A universal IC test system is designed to function both as an event tester and a cyclized tester. The universal test system is comprised of an event tester for testing DUT by test vectors produced based on event data derived directly from simulation of design data of DUT produced in an EDA environment. A cyclized tester for testing DUT by test vectors produced based on test data formulated in a cyclized format in which each test vector is defined by a waveform, a test rate, and a timing with respect to the test rate, in electronics for applying the test vector to DUT and comparing a response output of DUT, and means for changing a tester mode between an event tester mode and a cyclized tester mode thereby testing DUT either by the event tester or the cyclized tester, or by both testers.
Fig. 2C

Conventional Tester Data Structure
Vector + Cycle + Waveform

Simulation Dump Out

VCD Description
Sa1 = 20nS
Ra1 = 40nS
Sa2 = 150nS
Ra2 = 220nS
Sa3 = 230nS
Ra3 = 280nS
Rb1 = 20nS
Rb2 = 60nS
Rb3 = 220nS

Event Base Description
Pin Sa Sa1 20
Ra1 40
Sa2 150
Ra2 220
Sa3 230
Ra3 280
Pin Sb Rb1 20
Sb1 40
Rb2 60
Sb2 110
Rb3 220

rate (TS1, 100nS); rate (TS2, 110nS); rate (TS3, 200nS);

Sa.wavekind(TS1, RZ); Sa.wavekind(TS2, NRZ); Sa.wavekind(TS3, XOR);
Sb.wavekind(TS1, XORINV); Sb.wavekind(TS2, NRZ); Sb.wavekind(TS3, NRZINV);

Sa.timing(TS1, T1, 20nS); Sa(TS1, T2, 40nS);
Sa.timing(TS2, T1, 50nS);
Sa.timing(TS3, T1, 30nS); Sa.timing(TS3, T2, 70nS); Sa.timing(TS3, T3, 10nS);
Sb.timing(TS1, T1, 40nS); Sb.timing(TS1, T2, 60nS); Sb.timing(TS1, T3, 20nS);
Sb.timing(TS2, T1, 10nS);
Sb.timing(TS3, T1, 10nS);
<table>
<thead>
<tr>
<th>Structure of Fig. 3</th>
<th>Structure of Fig. 4A</th>
<th>Structure of Fig. 4B</th>
<th>Structure of Fig. 4C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Combined electronics</td>
<td>Maximum Double electronics</td>
<td>Moderate Double electronics</td>
<td>Moderate Double electronics</td>
</tr>
<tr>
<td>1 driver</td>
<td>2 drives</td>
<td>1 driver</td>
<td>1 drivers</td>
</tr>
<tr>
<td>1 comparator</td>
<td>2 comparators</td>
<td>1 comparator</td>
<td>2 comparators</td>
</tr>
<tr>
<td>1 Mux</td>
<td>1 relay</td>
<td>3 relays</td>
<td>1 relay</td>
</tr>
<tr>
<td>1 I/O pin or 1 input &amp; 1 output pin</td>
<td>1 input &amp; 1 output pin</td>
<td>1 I/O pin</td>
<td>1 I/O pin Double test rate</td>
</tr>
<tr>
<td>Fixed test rate</td>
<td>Fixed test rate</td>
<td>Fixed test rate</td>
<td>Double test rate</td>
</tr>
<tr>
<td>vs Test Rate</td>
<td>vs Test Rate</td>
<td>vs Test Rate</td>
<td>vs Test Rate</td>
</tr>
</tbody>
</table>

- Mirror image to apply cyclized vector and evaluate in event form
ARCHITECTURE AND DESIGN OF UNIVERSAL IC TEST SYSTEM

[0001] This is a continuation-in-part of U.S. application Ser. No. 10/150,777 filed May 20, 2002.

FIELD OF THE INVENTION

[0002] This invention relates to an architecture of a semiconductor IC test system, and more particularly, to a universal semiconductor IC test system that can work in both an event environment and a cycle based IC test environment. The event based testing utilizes test data in an event form thereby enabling a direct use of design simulation data produced in an electronic design automation (EDA) environment while the cycle based test system produces test patterns based on test data defined in a cyclized form according to traditional timesets and wave-groups.

BACKGROUND OF THE INVENTION

[0003] In testing semiconductor IC devices by a semiconductor IC test system (IC tester or tester), the basic procedure of functional testing of a semiconductor IC device contains creation of input stimuli for the IC device, application of these stimuli to the IC device and comparison of the output of the IC device with expected results by strobing the outputs at predetermined times. Such input stimuli and strobes are collectively called a test pattern or test vector and are traditionally generated based on test data in a cyclized form. Such a traditional test system is sometimes called a cycle based test system or a cyclized tester where various data for producing the input stimuli and strobes is defined relative to corresponding test cycles (tester rates or timesets). Today, cycle based test systems are commonly used in mass production of IC devices.

[0004] While the cyclized testers pose no significant problems when used in the production process of the IC devices, the cyclized testers have serious constraints that the original IC design data cannot be used directly. Thus, it is considered that the cyclized testers are not quite fit for a research and development process of IC designs. Recently, the assignee of this invention has proposed an event based test system (event tester) to overcome this constraint. The following descriptions provide differences and advantages of the event tester over the cyclized tester.

