(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 February 2007 (08.02.2007)

PCT

(10) International Publication Number
WO 2007/016699 A2

(51) International Patent Classification:
G01R 31/3183 (2006.01)

(21) International Application Number:
PCT/US2006/030417

(22) International Filing Date: 2 August 2006 (02.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/195,180 2 August 2005 (02.08.2005) US

(71) Applicant *(for all designated States except US)*: **SYNPLICITY, INC.** [US/US]; 600 W. California Street, Sunnyvale, CA 94086 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **NG, Chun, Kit** [US/US]; 12859 Nw Majestic Sequoia Way, Portland, OR 97229 (US). **LAROUCHE, Mario** [CA/US]; 11989 Nw Coleman Drive, Portland, OR 97229 (US).

(74) Agents: **SCHELLER, James, C.** et al.; BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

WO 2007/016699 A2

(54) Title: METHOD AND SYSTEM FOR DEBUG AND TEST USING REPLICATED LOGIC

(57) **Abstract:** A method and system for debug and test using replicated logic is described. A representation of a circuit is compiled. The circuit includes a replicated portion and delay logic to delay inputs into the replicated portion. The circuit may also include trigger logic and clock control logic to enable execution of the replicated portion of the circuit to be paused when a trigger condition occurs. The compiled representation of the circuit may be programmed into a hardware device. A debugger may then be invoked. One or more triggering signals are selected. For each selected triggering signal, one or more states are selected to setup a trigger condition. The hardware device may then be run. The replicated portion of the circuit will be paused when the trigger condition occurs. The states of registers in the replicated portion of the circuit and the sequence of inputs that led to the trigger condition are recorded. This recorded data may then used to generate a test to be run on a software simulator when the circuit is modified.

METHOD AND SYSTEM FOR DEBUG AND TEST USING REPLICATED LOGIC

PRIORITY INFORMATION

[0001] This application is a continuation-in-part (CIP) of application no. 11/112,092, filed April 22, 2005.

TECHNICAL FIELD

[0002] Embodiments of the invention relate to the field of debugging and testing integrated circuits, and more specifically to debugging and testing integrated circuits using replicated logic.

BACKGROUND

[0003] For the design of digital circuits, designers often employ computer aided techniques. Standard languages, such as Hardware Description Languages (HDLs), have been developed to describe digital circuits to aid in the design and simulation of complex digital circuits. As device technology continues to advance, various product design tools have been developed to adapt HDLs for use with newer devices and design styles.

[0004] After the HDL code is written and compiled, the design of an integrated circuit (IC) or a system which includes multiple ICs must be verified to be correct. Continually advancing processing technology and the corresponding explosion in design size and complexity have led to verification problems for complex circuit designs, such as Application Specific Integrated Circuits (ASICs) that are difficult to solve using traditional simulation tools and techniques.

[0005] As a result, some designers build prototype boards using multiple ICs such as field programmable gate arrays (FPGAs) to verify their ASIC designs. However, there are still problems with debugging the hardware design. When an error is detected during debug, designers may attempt to tap signals of interest from the circuit and use a logic analyzer to determine the cause of the error. However, this is a difficult process and is often not effective, especially in the case of intermittent errors. Errors that have already occurred are often difficult to repeat and reconstruct.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

Fig. 1 is a block diagram illustrating a suitable computing environment in which certain aspects of the invention may be practiced.

Fig. 2 is a flow chart illustrating an embodiment of a method of the invention.

Fig. 3 illustrates an example of a circuit section implementing an embodiment of the invention.

Fig. 4 illustrates an example of clock control logic according to an embodiment of the invention.

DETAILED DESCRIPTION

[0006] Embodiments of a system and method for debugging and testing using replicated logic are described. In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

[0007] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. Thus, the appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0008] Fig. 1 is a block diagram illustrating a suitable computing environment in which certain aspects of the illustrated invention may be practiced. In one embodiment, the method described above may be implemented on a computer system 100 having components that include a processor 102, a memory 104, an Input/Output (I/O) device 106, a data storage device 112, and a network interface 110, coupled to each other via a bus 108. The components perform their conventional functions known in the art and provide the means for implementing the system of the invention. Collectively, these components represent a broad category of hardware systems, including but not limited to general purpose computer systems, mobile or wireless computing systems, and specialized packet forwarding devices. It is to be appreciated that various components of computer system 100 may be rearranged, and that certain implementations of the present invention may not require nor include all of the above components. Furthermore, additional components may be included in system 100, such as additional processors (e.g., a digital signal processor), storage devices, memories (e.g. RAM, ROM, or flash memory), and network or communication interfaces.

