BUILT-IN-SELF-TEST USING EMBEDDED MEMORY AND PROCESSOR IN AN APPLICATION SPECIFIC INTEGRATED CIRCUIT

Field of Invention

The present invention relates to an integrated circuit with an embedded processor for execution of test routines; and a test method for the integrated circuit.

Background

Application specific integrated circuits (ASICs) must be tested under a variety of circumstances. For example, during development, an ASIC generally requires thorough testing and debugging to verify or fix the design of the integrated circuit. During production, ASICs generally must be tested to separate good chips from bad chips. During use, an ASIC is often tested to determine whether the integrated circuit is functioning properly in a system.

One method for testing an ASIC during production uses a traditional ASIC tester such as an Agilent 83000 F330 to apply test patterns to the terminals of the ASAIC. The test patterns ideally exercise all of the functional paths of the ASIC and can uncover any defects in the ASIC. Developing the test patterns required for thorough testing can take significant development effort. In particular, developing a test pattern that exercises all functional paths in multiple functional units including embedded memory can be difficult. Additionally, when an ASIC includes an interface to external memory, the test patterns must emulate the external memory, and developing a test pattern that emulates the timing of a high-speed external memory can be time consuming particularly when the ASIC uses a serial interface to reduce pin count.

Even if an exhaustive test pattern is developed, running through a complex test pattern during testing of an ASIC generally requires time, which potentially increases the manufacturing cost of the ASIC. Less exhaustive test patterns can reduce test times, but

This Sheet Has Been Replaced

simpler test patterns may not catch as many defects, resulting in more defective chips being delivered to customers.

[0004] Another limitation of ASIC tests requiring a tester is that such tests are limited to production or development of the ASICs and generally are impractical for testing of an ASIC in commercial product or system. Accordingly, ASICs need at least two types of tests, a test implemented with an external tester and a built-in self-test (BIST) that the ASIC performs in a product. Developing both tests requires duplicated effort and expense.

[0005] BIST tests are generally implemented using special BIST logic that applies deterministic signal patterns in an attempt to exercise the logic paths. Developing BIST logic that performs exhaustive tests is difficult both because there is no guarantee that the paths exercised are the actual functional paths and because creating the test logic often requires specialized design tools. Once produced, such logic is often complicated, increases the ASICs size and cost, and can be overhead that decreases the ASICs performance.

[0006] In view of the difficulties involved in testing, more efficient testing methods and structures are sought for testing ASICs during development, production, and use.

SUMMARY

[0007] In accordance with an aspect of the invention, an ASIC with an embedded processor executes test routines to test the operation of the ASIC. The test routines can perform at-speed functional tests of circuit blocks such as embedded memory, coders and decoders, and interfaces to external devices. The test implementation requires a small amount of IC area associated with the memory storing the test routines. External test equipment can use a simple test pattern that in one embodiment of the invention only involves three pins of the ASIC. Accordingly, the production testing can be performed quickly with simple test equipment and without burdening the ASIC with complex test logic. Additionally, the same or similar test routines in the embedded memory can be used in a self-test when the ASIC is in a product.

[0008] One embodiment of the invention is an integrated circuit including a processing core and a non-volatile memory containing test routines that the processing core executes to

This Sheet Has Been Replaced

test the integrated circuit. An interface block in the integrated circuit can handle signals associated with testing. In particular, the processing core executes the test routines selected according to control signals input via the interface block.

[0009] In one embodiment, the interface block includes first and second terminals. The processing core use a first signal on the first terminal to indicate a test result, i.e., to indicate whether execution of the test routines detected a defect in the integrated circuit. On the second terminal, the processing core activates a second signal to indicate when the first signal indicates the test result. A third terminal can receive a control signal for selection of the test routines from embedded memory for a production test or a system-level test or for selection of execution of firmware downloaded from external memory.

[0010] Functional blocks in the integrated circuit can include data paths that facilitate software testing of the blocks. For example, an input buffer that normally receives data input from an external source can be connected so that the processor can write to the input buffer to test data flow through the input buffer. To reduce the need for test routines that monitor an entire data stream, check code or CRC calculators can be added to specific units to provide a code that is easily checked to detect errors. Further loop-back capabilities can be added to facilitate testing of the data from in and out of network interfaces.