[0005] Today, IC design is made under an electronic design automation (EDA) environment where IC designers develop new ICs with use of a high-level language such as Verilog or VHDL and simulate the design by behavioral, gate-level Verilog/VHDL simulators. Such design simulation is targeted to check the functionality and performance before the design is fabricated into silicon IC. To use the simulation data, the traditional IC test systems require conversion of design simulation data into cyclized form, such as WGL (Waveform Generation Language) or STIL (Standard Test Interface Language) format.

[0006] As noted above, the present day semiconductor IC test systems are single or multiple timesets (cyclized or cycle based) machines, in which data are associated with each pin (T. Karamaki et al. "Model of 100 MHz test station for high speed LSI test system", IEEE Int. Test Conf., pp. 139-145, 1978, T. Sudo, “A 100 MHz tester—challenge to new horizon of testing high speed LSI” IEEE Int. Test Conf., pp. 362-368, 1979, and M. Shimizu et al., “100 MHz algorithmic pattern generator for memory testing”, IEEE, Int. Test Conf., pp. 56-67, 1980). The variations are that in some test systems, these timesets can be switched on the fly; in other test systems, waveform formaters are available to generate complex waveforms; and third variation is that relay functionalities are incorporated for shared resource machines in order to share or distribute the timing generators to the desired pins (S. Bisset, “The development of a tester per pin VLSI test system architecture”, IEEE Int. Test Conf., pp. 151-155, 1983).

[0007] Because of these limitations in today’s test systems, IC testing requires a different environment than the original IC design and simulation environment.

[0008] From the user’s point of view, the above limitations cause following problems: (i) vector conversion consumes extensive time, server and disk capacities, and is very error-prone, (ii) cyclization of vectors makes multiple close domain devices untestable, and (iii) due to a finite number of resources such as timesets, waveform groups, timing generators, etc., there arises tester limitations.

[0009] While the primary IC design and simulation environment is event oriented, because of the above limitations, the test environment is cyclized. Hence, it is a foreign environment from the original IC design environment and causes difficulties during testing and debug of ICs. Also, to get to the test environment, engineers are required to reformat simulation testbenches and re-run simulation to capture the cyclized vectors that are required for testing. Essentially, it makes the test data very different than the original design and simulation data. The engineers translate vectors into another intermediate format such as STIL (Standard Test Interface Language, IEEE standard 1450, 1999) and WGL (Waveform Generation Language) to create test program as illustrated in FIG. 1A.

[0010] FIG. 1A shows a basic process involved in today’s cyclized test systems for testing the IC by using the design testbench data (simulation vectors). In this example, the left side indicates a design domain 10 where the design testbench is executed through a logic simulator thereby producing input/output event data of the design, i.e. VCD (Value Change Dump) in Verilog or VHDL simulation. The right side in FIG. 1A indicates a test domain 20 where the designed IC device is tested by the IC tester with use of the test vectors produced based on the VCD data produced in the design domain.

[0011] As shown in FIG. 1A, in the conventional technology involving the cycle based test system, the test program development requires (i) extracting event based simulation vectors (VCD format) at step 11, (ii) converting the simulation vectors into cycle based vectors at step 12, and (iii) converting the cycle based vectors into tester’s format such as WGL, STIL at step 21 and/or a proprietary format such as TDL ("Test Description Language" by Advantest Corporation, Tokyo, Japan) at step 22. The resultant test vectors in the cycle format are used to create a test
program at step 23. This test program is executed on a cyclized tester to evaluate the response outputs of IC.

[0012] Converting the IC simulation vectors from the VCD format to the cyclized format is time consuming, complex and error-prone. This problem is further compounded by the fact that every tester has its own unique and proprietary language and vector format (e.g., TDL and IPAT by Advantest). Subsequently, all this effort in vector conversion becomes extremely time consuming and costly. The time required to process these vectors has grown proportionately with the size of the vectors, taking as much as a month to process all VCD files into cyclized format.

[0013] This lengthy process also impedes the ability to process new or improved vectors quickly, thereby slowing the test and debug process. Moreover, the very act of converting original IC simulation vectors into a tester’s cyclized format endangers the accuracy of data. This results in errors and test vectors that are no longer simulation-faithful. All these problems result in additional time and cost.

[0014] As noted above, to overcome this problem, the event based test system (event tester) has been proposed which is structured based on the concept illustrated in FIG. 1B. The design testbench data 11, i.e. VCD, produced in the design domain 10 can be directly used by an event based IC tester 30 in the test domain 20. This is a stark difference from the traditional process of FIG. 1A where the test program development requires conversion of event based simulation vectors into cycle based vectors, capturing them into the VCD format, and converting them into tester’s format.

[0015] Although the cyclized testers have various drawbacks as described above, such problems are not very serious when the testers are used in the production stage of IC devices. This is because a need of direct use of the design data in the production stage testing is relatively low. Further, because the cyclized testers are widely used today, test resources such as test programs, test result data, etc., are accumulated for cyclized testers, which will be important assets for users. On the other hand, event testers are extremely useful in the IC development and debug phase. Accordingly, there is a need in the industry for an IC test system that has a universal architecture so that it is capable of functioning as an event tester as well as a cyclized tester.

SUMMARY OF THE INVENTION

[0016] It is, therefore, an object of the present invention to provide a universal semiconductor IC test system which is capable of functioning both as an event tester and/or a cyclized tester.

[0017] It is another object of the present invention to provide a new tester architecture which is capable of direct use of design simulation data produced in an electronic design automation (EDA) environment while making use of test resources produced under traditional test systems.

[0018] It is a further object of the present invention to provide a universal tester that can debug and validate initial IC devices in an event domain while it can perform production testing in a cyclized domain.