[0009] As will be appreciated by those skilled in the art, the content for implementing an embodiment of a method of the invention, for example, computer program instructions, may be provided by any machine-readable media which can store data that is accessible by system 100, as part of or in addition to memory, including but not limited to cartridges, magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read-only memories (ROMs), and the like. In this regard, the system 100 is equipped to communicate with such machine-readable media in a manner well-known in the art.

[0010] It will be further appreciated by those skilled in the art that the content for implementing an embodiment of the method of the invention may be provided to the system 100 from any external device capable of storing the content and communicating the content to the system 100. For example, in one embodiment, the system 100 may be connected to a network, and the content may be stored on any device in the network.

[0011] Fig. 2 is a flow chart illustrating an embodiment of a method of the invention. At 200, a representation of a circuit is compiled. In one embodiment, the compilation generates a first register transfer (RTL) netlist. In one embodiment, the circuit is described by a text representation by writing Hardware Description Language (HDL) source code descriptions of the elements of the circuit. In one embodiment, the circuit is described by a netlist representation.

[0012] The representation of the circuit is then input into a compiler. One example of a compiler is a logic synthesis compiler, which is typically a computer program that operates on a general purpose computer system, although in some embodiments, the computer system may be a dedicated, special purpose computer system. After compilation, a RTL netlist may be generated. The RTL netlist usually shows registers and other logic interconnected to show the flow of data through the circuit.

[0013] In one embodiment of the invention, the RTL netlist is mapped to a target architecture. The target architecture is typically determined by a supplier of the integrated circuit (IC). Examples of target architectures include field programmable gate arrays (FPGAs) and complex programmable logic devices from vendors such as Altera, Lucent Technologies, Advanced Micro Devices (AMD), and Lattice Semiconductor. The mapping operation converts the RTL level description of the desired circuit into the equivalent circuit implemented using building blocks of the target architecture. A technology specific netlist is generated. Conventional place and route software tools may then be used to create a design of circuitry in the target architecture.

[0014] For debugging purposes, IC designers may build prototype boards using multiple ICs such as FPGAs to verify their designs. For example, after the compilation, mapping, and place and route operations, the circuit may be programmed into FPGAs to create a prototype of the design. The FPGAs can then be tested to determine any problem areas in the design.

[0015] When a problem area is found in the design, the designer may further analyze the problem by selecting that portion of the circuit to replicate and by inserting trigger logic. One or more signals may be selected for triggering. These selected signals may be used later as trigger signals to enable a trigger condition. Triggering logic may then be inserted into the circuit. One or more controllers for the triggering logic may also be inserted into the circuit. A portion of the circuit may then be selected for replication and the selected portion of the circuit is replicated. This replication may include a replication of the logic elements, the input signals, and the output signals of the selected portion of the circuit. In one embodiment, each register in the replicated portion of the circuit is connected together in a scan chain, such as a JTAG chain. This scan chain allows information from the registers, such as their states, to be scanned out during debug.

[0016] In one embodiment of the invention, clock signals are also replicated. Clock control logic is inserted to control the clock signals. The clock control logic allows the clock to the replicated logic block to be paused to stop the replicated logic from executing when certain conditions are present and to allow for single-stepping through the replicated logic to analyze an error. The designer may select a breakpoint to pause the clock to the replicated portion of the circuit when certain conditions are present. For example, the designer may choose values for the outputs or inputs that will pause the clock. This allows the designer to analyze the selected logic more carefully when certain problem conditions are present.

[0017] Delay logic may be inserted to delay inputs into the replicated portion of the circuit. The length of the delay may be selected by the circuit designer. The delay logic allows an error observed in the selected portion of the circuit to be analyzed after the error is seen to occur since the error will reappear in the replicated portion of the circuit at a later time.