[0011] Another embodiment of the invention is a test method for an integrated circuit. The test method uses an embedded processing core in the integrated circuit to execute test routines stored in an embedded non-volatile memory in the integrated circuit. The test routines can implement complicated tests such as exhaustive testing of an internal memory or other functional blocks of the integrated circuit and output a signal that a tester observes to determine a test result. Generally, a first signal indicates whether the execution of the test routines detected a failure in the integrated circuit. The processing core in executing the test routines can activate a second signal to indicate when a state of the first signal indicates whether the test routines detected a failure. The first signal can be activated before activation of the second signal to acknowledge or signal that the processing core is executing the test routines. One or more additional signals from the integrated circuit can indicate a type or location of a failure that executing the test routines detected.

[0012] A tester during production testing of the integrated circuit thus has a simple test

This Sheet Has Been Replaced

pattern involving only a few pins but obtains a test result from an exhaustive test implemented in the test routines. Additionally, during use of the integrated circuit in a system the system's circuitry can easily initiate a test of the integrated circuit and monitor results.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Fig. 1 is a block diagram of an ASIC with self-test capabilities in accordance with an embodiment of the invention.

[0014] Figs. 2A and 2B are timing diagrams of output signals for a passed self test and a failed self test, respectively.

[0015] Fig. 3 is a flow diagram of an external memory portion of a test process in accordance with the invention.

[0016] Fig. 4 is a block diagram of an ASIC connected to test equipment for production testing in accordance with an embodiment of the invention.

[0017] Fig. 5 is a block diagram of an ASIC in a system capable of system-level testing in accordance with an embodiment of the testing.

[0018] Use of the same reference symbols in different figures indicates similar or identical items.

DETAILED DESCRIPTION

[0019] In accordance with an aspect of the invention, an ASIC with an embedded processor has test routines in an embedded memory. The embedded processor executes the test routines to test the operation of the ASIC. The test routines may be used for ASIC production tests and system-level power-on self-tests. For highly integrated circuits that already contain an embedded ROM, the overhead logic for these self-test functions is minimal.

[0020] Fig. 1 is a block diagram of an ASIC 100 in accordance with an exemplary embodiment of the invention. In Fig. 1, ASIC 100 is a formatter for a printer and in an end

This Sheet Has Been Replaced

product would participate in communications between a printer and a host computer (not shown). This exemplary embodiment is described here to provide a concrete example of one ASIC application, but broad aspects of the invention can be used more widely in any integrated circuit or ASIC that contains an embedded processor having sufficient processing power to execute test routines. Embodiments of the invention are clearly not limited to integrated circuits containing the specific functional units of ASIC 100.

[0021] As illustrated in Fig. 1, ASIC 100 includes a processing core 110, an internal memory 120, a general purpose input/output (GPIO) interface 130 and functional units including a codec 140, an external device interface 150, a print engine communications unit 160, a DMA unit 170, and timing circuits 180. An arbitrated internal bus 190 conducts communication signals among the various blocks of ASIC 100.

[0022] Processing core 110 executes instructions that can be stored in internal memory 120 or external memory (not shown). Any type of processor may be suitable for processor 100, but in an exemplary embodiment of the invention, processing core 110 is an ARM7 processing core, which can be licensed from ARM Ltd.

[0023] Internal memory 120 includes volatile memory such as DRAM 122 and SRAM 124 and non-volatile memory such as ROM 126. ROM 126 can be any type of non-volatile memory such as a mask ROM, an EPROM, or an EEPROM and stores firmware including but not limited to test routines 128. An exemplary set of test routines 128 is described further below and generally includes tests of the operation internal memory 120 and associated memory interface circuits and tests of the other functional units 140, 150, 160, 170, and 180.

[0024] In the exemplary embodiment, interface 150 operates in a normal mode to implement a universal serial bus (USB) interface for communication with a host computer, but interface 150 can also operate to download firmware as described below. Alternatively, GPIO interface 130 can be used as a memory interface to download firmware from an external memory such as a serial EEPROM. Downloaded firmware can replace all or portions of test routines 128 as described further below. Print engine communications 160 implements a communication interface to a printer, and DMA 170 implements direct memory access for transfer of print images. Codec 140 performs coding and decoding operations on print images.