[0019] The present invention is a universal test system for testing an IC device under test (DUT). The universal test system includes:

[0020] an event tester for testing DUT by test vectors produced based on event data derived directly from simulation of design data of DUT produced in an EDA (electronic design automation) environment wherein each event in the event data is defined by an event time indicating a time length of the event from a predetermined point and an event type indicating a type of change at the event,

[0021] a cyclized tester for testing DUT by test vectors produced based on test data formulated in a cyclized format in which each test vector is defined by a waveform, a test rate, and a timing with respect to the test rate,

[0022] a pin-electronics for applying the test vector to DUT and comparing a response output of DUT; and

[0023] means for changing an operation mode between an event mode and a cyclized mode thereby testing DUT either by the event tester or the cyclized tester, or by both the event tester and the cyclized tester.

[0024] In one aspect of the universal test system, the event tester and the cyclized tester are separately established. The event tester is comprised of an event memory for storing the event data which defines the event time and the event type of each event, an event generator for generating the test vectors based on the event data from the event memory. The cyclized tester is comprised of a rate generator for producing the test rate, at least one memory for storing pattern data, timing data, and waveform data, a timing generator for producing timing signals based on the timing data, and a waveform formatter for producing the test vectors based on the timing signals, pattern data and waveform data.

[0025] In the universal test system, the means for changing the operation mode includes a relay to switch a pair of driver and comparator in the pin-electronics. Alternatively, the means for changing the testing mode includes two or more relays to independently switch the driver and the comparator in the pin-electronics so that one test channel of the universal test system is able to test two pins of DUT at the same time.

[0026] In a further aspect of the universal test system, shared resources are used for event operation and cyclized operation. The universal test system is comprised of a memory for storing the event data for use in the event test mode and the test data for use in the cyclized tester mode, a memory controller for managing operations of the memory depending on whether the DUT is tested by the event tester mode or the cyclized tester mode, an event generator for generating the test vectors based on the event data from the memory, a rate generator for producing the test rate in the cyclized tester mode, a timing generator for producing timing signals based on the timing data in the cyclized tester mode, a waveform formatter for producing the test vectors based on the timing signals, pattern data and waveform data in the cyclized tester mode, and means for selecting the test vectors from the event generator or the waveform formatter to apply the test vectors to DUT.

[0027] According to the present invention, since the universal test system functions both as an event tester and as a cyclized tester, it is possible to debug and validate IC devices in the design and simulation environment, i.e., in the event environment as well as to test IC devices in the mass
production process in the cyclized environment without discontinuity in the existing test methods. Because it performs as the event tester, the universal test system of the present invention allows testing and debug of ICs without diverting from the environment in which the IC device was designed. Thus, the universal test system avoids test pattern conversion such as the WGL or STIL format and uses the design simulation data as-is. Further, because it performs as the cyclized tester, the universal test system of the present invention allows production testing of IC devices with use of existing test methods and test resources.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1A is a flow diagram showing a process of test program development for conventional cycle based test systems, and FIG. 1B is a flow diagram showing a process of testing obtained through the event based test system.

[0029] FIG. 2A is a schematic block diagram showing an architecture of the event based test system, FIG. 2B is a schematic diagram showing an architecture in the cycle based test system, and FIG. 2C is a diagram for comparing an example of descriptions in the cycle format and in the event format for the same test vectors.

[0030] FIG. 3 is a schematic diagrams showing an example of overall system architecture of the universal test system according to the present invention.

[0031] FIG. 4A is a schematic diagram showing an alternative implementation of overall system architecture of the universal test system of the present invention, FIGS. 4B and 4C(1)-4C(2) show further alternative implementation of the universal test system of the present invention.

[0032] FIG. 5 is a schematic diagram showing an example of architecture in a universal test system incorporating a plurality of different tester modules based on the concept of the present invention.

[0033] FIG. 6 is a schematic block diagram showing an overall hardware structure of the universal test system of the present invention.

[0034] FIG. 7 is a schematic diagram showing an example of overall system software architecture of the universal test system of the present invention.

[0035] FIG. 8 is a diagram showing a file structure and data flow from GUI to tester hardware in the universal test system of the present invention.

[0036] FIG. 9 is a schematic flow diagram showing an IC testing process using the universal test system of the present invention.

[0037] FIG. 10 shows a table comparing features of various implementation of the universal test system of the present invention shown in FIGS. 3 and 4A-4C.

DETAILED DESCRIPTION OF THE INVENTION

[0038] The present invention is now described in more detail with reference to the accompanying drawings. The assignee of this invention has disclosed a concept of an event based test system (event tester) in previous patent applications such as U.S. application Ser. Nos. 09/406,300 and 09/340,371. It should be noted that the technologies disclosed in these prior applications are inventors' prior knowledge but not prior art against the present invention under the patent law. This application discloses a universal test system having the event tester architecture as well as the traditional cyclized tester architecture.

[0039] The event tester architecture is oriented towards IC design and simulation environment. In this architecture, the tester uses changes in the signal values (events) identical to the events as observed in the IC design simulation. Secondly, events on each pin are treated independently rather than cyclizing them according to timesets. Thus, the event tester eliminates the vector conversions and need to develop test programs.