[0018] The representation of the circuit may then be recompiled. In one embodiment, the compilation generates a second RTL netlist. Then, the mapping and place and route operations may be performed using the second RTL netlist to implement the circuit in a target architecture, such as a FPGA. In one embodiment of the invention, a synthesis operation is performed to generate an application specific integrated circuit (ASIC) from the second RTL netlist. A circuit with replicated logic is produced that allows a circuit designer to analyze a problem area in the design. The designer may invoke a debugger to assist in the debugging of the circuit.

[0019] At 202, one or more triggering signals are selected. These signals may be selected from the set of signals chosen previously, as discussed above. At 204, one or more states of each selected triggering signal are set to setup a triggering condition. At 206, when the trigger condition occurs, data is recorded. This data includes one or more states of one or more registers and the sequence of inputs that led to the trigger condition. The replicated logic may be stepped clock by clock with the value of the inputs recorded at every clock. This input stream represents the sequence of inputs leading to the trigger condition that is being analyzed. The states of the registers in the replicated logic may also be recorded by using the scan chain implemented as described above.

[0020] In one embodiment, the recorded data may be converted into a format that is compatible with a software simulator. For example, if the software simulator is a VHDL or a Verilog simulator, then the recorded information may be converted to VHDL or Verilog, respectively. In one embodiment, the formatted recorded data may then be input into the software simulator to debug logic problems in the circuit design. In one embodiment, the recorded data may be input into the software simulator to verify that the circuit design remains functional when changes to the logic are made.

[0021] At 208, a test is generated using the data recorded at 206. The test includes a sequence of inputs and internal states of registers that are based on the recorded sequence of inputs that led to the trigger condition and the recorded internal states of registers. The test may then be input into a software simulator and run when the circuit is modified to verify that the circuit remains functional. This test may be run each time the circuit is modified to ensure that the circuit is still working as expected after any modifications.

[0022] Fig. 3 illustrates an example of a section of a circuit 300 implementing an embodiment of the invention. Logic block 302 is a portion of the circuit in the original IC design. Debug of the original IC design revealed a problem with logic block 302. Therefore, original logic block 302 was selected and replicated to enable further analysis of the problem. The original logic block 302 is replicated to produce a replicated logic block 304. Outputs 308 from the original logic block 302 are replicated to produce replicated outputs 310. Inputs 306 may also be replicated.

[0023] Delay logic 312 is inserted to delay inputs 306 into replicated logic block 304. The delay logic includes typical circuit logic and elements, such as inverters, that cause the inputs 306 to arrive at the replicated logic block 304 later in time than the inputs 306 will arrive at the original logic block 302. In this way, an error can be analyzed after the error is seen to occur in the original logic block, since the error will appear in the replicated logic block at a later time.

[0024] Trigger logic 330 is inserted into the circuit to enable the setup of a trigger condition that pauses the replicated portion of the circuit. One or more controllers may also be inserted to control the trigger logic. The trigger logic 330 has two outputs: breakpoint 318 and delay pause 328. Breakpoint 318 enables the clock control logic 314 to stop advancing. Delay pause 328 enables the delay logic 312 to stop advancing.

[0025] Clock control logic 314 is inserted to control the clock signals 322 to the replicated logic block 304. The clock control logic 314 contains typical logic and circuit elements that allow the clock 322 to the replicated logic block 304 to be paused to stop the replicated logic from executing when certain conditions are present. The clock control logic 314 may also allow for single stepping through the replicated logic on a clock by clock basis to analyze an error. The breakpoint 318 may be set to pause the clock when certain conditions are present, such as when the trigger condition occurs.

[0026] Fig. 4 illustrates an example of the clock control logic 314 according to an embodiment of the invention. During normal operation, the system clock 316 that clocks the circuit flows through the latch 400 and acts as the clock 322 to the replicated logic block 304. The breakpoint 318 switches the clock 322 to a latched version of the system clock 316, which can be controlled by clock control signals

320 in order to allow the clock 322 to be paused and single-stepped on a cycle by cycle basis.