This Sheet Has Been Replaced

[0025] GPIO interface 130 provides the control and output interface for self-test functions of ASIC 100. In particular, GPIO interface 130 employs three signals, ASICTEST, BISTERROR, and BISTDONE. Processing core 110 checks input signal ASICTEST via GPIO interface 130 to determine whether to execute system-level self-test or an ASIC production test. Test routines 128 will generally include slightly differences for system-level tests or production tests. Processing core 110 controls signals BISTERROR, and BISTDONE to indicate test results.

[0026] Figs. 2A and 2B are timing diagrams for signals BISTERROR and BISTDONE and respectively illustrate a passed self-test and a failed self-test in the exemplary embodiment of the invention.

[0027] In Fig. 2A, processing core 110 executes test code 128 (for production or system-level testing) in response to a reset of ASIC 100. Processing core 110 begins by activating single BISTERROR for a short period (e.g., about 100ns) to demonstrate that signal BISTERROR is functional. External test equipment (not shown) detects a failure if processing core 110 fails to activate signal BISTERROR within a period after activation of the reset signal. After deactivating signal BISTERROR, processing core 110 executes the portions of test routines 128 that input control signals such as signal ASICTEST designate. For the example of Fig. 2A, no error is detected, and processing core 110 activates signal BISTDONE upon completing execution of test routines 128. External test equipment during production testing or system circuitry during in system testing identifies the passed self-test from the toggling of signal BISTERROR followed by the activation of signal BISTDONE while signal BISTERROR is in a state indicating no error (e.g., inactive).

[0028] In Fig. 2B, processing core 110 again toggles single BISTERROR to demonstrate signal BISTERROR is functional and then executes the portions of test routines 128 designated by the input control signals. For the example of Fig. 2B, execution of testroutines 128 detects a failure, and processing core 110 reactivates signal BISTERROR and then activates signal BISTDONE to indicate the detection of a test failure. External test equipment or system circuitry identifies the failed self-test from activation of signal BISTDONE while signal BISTERROR is in a state indicating an error (e.g., active). Processing core 100 further can use other output signals from GPIO interface 130 to indicate the type and/or location of the defect or failure.

This Sheet Has Been Replaced

[0029] In the exemplary embodiment of the invention, test routines 128 are part of the boot code that ASIC 100 executes during power up, and test routines 128 include three main portions referred to herein as BIST, EEPROM, and self-test. In the exemplary embodiment of the invention, a control signal input via GPIO interface 130 controls whether processing core 110 executes test routines 128 or attempts to download firmware from external memory via interface 130 or 150. If test routines 128 are executed, processing core 110 starts with the BIST portion of test routines 128.

[0030] The BIST portion is executed before the system is setup to use internal memory or the internal operating system of ASIC 100. The BIST portion tests internal DRAM 122 and SRAM 124, e.g., by performing extensive write and read patterns and validating whether the data read is correct. Processing core 110 checks an error code resulting from the internal memory test and activates signal BISTERROR and then BISTDONE if an error is detected. If ASIC 100 passes the BIST test, test routines 128 enable use of a memory controller in interface 150 to download firmware from an external device.

[0031] In one embodiment of the invention, the BIST test of internal memory is only performed for system-level self-tests and not for production tests. Production testing of ASICs including internal memory such as DRAM often must identify the location of any defects in the memory to enable conventional laser repair operations. To identify the location of a defect in a memory array, the simple error signal timing illustrated in Figs. 2A and 2B can be augmented to provide additional output signals indicating the exact location of any reparable defect in a memory array. However, the resulting increase in the complexity of output of the errors signals increases the complexity of test routines 128 so that a conventional memory array test method may be suited for production testing. For a system-level test, repair is generally not an option, and the memory testing that test routines 128 provides all the required information. As noted above, in the exemplary embodiment, the signal ASICTEST is activated or not to indicate whether a BIST of internal memory is required.

