as amended. Also, include claims by hospitals, skilled nursing facilities, etc., for certification as providers of services under the program. (42 U.S.C. 1935FF(b))
862	BL	All claims for "Black Lung" benefits under Title 4, Part B, of the Federal Coal Mine Health and Safety Act of 1969. (30 U.S.C. 923)
863	DIWC	All claims filed by insured workers for disability insurance benefits under Title 2 of the Social Security Act, as amended; plus all claims filed for child's insurance benefits based on disability. (42 U.S.C. 405(g))
863	DIWW	All claims filed for widows or widowers insurance benefits based on disability under Title 2 of the Social Security Act, as amended. (42 U.S.C. 405(g))
864	SSID	All claims for supplemental security income payments based upon disability filed under Title 16 of the Social Security Act, as amended.
865	RSI	All claims for retirement (old age) and survivors benefits under Title 2 of the Social Security Act, as amended. (42 U.S.C. (g))

James M. Dowd (SBN 259578)
Wilmer Cutler Pickering Hale and Dorr LLP
350 S. Grand Avenue, Suite 2100
Los Angeles, CA 90071
Telephone: (213) 443-5300
Facsimile: (213) 443-5400

ORIGINAL

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

TERADYNE, INC.

PLAINTIFF(S)

v.

XYRATEX' LTD. and XYRATEX INTERNATIONAL, INC.

DEFENDANT(S).

CASE NUMBER

CV09-2580 MMM (CWx)

SUMMONS

TO: DEFENDANT(S): XYRATEX LTD. and XYRATEX INTERNATIONAL, INC.

A lawsuit has been filed against you.

Within 20 days after service of this summons on you (not counting the day you received it), you must serve on the plaintiff an answer to the attached ☑ complaint □ ______________ amended complaint □ counterclaim □ cross-claim or a motion under Rule 12 of the Federal Rules of Civil Procedure. The answer or motion must be served on the plaintiff's attorney, James M. Dowd, whose address is 350 S. Grand Ave., Suite 2100, Los Angeles, CA 90071. If you fail to do so, judgment by default will be entered against you for the relief demanded in the complaint. You also must file your answer or motion with the court.

Clerk, U.S. District Court

Dated: APR 14 2009

By: Natalie Gongora
Deputy Clerk

(Seal of the Court)

[Use 60 days if the defendant is the United States or a United States agency, or is an officer or employee of the United States. Allowed 60 days by Rule 12(a)(3)].

James M. Dowd (SBN 259578)
Wilmer Cutler Pickering Hale and Dorr LLP
350 S. Grand Avenue, Suite 2100
Los Angeles, CA 90071
Telephone: (213) 443-5300
Facsimile: (213) 443-5400

COPY

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

TERADYNE, INC.

PLAINTIFF(S)

v.

XYRATEX LTD. and XYRATEX INTERNATIONAL, INC.

DEFENDANT(S).

CASE NUMBER

CV09-2580 MMM (CWx)

SUMMONS

TO: DEFENDANT(S): XYRATEX LTD. and XYRATEX INTERNATIONAL, INC.

A lawsuit has been filed against you.

Within 20 days after service of this summons on you (not counting the day you received it), you must serve on the plaintiff an answer to the attached ☑ complaint □ ______________ amended complaint □ counterclaim □ cross-claim or a motion under Rule 12 of the Federal Rules of Civil Procedure. The answer or motion must be served on the plaintiff's attorney, James M. Dowd, whose address is 350 S. Grand Ave., Suite 2100, Los Angeles, CA 90071. If you fail to do so, judgment by default will be entered against you for the relief demanded in the complaint. You also must file your answer or motion with the court.

Clerk, U.S. District Court

Dated: APR 14 2009

By: NATALIE LONGORIA

Deputy Clerk

(Seal of the Court)

SEAL OF THE U.S. DISTRICT COURT CENTRAL DIST. OF CALIFORNIA

[Use 60 days if the defendant is the United States or a United States agency, or is an officer or employee of the United States. Allowed 60 days by Rule 12(a)(3)].

1198