[0027] Thus, embodiments of a method and apparatus for debug and test using replicated logic have been described. The above description of illustrated embodiments of the invention, including what is described in the abstract, is not intended to be exhaustive or to limit the invention to the precise forms disclosed. While specific embodiments of, and examples for, the invention are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize. These modifications can be made to the invention in light of the above detailed description. The terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification and the claims. Rather, the scope of the invention is to be determined by the following claims, which are to be construed in accordance with established doctrines of claim interpretation.

CLAIMS

What is claimed is:

1. A method comprising:

compiling a representation of a circuit, the circuit including a replicated portion and delay logic to delay inputs into the replicated portion;

selecting one or more triggering signals;

setting one or more states for each selected triggering signal to setup a trigger condition;

recording data that includes one or more states of one or more registers in the replicated portion of the circuit and a sequence of inputs that led to the trigger condition when the trigger condition occurs; and

generating a test based on the recorded data, the test including the recorded states of the registers and the sequence of inputs that led to the trigger condition.

2. The method of claim 1, wherein the representation of the circuit is written in a hardware description language (HDL).

3. The method of claim 1, wherein generating a test based on the recorded data comprises converting the recorded data to a format compatible with a software simulator.

4. The method of claim 3, wherein converting the recorded data to a format compatible with a software simulator comprises converting the recorded data to a VHDL format to be compatible with a VHDL simulator.

5. The method of claim 3, wherein converting the recorded data to a format compatible with a software simulator comprises converting the recorded data to a Verilog format to be compatible with a Verilog simulator.

6. The method of claim 3, further comprising running the test on the software simulator when the circuit is modified.

7. The method of claim 1, wherein compiling the representation of the circuit comprises compiling the representation of the circuit to generate a register transfer level netlist.

8. The method of claim 7, further comprising mapping the register transfer level netlist to a selected technology architecture.

9. The method of claim 8, further comprising performing a place and route operation to implement the circuit in the selected technology architecture.

10. The method of claim 9, further comprising programming the register transfer level netlist into a programmable hardware device.

11. The method of claim 10, wherein the circuit further includes trigger logic and clock control logic coupled to the replicated portion of the circuit to enable execution of the replicated portion of the circuit to be paused when the trigger condition occurs.

12. The method of claim 11, further comprising running the circuit on the programmable hardware device and pausing the replicated portion of the circuit when the trigger condition occurs.

13. An article of manufacture comprising:

a machine accessible medium including content that when accessed by a machine causes the machine to perform operations including:

compiling a representation of a circuit, the circuit including a replicated portion and delay logic to delay inputs into the replicated portion;

selecting one or more triggering signals;

setting one or more states for each selected triggering signal to setup a trigger condition;

recording data that includes one or more states of one or more registers in the replicated portion of the circuit and a sequence of inputs that led to the trigger condition when the trigger condition occurs; and

generating a test based on the recorded data, the test including the recorded states of the registers and the sequence of inputs that led to the trigger condition.

14. The article of manufacture of claim 13, wherein generating a test based on the recorded data comprises converting the recorded data to a format compatible with a software simulator.

15. The article of manufacture of claim 14, wherein the machine-accessible medium further includes content that causes the machine to perform operations comprising running the test on the software simulator when the circuit is modified.

16. The article of manufacture of claim 13, wherein compiling the representation of the circuit comprises compiling the representation of the circuit to generate a register transfer level netlist.

17. The article of manufacture of claim 16, wherein the machine-accessible medium further includes content that causes the machine to perform operations comprising mapping the register transfer level netlist to a selected technology architecture.

18. The article of manufacture of claim 17, wherein the machine-accessible medium further includes content that causes the machine to perform operations comprising performing a place and route operation to implement the circuit in the selected technology architecture.

19. The article of manufacture of claim 18, wherein the machine-accessible medium further includes content that causes the machine to perform operations comprising programming the register transfer level netlist into a programmable hardware device.

20. The article of manufacture of claim 19, wherein the machine-accessible medium further includes content that causes the machine to perform operations comprising recompiling the representation of the circuit after an analysis of the recorded data.