[0032] After ASIC 100 passes the test in the BIST portion of test routines 128, processing core 110 executes the EEPROM portion of test routines 128. Fig. 3 is a flow diagram of a process 300 implemented in the EEPROM portion. Process 300 in an initial decision step 310 checks a control signal input via GPIO interface 130 (e.g., a pin GPIO[13]). If the

This Sheet Has Been Replaced

control signal is not activated, processing core 110 executes of the selftest portion 370 of test routines 128.

[0033] If the control signal is activated, processing core 110 in step 320 attempts to reset an external memory such as an external serial EEPROM and then jumps to executing selftest routines 370 if the reset operation failed (e.g., because no external memory is connected). If the reset operation is successful, processing core 110 in step 340 reads or checks identifying data that should be stored in the external memory and then in a decision step 350 determines whether the external memory contains expected information, e.g., the first word has a value not equal xFFFF. If the external memory contains the expected information, processingcore 110 in step 350 downloads firmware to internal memory 120 and executes that firmware, instead of continuing execution of test routines 128. If the external memory does not provide the expected information, processing core 110 jumps from decision step 350 to execute the self-test portion 370 of test routines 128.

[0034] The ability to load firmware from external memory during the boot process facilitates implementation of tests that are specialized for debugging the design of ASIC 100 or specialized for the particular system using ASIC 100. A co-owned U.S. patent application entitled, "Point-Of-Sale Demonstration of Computer Peripherals", describes use of the firmware download capability to implement non-testing functions (e.g., providing a system demonstration) and is hereby incorporated by reference in its entirety.

[0035] The self–test portion of test routines 128 verifies the operation of the major blocks within ASIC 100. In particular, in the exemplary embodiment, the self-test portion of test routines 128 tests interface 150, DMA block 170, and codec 140. Additional testing of internal memory 120 can also be performed.

[0036] The specific tests of particular blocks depend on the specific functions of the block. For example, for codec 140, processing core 110 can direct data form internal memory 120 to codec 140 for coding or decoding. Processing core 110 then determines whether the output data from codec 140 matches correctly coded or decoded data that is stored in ROM 126.

[0037] To provide tests that replicate actual system operation, test routines can try to emulate the normal data flow in ASIC 100. For example, a normal data flow in the

This Sheet Has Been Replaced

exemplary embodiment starts with input of data to an input FIFO buffer in USB interface 150. The input FIFO buffer can include a normal input path from external circuits and an alternative input path that allows processing core 110 to write data into the input FIFO buffer to start a data flow. Similarly, processing core 110 can read data in an output buffer to check data output.

[0038] A data flow can proceed from the input FIFO buffer to internal memory 120, from internal memory 120 to codec 140 for coding, back to internal memory 120, from internal memory 120 to codec 140 for decoding or to DMA block 170. The proper passage of data through the entire system provides a high degree of verification of the operation of ASIC 100. Additionally, testing a data flow through several functional blocks may avoid the need to separately test each data transfer step because processing core 110 can observe the data at the final stage of the flow to detect errors. To further facilitate error checking, the last functional block in the data flow, e.g., DMA block 170 can include a CRC code calculator, so that processing core 110 only needs to check a CRC code instead of the entire output data stream.

[0039] Although the functional blocks of ASIC 100 can be conventional in implementation, specific features can be built into various functional block to facilitate test operations executed embedded processor 110. For example, the DMA block 170 can include circuitry that performs the CRC calculations to allow error detection without requiring processing core 110 to monitor the entire data stream. Input blocks such as USB interface 150 can provide paths that permit processing core 110 to write input values into incoming FIFO buffers to simulate data input when testing a data flow. Implementations of "loopback test" capabilities in the functional block can facilitate tests executed by processing core 110. Such tests would be particularly useful for testing on-chip network interfaces.

[0040] Processing core 110 can also test data flow through the blocks of ASIC 100 at specific clock speeds used in ASIC 100 and determine whether the blocks are meeting the required timing. For this testing, the clock and reset pins of ASIC 100 can driven during production or system-level testing in the manner required for normal operation of ASIC 100. Conventional test equipment 400 as illustrated in Fig. 4 can easily implement the timing signal CLK and control signals ASICTEST to test ASIC at full speed or at an elevated speed, for example, to prove timing margins during production testing.