[0040] In other words, the IC testing environment in the event tester is the same as the original IC design environment; engineers are not required to change the simulation testbenches into the conventional ATE cyclized format, and VCD vectors need not be converted into STIL or other formats. The engineers develops only one simulation testbench that can be used without any changes for both IC design simulation as well as IC testing. It means that there should be no cyclization of testbenches, no vector conversion from VCD to other formats and subsequently no need to develop complicated specialized test programs.

[0041] The basic architecture of the event based test system (event tester) is shown in FIG. 2A. The architecture of the cycle based test system (cyclized tester) is shown in FIG. 2B. Further, FIG. 2C illustrates a specific example of descriptions of the test vector in the cycle format and the event format for clearly showing the difference between the two.

[0042] In FIG. 2A, the event based test system is configured by an event memory 40 for storing event data, an event generator 41 for generating events based on the event data, a driver 42 for supplying the test vectors to the device under test (DUT) and a comparator 50 for comparing an output of the DUT with expected data. Typically, the event data includes event time data indicating timing of the event and event type data indicating type of the event. The event time data is formed, for example, with an integer multiple of a reference clock period (integral part data) and a fraction of the reference clock period (fractional part data). In the semiconductor test industry, the driver and comparator are collectively called a pin-electronics.

[0043] In FIG. 2B, the cycle based test system is configured by a rate generator 43 for producing tester rates (test cycles), a pattern memory for storing pattern data, a timing memory 45 for storing timing data, a waveform memory for storing waveform (action) data, a timing generator 47 for producing timing signals based on the timing data, a waveform formatter 48 for producing a test pattern based on the timing signals, pattern data and waveform data, a driver 49 for supplying the test vectors to the DUT, and a comparator 51 for comparing an output of the DUT with expected data. As noted above, the driver and comparator are collectively called a pin-electronics.

[0044] As clear from FIGS. 2A and 2B, in the event tester, the rate generator 43, timing generator 47, pattern memory 44, waveform memory 46 and timing memory 45 of the cyclized tester are eliminated, and instead, the event memory 40 and event generator units 41 are used. The event
memory 40 contains the events as observed in the Verilog/VHDL simulation. Due to the need to store timed-events, an event tester requires a significantly large amount of memory compared to a cycled tester. The event generator 41 converts these events into actions (for driver to apply a test vector) by using the associated timing as recorded in the Verilog/VHDL simulation. Through the driver 42, these actions are applied to the DUT and the response of the DUT is compared by the comparator 50 against the IC simulation values (expected values) to detect a fault.

[0045] By eliminating the rate and timing generators, pattern memory, waveform memory and timing memory, the event tester architecture effectively eliminates the need of cycled the vectors and translation into other formats such as WGL or STIL. The event memory 40 in FIG. 2A stores events as they were recorded in the IC simulation. Thus, each test vector (action) will be created by driving an event (data “0” or “1”) with its timing. In the cycled tester of FIG. 2B, each test vector will be created by driving a specified waveform (action) based on a pattern data (“0” or “1”) at a timing specified by a timeset (test cycle).

[0046] Thus, the event tester architecture accomplishes the objective that cycled and vector translation process should be eliminated from the testing; and that the test environment should be the same as the IC design environment. The event tester architecture allows IC design environment to extend from simulation to the physical testing of ICs. As the event tester environment is the same as the design simulation environment, it simplifies debug and validation of the first silicon drastically.

[0047] To explain the difference between the cycle format and event format more clearly, brief comparison is made between the two formats in FIG. 2C. In this example, the same test vector having waveforms 131, which is typically designated simulation VCD, is described in the event format and the cycle format. The simulation VCD of the waveform 131 is stored in a file 137. The description in this file 137 is shown in VCD description 139 which is set/reset type event based format showing the changes in the input and output of the designed IC. The waveforms 131 illustrate two pins Sa and Sb.

[0048] In the event format, an “event” means each change point of the waveform 131. In this example, the event data describing the waveforms 131 is formed of set and reset edges Sa, Sb, Ran and Rbn (event types), and their timings (event times). Typically, the event time is defined by a time length from a previous event (delta time) or a specified reference point (absolute time). Such an event time is described in the event data by an integer multiple of a reference clock period and a fraction of the reference clock period.

[0049] For producing the waveform 131 in the semiconductor test system using the cycle format, the test data must be divided into test cycles (tester rate), waveforms (types of waveforms, and their edge timings), and pattern values. An example of such descriptions is shown in the center and left of FIG. 2C. In the cycle based description, test pattern data 135 and timesets 133 are shown in the left part of FIG. 2C, a test pattern is divided into each test cycle (TS1, TS2 and TS3) to define the waveforms and timings (delay times) for each test cycle.

[0050] An example of data descriptions for such waveforms, timings and test cycles is shown in timing data 136. An example of logic “1”, “0” or “Z” of the waveforms is shown in the pattern data 135. For example, in the timing data 136, the test cycle is described by “rate” to define time intervals between test cycles, and the waveform is described by RZ (return to zero), NRZ (non-return to zero) and XOR (exclusive OR). Further, the timing of each waveform is defined by a delay time from a predetermined edge (ex. start edge of each rate) of the corresponding test cycle.

[0051] As seen in FIG. 2C, the event base description 138 is identical to the design simulation results (VCD) while the cycle base description requires the time-sets of various types of descriptions which are remote from the original design simulation result. Accordingly, the event tester allows testing and debugging of ICs without diverting from the environment in which the IC was designed. As noted above, the traditional cycled tester requires conversion of the design simulation data into a cycled form, such as the WGL or STIL format.