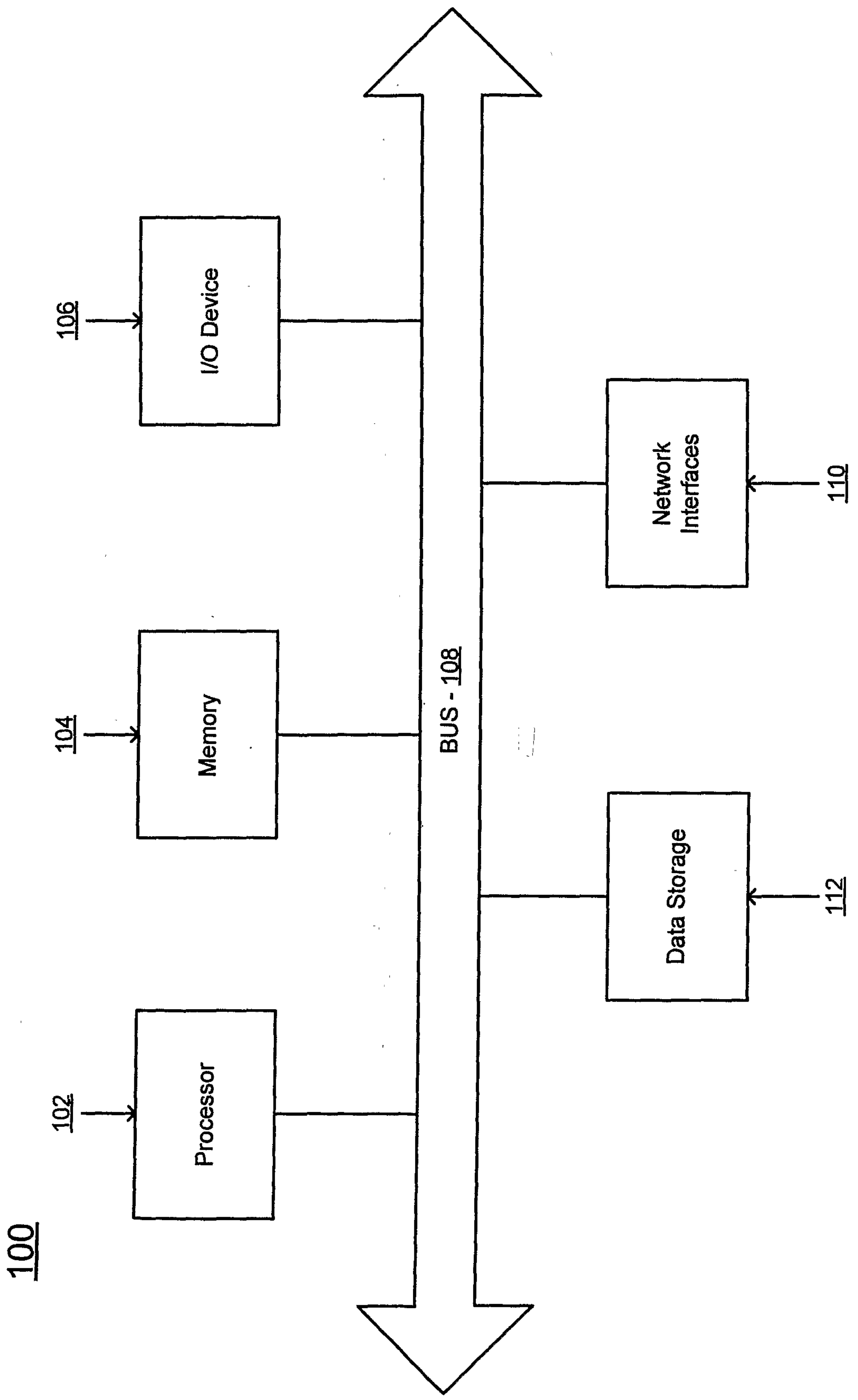
100
102
Processor
104
Memory
106
I/O Device
BUS - 108
Network Interfaces
110
Data Storage
112