This Sheet Has Been Replaced

[0041] In the above test process, processing core 110 can provide a failure code CODE in addition to asserting signal BISTERROR. Failure code CODE would indicate the nature or the location of the failure found during testing. For example, the value a 2-bit failure code can designate that whether the detected failure is in internal memory 120, codec 140, DMA block 170, or interface block 150.

[0042] In a product, ASIC 100 is connected to other system components 500 such as illustrated in Fig. 5. In the embodiment of Fig. 5, for example, external interface 150 is connected to a host connector 510 for connection to a host computer or external memory, and print engine communications 160 connects to a printer connector 520 of a printer. Clock and system control circuit 540 and an optional external memory 530 connect to timing circuits 180 and GPIO interface 130. For system-level testing, system control 540 controls signal ASICTEST to select a system-level test and monitors signals BISTDONE and BISTERROR to determine whether ASIC 100 is functioning properly. In a system such as illustrated in Fig. 5, when executing test routines, which may be from internal memory 120 or downloaded to ASIC 100 from memory 530, processing core 110 can test ASIC 100 and any of the other system components 500.

[0043] Although the invention has been described with reference to particular embodiments, the description is only an example of the invention's application and should not be taken as a limitation. For example, although the above-described embodiment is a formatter for a printer, embodiments of the invention can be employed in other types of integrated circuits. Various other adaptations and combinations of features of the embodiments disclosed are within the scope of the invention as defined by the following claims.

This Sheet Has Been Replaced

We claim:

1. An integrated circuit comprising:

a processing core;

an internal memory containing test routines that the processing core executes to test the integrated circuit; and

an interface coupled to the processing core to permit activation of a first output signal indicating a test result from executing the test routines.

2. The integrated circuit of claim 1, wherein the processing core executes the test routines in response to a reset signal.

3. The integrated circuit of claim 2, wherein the interface block comprises:

a first terminal on which the processing core activates the first output signal to indicate the test result; and

a second terminal on which the processing core activates a second output signal to indicate when the first output signal indicates the test result.

4. The integrated circuit of claim 3, wherein the processor toggles the first output signal to verify that the first output signal is functional.

5. The integrated circuit of claim 1, wherein the processing core executes the test routines from the internal memory during a production test of the integrated circuit.

6. The integrated circuit of claim 1, wherein the test routines include tests of the internal memory.

7. The integrated circuit of claim 1, wherein the internal memory contains a first set of test routines for execution during a production test of the integrated circuit and a second set of test routines for execution during an in-product test of the integrated circuit.

8. The integrated circuit of claim 7, wherein a control signal input to the integrated

This Sheet Has Been Replaced

circuit controls whether the first or second set of routines are executed.

9. A test method for an integrated circuit, comprising:

using a processing core in the integrated circuit to execute test routines stored in the integrated circuit; and

observing a first signal output from the integrated circuit as a result of the processing core executing the test routines, the first signal indicating whether the execution of the test routines detected a failure in the integrated circuit.

10. The test method of claim 9, further comprising observing a second signal output from the integrated circuit, wherein processing core in executing the test routines activates the second signal to indicate when a state of the first signal indicates whether the execution of the test routines detected a test failure.

11. The test method of claim 10, further comprising activating the first signal before activation of the second signal to verify that the first signal is functional.

12. The test method of claim 11, activating an input signal to the integrated circuit to causes the processing core to execute the test routines, wherein activating the first signal before activation of the second signal is in response to the activating of the input signal.

13. The test method of claim 9, observing one or more additional signals from the integrated circuit, wherein the processing core controls the additional signals to indicate a type of failure that executing the test routines detected.

14. The test method of claim 9, wherein the test method is performed during production of the integrated circuit.

15. The test method of claim 14, further comprising controlling an input signal to the integrated circuit to select which of the test routines stored in the integrated circuit are executed.

This Sheet Has Been Replaced

16. The test method of claim 9, further comprising:

applying a control signal to the integrated circuit; and

selecting the test routines according to the control signal.

17. The test method of claim 16, wherein:

~~when the control signal has a first state, the test routines selected implement a~~ production test of the integrated circuit; and

when the control signal has a second state, the test routines selected implement a system-level test of the integrated circuit.

This Sheet Has Been Replaced