[0052] Thus, the traditional cycled tester is unfit for testing in the early stage of semiconductor production which requires seamless interactions between the design data and test vector generation. However, the cycled tester is still useful in the later stage of semiconductor production, i.e., in the mass production testing which does not require direct use of the design data. Further, since a large number of cycled testers are in use today, users are familiar with the cycled testers in using them, and developing test programs, etc., and have many test resources (test programs, test analysis data, device mounting boards, etc.). Therefore, it is advantageous to develop a tester that works in both the event domain and cycled domain.

[0053] FIG. 3 shows an example of preferable architecture of the universal test system in which hardware electronics is so configured to function in both the event domain and cycled domain. For example, a memory 82 functions as an event memory in the event tester mode as well as a memory for storing pattern data, timing data and waveform data in the cycled tester mode. Data from the memory 82 is provided to an event generator 86 and/or a timing generator 88 depending upon a tester mode signal 81.

[0054] It should be noted that this combination of elements is not straightforward due to completely disjoint natures of the event and cycled domains. For example, one cannot simply combine the memories used in the event tester and the cycled tester to create the memory 82 in FIG. 3 because the different nature of data arrangement, read/write and data access requirements. Thus, an actual implementation requires special data structures, manners of access to the data, and the like to establish the memory 82 with optimum cost and efficiency. Further, an additional controller circuit 80 that works as a memory management unit is required as shown in FIG. 3. The tester mode signal 81 determines the operation of the memory management unit 80 that in turn determines the operation of the memory 82.

[0055] Similarly, the event generator 86 and the waveform generator 90 have completely disjoint requirements. While the event generator 86 requires a delay generator circuit for a coarse and fine delay generation for event timing, the waveform generator 90 requires a timing generator that requires synchronized pattern, timing and waveform data to generate waveform. A rate generator 84 can be basically the same as that shown in FIG. 2B. A multiplexer 92 is provided.
to select the data between the event domain and the cycle domain in response to the tester mode signal 81. The advantage of the structure of FIG. 3 is that it allows the operation in both the event and cyclized domains, however, it uses significantly less electronics in comparison with a simple sum of electronics in FIGS. 2A and 2B.

[0056] As is known in the art, an IC device has either (1) input pin, (2) output pin, or (3) I/O pin that switches between input and output modes. Since the pin electronics in the example of FIG. 3 is configured by a set of driver and comparator connected to one another, one tester pin can test one I/O pin with the test rate of used in the test configuration of FIG. 2. Alternatively, one tester pin can test either one of input pin or output pin with the test rate of FIG. 2.

[0057] FIG. 4A shows an alternative implementation for a universal test system (universal tester) that can act as an event tester and a cyclized tester in accordance with the present invention. This example shows a simple method to obtain a universal tester that works in both the event domain and cyclized domain in which different types of pin-electronics and relays 55 are used to switch the mode of operation (event domain or cycle domain). In this example, one pin-electronics has a set of driver 42 and comparator 50 and the other pin-electronics has a set of driver 49 and comparator 51. Thus, one tester pin can test two I/O pins with the test rate of FIG. 2, or both one input pin and one output pin with the test rate of FIG. 2. Either mechanical relays or solid-state relays can be used as the relays 55; the advantage of using solid-state relays is fast and ease of operation. The example of FIG. 4A is the most straightforward approach but it requires double pin-electronics and hardware is equivalent to the combined hardware of FIGS. 2A and 2B.

[0058] Hence, the another alternative approach is a design that employs multiple relays 55 and 57 to utilize pin-electronics in a separate driver and comparator mode, which is illustrated in FIG. 4B. The relays 55 and 57 switch between the event domain and cyclized domain for transmitting test vectors to the driver and receiving comparator outputs. A relay 59 is used to isolate an IC device pin from the driver output when necessary. As the driver and comparator become independent in this design, the driver 49 can continuously apply test vectors to an IC device input pin without the need to intermediate switching to comparator; while the comparator 51 can continuously compare the response from an IC device output pin. Hence, the test rate that is twice the test rate of FIG. 4A (also FIG. 2) can be achieved.

[0059] With the design of FIG. 4B, one tester pin can also be used to test two device pins (one input pin and one output pin). There are two possibilities: (i) apply an event test vector to an input pin of the IC device under test through the driver 49 and compare the response from the output pin of the IC device in a cyclized format through comparator 51; or (ii) apply a cyclized test vector to the input pin of the DUT through the driver 49 and compare the response in the event format through comparator 51. Alternatively, one tester pin can test one I/O pin with the test rate of FIG. 2.

[0060] Another possible approach is to configure the design of FIG. 4B into that of FIG. 4A, an example of which is shown in FIGS. 4C(1) and 4C(2). In the example of FIG. 4C(1), the driver 49 continuously applies test vector and the comparator 51 continuously evaluates the response (comparator 52 is off by relay 58). In this manner, one tester pin can test either one I/O pin, or both one input pin and one output pin, with the test rate twice the test rate of FIG. 2. Alternatively, the driver 49 applies test vector and the comparator 52 evaluates the response (comparator 51 is off by relay 57). In this arrangement, one tester pin can test one I/O pin with the test rate of FIG. 2.

[0061] The example of FIG. 4C(1) applies the test vector in the event form and evaluate the response in the cyclized form. FIG. 4C(2) is a mirror image of the implementation of FIG. 4C(1) where the cyclized test vector produces the test vector and the event tester evaluates the response of the IC. The comparators 51 and 52 and relays 57 and 58 are controlled in the same manner described above.