FIG. 1

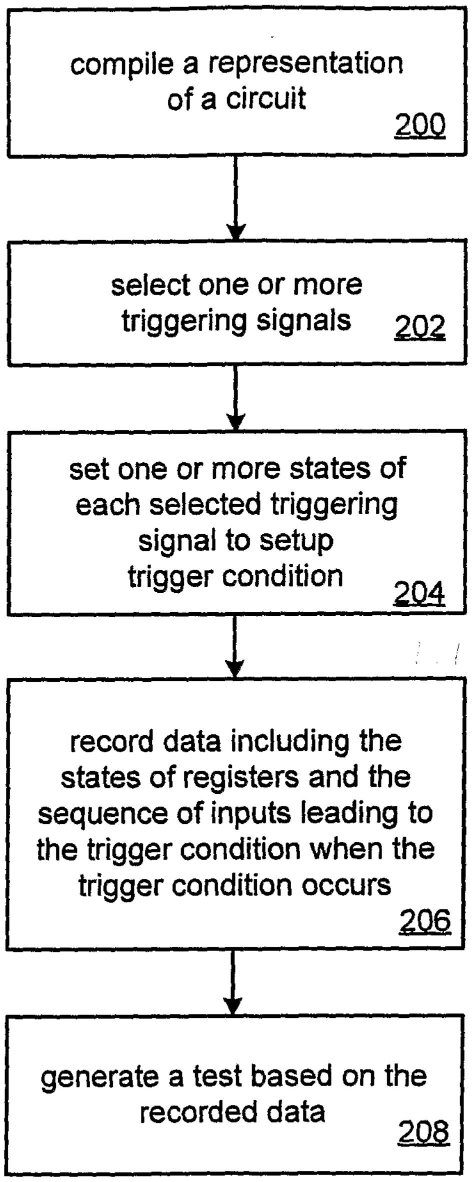
compile a representation of a circuit
200
select one or more triggering signals
202
set one or more states of each selected triggering signal to setup trigger condition
204
record data including the states of registers and the sequence of inputs leading to the trigger condition when the trigger condition occurs
206
generate a test based on the recorded data
208

FIG. 2

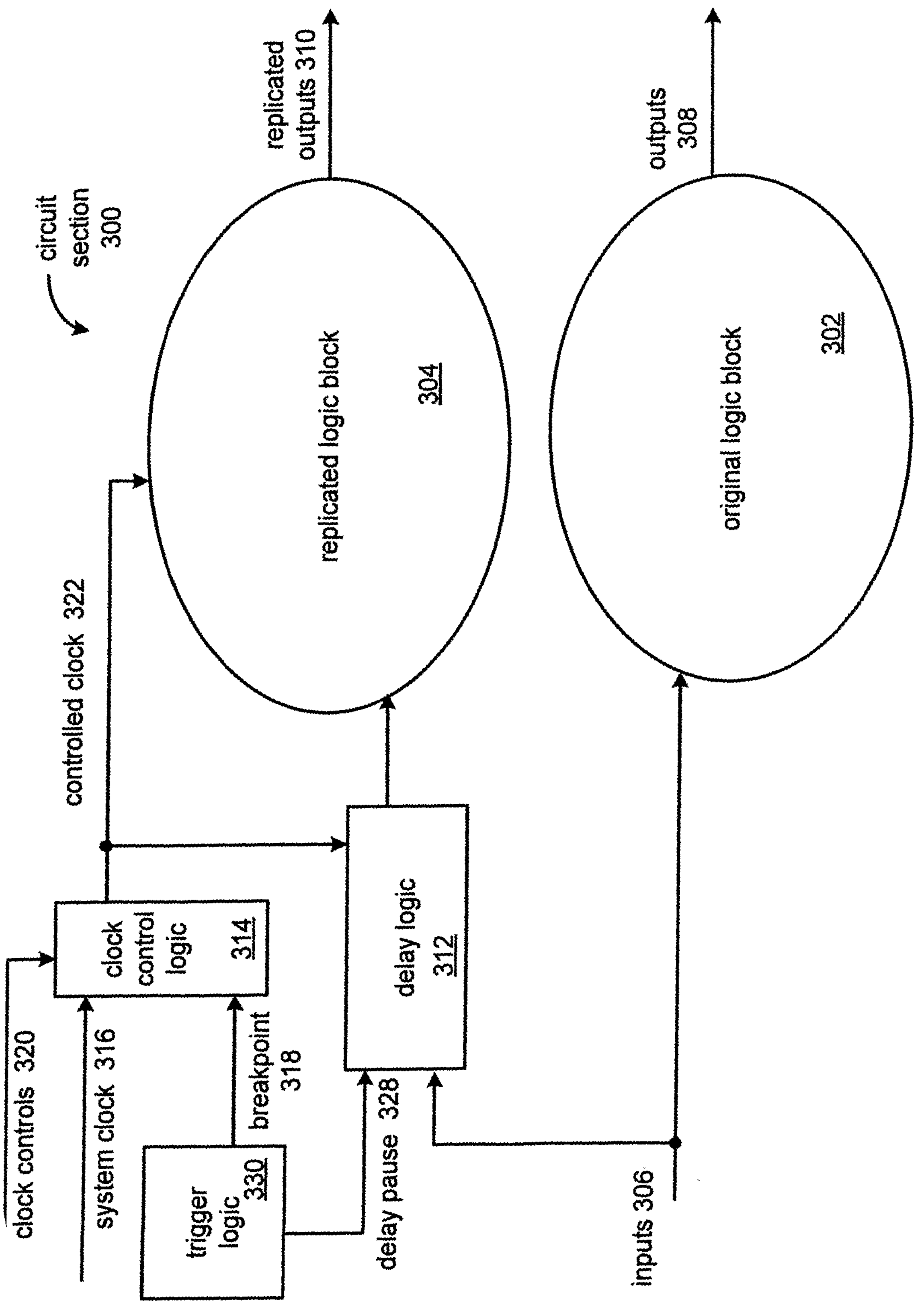
circuit section 300
replicated outputs 310
outputs 308
replicated logic block
304
original logic block
302
controlled clock 322
clock controls 320
system clock 316
clock control logic
314
breakpoint 318
trigger logic
330
delay pause 328
delay logic
312
inputs 306

FIG. 3

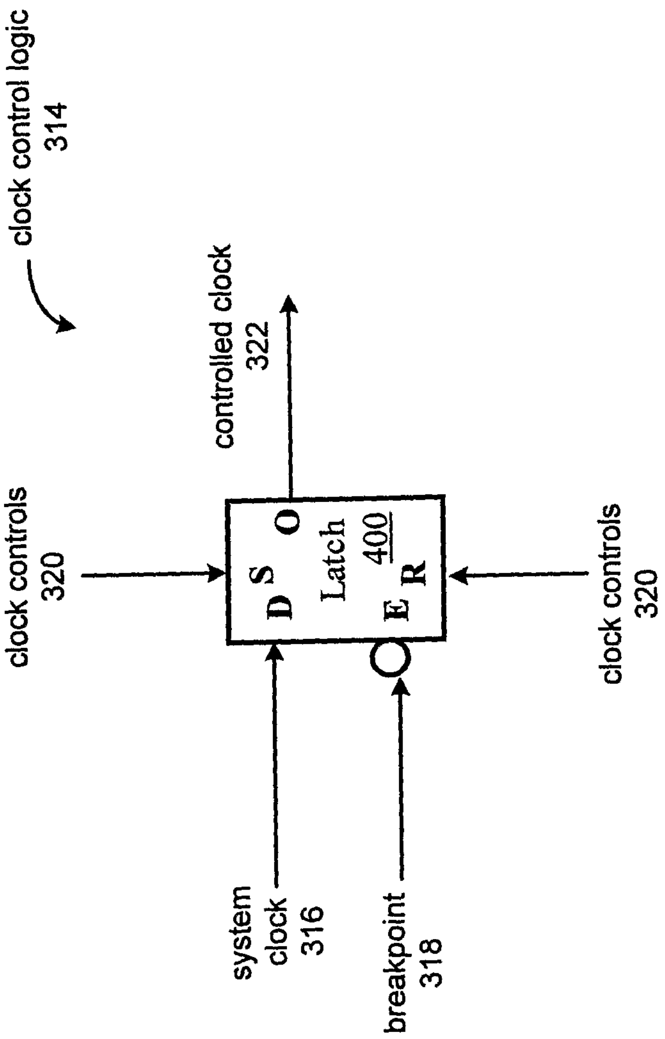
clock control logic
314
controlled clock
322
clock controls
320
D
S
O
Latch
400
E
R
clock controls
320
system
clock
316
breakpoint
318

FIG. 4