[0062] FIG. 10 shows a table summarizing the features of implementation of FIGS. 3 and 4A-4C described above. A number of other variations are also possible that provide some additional benefit at additional cost such as an extra driver, comparator or relay. The basic limitation of the designs shown in Figs. 4A-4C is that it requires additional hardware electronics because of having both event tester hardware and cyclized tester hardware and hence, it is costly compared to that of FIG. 3.

[0063] Using the design of FIGS. 3 and 4A-4C, a universal test system can be developed in a modular form as shown in FIG. 5. In the modular structure of FIG. 5, the universal test system includes a plurality of tester modules 101-106 and a control module 107. In such an approach, each module can be made-up of either a dedicated cyclized module, or a dedicated event module, or a combined module according to FIGS. 3 or 4A-4C. For example, a tester module 101 is solely configured as a cyclized tester (cyclized module such as FIG. 2B) while a tester module 102 is solely configured as an event tester (event module such as FIG. 2A). Some tester modules can be mixed type (such as FIG. 3 or FIGS. 4A-4C). In FIG. 5, the tester modules 103-106 will be constituted such mixed (universal) type modules.

[0064] Such tester modules may be freely configured in the universal test system, for example, by inserting in slots of the test system to establish a test system of desired characteristics. For example, different types of tester modules such as for mixed (analog/logic) signal testing, for high speed and low pin count testing, for low speed and high pin count testing, or the like will be selected. As having a plurality of tester modules with different architectures and capabilities, the universal tester of FIG. 5 is advantageously applied to testing a complex IC such as a system-on-a-chip (SoC) having a plurality of functional cores.

[0065] In this example, a controller (CPU) 112 is provided for each tester module, although other arrangements, such as each controller controls two or more tester modules or one controller controls an overall system, is also possible. Each controller (CPU) 112 controls the data flow in the tester module, application of test vectors to DUT, response comparison, scheduling of various tasks for DUT as well as monitoring the status of DUT. The controllers 112, 112, are connected with one another and also connected to a main system CPU 115 through system buses 111 and 113. In the tester module 107, a synchronized clock unit 109 and an arbitration unit 110 are provided to promote data transfer to
and from the main system CPU 115 and the controllers 112, 112, in the tester modules 101-106.

Prior to the test process, the main system CPU 115 installs the individual test data (event format or cycled format) and distributes the test data to the tester modules 112, 112, through a multiple distribution control 114 and a user interface for software verification 116. The main system CPU 115 controls an overall procedure in the design validation (preferably by event domain) and/or production testing (preferably by cycled domain).

Various hardware components in the universal test system are illustrated in FIG. 6. The host computer 151 that contains various software components is connected to a tester controller 122 via a PCI bus card 117. In this implementation, the PCI bus is used although any other interface can also be used. The controller 122 resides within a test head chassis 125. The test head chassis 125 also contains a cycled testor pin card 118, an event tester pin card 119, a mixed event/cycled pin card 120 and a device power supply (DPS) pin card 121. The above components in the test head chassis 125 are powered by a power supply 129. A backplane card 126 within the test head chassis 125 provides a simple method to connect various pin cards 118-120, DPS pin card 121 and tester controller 122.

Further, through pogo-pins 123 and a test fixture 127 having various connectors and pins, the pin cards 118-121 and have a bi-directional access to a loadboard 124 on which a device under test (DUT) is placed. Thus, when a user applies a command or data, it is interpreted by the embedded software on the host computer 151 and appropriate message/data is passed to the tester controller 122 and pin cards 118-121. The tester controller 122 and the pin cards 118-121 apply these commands/data to DUT via the test fixture 127, pogo-pins 123 and loadboard 124 (vice-versa if the user obtains data from DUT).

The overall system software architecture is shown in FIG. 7. In the example of FIG. 7, the overall software architecture includes a host computer 151, interface software 152, data interpretation and management software 156 and universal tester hardware 161. The host computer 151 and the universal tester hardware 161 can be remotely located from one another and communicate directly or through a public communication network such as through a Web server 155 or a dedicated communication network. The interface software 152 includes a GUI (Graphical User Interface) 153 and a user-side communication classes 154. Through the host computer 151, a user accesses the GUI 153 and the universal tester hardware 161. The software of the GUI 153 allows to specify various commands and necessary files as well as allows data editing and manipulation during testing.

The data interpretation and management software 156 primarily contains three components: (i) a middleware 158 for data processing and interpretation, (ii) a kernel 159 for mediating between hardware and software, and (iii) a tester application server 157 for coordinating the communication from the user (via GUI 153) to the middleware 158 or vice versa. As noted above, the communication from the GUI 153 to the application server 157 can take place either by direct link or through the communication network such as the Web access.

The middleware 158 includes various software components for data processing and interpretation and produces various types of data as will be described with reference to FIG. 8. The middleware 158 interprets the test data (event format and cycle format) from the host computer 151 and provides it to the kernel 159 in the form of time index and binary (event tester) or in the form of time-sets of waveform, pattern data and timings (cycled tester). Similar to the test data, the middleware 158 also interprets user’s commands such as run-test, move-event, add/delete-event etc., via the GUI 153 and the application server 157 and provides start/stop, setup address, power supply sequences etc. to the tester hardware 161 via the kernel 159 (or vice-versa).

The kernel 159 mediates between the hardware and software by, for example, converting values in the data from the middleware 158 into physical voltage/current levels for the tester hardware 161. Similarly, the kernel 159 mediates between the hardware and software by interpreting the voltage/current values obtained by the tester hardware 161 in logic 0, 1, or Z values and supplies these values to the middleware 158.

FIG. 8 shows a file structure and data flow from the GUI 153 to the tester hardware 161 in the universal test system of the present invention. As shown in FIG. 8, a test plan file 163, a test parameter file 164, a pin file 165 and a socket file 166 are specified by the user through the host computer 151 and the GUI 153 of FIG. 7.

The test plan file 163 contains the type of tests to be performed such as contact test, DC/AC measurements, functional tests, etc. The parameter file 164 specifies various parameters such as Volt (voltage-out “high”), Vol (voltage-out “low”), Volt (voltage-in “low”), Vol (voltage-in “high”), I1 (current-in “high”), I2 (current-in “low”), power-supply (PS), etc. When using the universal test system as a cycled tester, the parameter file 164 further specifies time-sets (rates or cycles) and timings, and other parameters as shown in FIG. 2C. The pin file 165 specifies tester logic pin assignment given by the user. The socket file 166 specifies test socket pin assignment given by the user.

Based upon the user’s commands via the GUI 153, the application server 157 passes this data to the middleware 158. The middleware 158 interprets this data and based upon it, the kernel 159 appropriately switch-on or off the hardware drivers to apply this data to the IC device under test (DUT). The middleware 158 and the kernel 159 interpret the data based on whether the universal test system is in the event domain or cycled domain and mediate the data to fit for the tester hardware.

In the example of FIG. 8, through data interpretation and processing, the middleware 158 produces various types of data, i.e., test plan 167, test measurement 169, logical PIN/PS 170 and tester PIN/PS 172. Based on the data from the test plan file 163, the test plan 167 describes, for example, an order in which test should be applied, such as contact test, then AC test, and then DC test. The test 168 describes, for example, the nature of test and associated test vectors that will be applied to the DUT.

The measurement 169 defines the type of measurement, such as AC or DC and associated voltage/current values based on the data from the parameter file 164. Based
on the data from the pin file 165, the logic PIN/PS 170 describes the logical pin assignment with use of a pin list stored in the middleware and specifies I/O pins (input stimulus, response output) and power supply pins (VDD, GND). Based on the data from the socket file 166 and logical PIN/PS 170, the tester PIN/PS 172 identifies physical connection between device pin and the tester channel, i.e., which I/O pin is connected to which test channel, and which VDD/GND pin is connected to which tester channel.

[0078] In the example of FIG. 8, for mediating between the tester hardware 161 and the middleware 158, the kernel 159 produces data, kernel FuncMeas 174 and kernel PIN/PS 176. The kernel FuncMeas 174 describes measurement voltage and current level by interpreting (mapping) the data values in the measurement 169 into actual voltages and current levels. The kernel PIN/PS 176 describes commands to activate/de-activate the specific tester channels based upon the values from the tester PIN/PS 172 so that the tester hardware acts based upon these commands.

[0079] It should be noted that FIG. 8 illustrates only one file of each type for simplicity, such as one testplan file (with one test), one parameter file, one measurement and one pin-map file (for pin and power supply). In an actual test system, however, there are multiple files of each type that constitute a complete test.

[0080] FIG. 9 is a schematic flow diagram showing an IC testing process using the universal test system of the present invention. Any type of test vectors, either in the event format or in the cyclized format such as WGL/STIL can be used. In the example of FIG. 9, the designers produce design data, typically behavioral or RTL level data describing the function of the circuits in the intended IC. Such design data is converted to device netlist 182 indicating sets of components connected by signals. Based on simulation conditions 181 and the device netlist 182, the designer runs a simulator 183, such as a Verilog/VHDL simulator, with use of test benches produced when designing the IC to simulate the functions of the IC. The results of this simulation are input and output value sets, i.e., verilog VCD, which is in an event format.

[0081] To function as an event tester, the Verilog VCD is used to directly produce the event data through a compiler 192. To function as a cyclized tester, VCD is converted to a cyclized format such as WGL or STIL through a data converter 190. Alternatively, an ATPG (automatic test pattern generator) 184 can be run with the device net list to generates test vectors. The universal tester 180 receives the test data either in the event format or cyclized format from the compiler 192.

[0082] A silicon IC 185 is produced based on the device netlist 182 which may be mounted on a loadboard 186 of the universal tester 180. The universal tester 180 applies test vectors in the event format to the silicon IC 185 by directly using the VCD produced by the simulator 183 or in the cyclized format converted from the design data. The universal tester 180 applies the test vectors to the silicon IC 185 and evaluates response outputs therefrom. Accordingly, the test result is obtained and stored in a pass/fail file 187 for failure analysis.

[0083] As described in the foregoing, according to the present invention, since the universal test system functions both as event tester and as cyclized tester, it is possible to debug and validate IC devices in the design and simulation environment, i.e., in the event environment as well as to test IC devices in the mass production process in the cyclized environment without discontinuity in the existing test methods. Because it performs as the event tester, the universal test system of the present invention allows testing and debug of ICs without diverting from the environment in which the IC device was designed. Thus, the universal test system avoids test pattern conversion such as the WGL or STIL format and uses the design simulation data as is. Further, because it performs as the cyclized tester, the universal test system of the present invention allows production testing of IC devices with use of existing test methods and resources.

[0084] Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that various modifications and variations may be made without departing from the spirit and scope of the present invention. Such modifications and variations are considered to be within the purview and scope of the appended claims and their equivalents.

What is claimed is:

1. A universal test system for testing an IC device under test (DUT), comprising:
   - an event tester for testing DUT by test vectors produced based on event data derived directly from simulation of design data of DUT produced in an EDA (electronic design automation) environment wherein each event in the event data is defined by an event time indicating a time length of the event from a predetermined point and an event type indicating a type of change at the event;
   - a cyclized tester for testing DUT by test vectors produced based on test data formulated in a cyclized format in which each test vector is defined by a waveform, a test rate, and a timing with respect to the test rate;
   - a pin-electronics for applying the test vector to DUT and comparing a response output of DUT; and
   - means for changing a tester mode between an event tester mode and a cyclized tester mode thereby testing DUT either by the event tester or the cyclized tester, or both the event tester and the cyclized tester.

2. A universal test system as defined in claim 1, wherein the event tester is comprised of:
   - an event memory for storing the event data which defines the event time and the event type of each event;
   - an event generator for generating the test vectors based on the event data from the event memory;
   - wherein the cyclized tester is comprised of:
     - a rate generator for producing the test rate;
     - at least one memory for storing pattern data, timing data, and waveform data;
     - a timing generator for producing timing signals based on the timing data; and
     - a waveform formatter for producing the test vectors based on the timing signals, pattern data and waveform data.

3. A universal test system as defined in claim 1, wherein the event time in the event tester is defined by an integer
multiple of a reference clock period (integral part data) and
a fraction of the reference clock period (fractional part data).

4. A universal test system as defined in claim 2, wherein
the predetermined point for defining the time length by the
event time is an event immediately prior to a current event
so that the event time expresses a delta time between two
adjacent events.

5. A universal test system as defined in claim 2, wherein
the predetermined point for defining the time length by the
event time is a start point of operation so that the event time
expresses an absolute time of the event.

6. A universal test system as defined in claim 2, wherein
the means for changing the tester mode includes a relay to
switch a pair of a driver and a comparator in the pin-
electronics.

7. A universal test system as defined in claim 2, wherein
the means for changing the tester mode includes relays to
independently switch a driver and a comparator in the pin-
electronics so that one test channel of the universal test
system tests two pins of DUT.

8. A universal test system as defined in claim 1, comprising:

a memory for storing the event data for use with the event
tester and the test data for use with the cyclized tester;
a memory controller for managing operations of the
memory depending on whether the DUT is tested by the
event tester mode or the cyclized tester mode;
an event generator for generating the test vectors based on
the event data from the memory;
a rate generator for producing the test rate in the cyclized
tester mode;
a timing generator for producing timing signals based on
the timing data in the cyclized tester mode;
a waveform formatter for producing the test vectors based
on the timing signals, pattern data and waveform data in
the cyclized tester mode; and
means for selecting the test vectors from the event gen-
erator or the waveform formatter to apply the test
vectors to DUT.

9. A universal test system for testing an IC device under
test (DUT), comprising:

a host computer for controlling an overall operation of the
universal test system;
a plurality of tester modules for generating test vectors
and supplying the test vectors to DUT and evaluating
response outputs of the DUT;
wherein one of the tester modules is configured by a
combination of an event tester and a cyclized tester
where the event tester tests DUT by test vectors pro-
duced based on event data derived directly from simu-
lation of design data of DUT produced in an EDA
(electronic design automation) environment and the
cyclized tester tests DUT by test vectors produced
based on test data formulated in a cyclized format in
which each test vector is defined by a waveform, a test
rate, and a timing with respect to the test rate.

10. A universal test system as defined in claim 9, wherein
at least one of the tester modules is configured solely by the
event tester and at least another one of the tester modules is
configured solely by the cyclized tester.

11. A universal test system as defined in claim 9, where at
least one of the tester modules is configured a universal type
tester which can operate either in an event form or in a
cyclized form.

12. A universal test system for testing an IC device under
test (DUT), comprising:

a host computer for controlling an overall operation of the
universal test system;
interface software for interfacing the host computer and
the universal test system, the interface software includ-
ing graphic user interface (GUI) establishing an input
means, command control and viewer for monitoring
and editing test vectors for the universal test system;
data interpretation and management software for inter-
preting and managing data from the host computer
through the interface software; and
universal test system hardware having an event tester
mode for testing DUT by test vectors produced based on
event data derived directly from simulation of design data of DUT produced in an EDA (electronic
design automation) environment and a cyclized tester
mode for testing DUT by test vectors produced based on
test data formulated in a cyclized format in which
each test vector is defined by waveform, a test rate, and
a timing with respect to the test rate.

13. A universal test system as defined in claim 12, wherein
the interface software and the data interpretation and man-
agement software communicate directly or through a public
communication network or a dedicated communication
network.

14. A universal test system as defined in claim 12, wherein
the data interpretation and management software includes
middleware for data processing and interpretation, and ker-
nel for mediating data values between the event tester
hardware and the middleware.

15. A universal test system as defined in claim 12, wherein
the middleware interprets information specified by a user
and produces data including types of test, an order of tests
and test parameters for supplying the data to the event tester
hardware through the kernel.

16. A universal test system as defined in claim 12, wherein
the middleware interprets information specified by a user
and produces data including I/O pin and power supply pin of
DUT in addition to types of test, order of tests and test
parameters for supplying the data to the universal test
system hardware through the kernel.