**Title:** OVERRIDABLE ELEMENTS IN RECONFIGURABLE LOGIC DEVICES

**Abstract:** A method and system for configuring an area of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool. The method comprises the steps of identifying all inverting and potentially-inverting logic units in the area and grouping each inverting logic unit and potentially-inverting logic unit into at least one chain comprising inverting logic units and/or appropriately-configured potentially-inverting logic units, which chains form non-inverting paths through the area. The method also comprises the step of overriding the reconfigurable area to implement the chains and the appropriate configurations of the potentially-inverting logic units during testing.

Published:

— with international search report (Art. 21(3))
The present invention relates to the field of reconfigurable logic devices. More specifically, the present invention relates to a system and method for safely testing a reconfigurable array using a standard Automatic Test Pattern Generation (ATPG) tool.

ATPG tools allow manufacturers of logic circuits to distinguish between correct circuit behaviour and incorrect circuit behaviour caused by defects. In order to do this, ATPG tools analyse a model of the logic circuit to create pseudo-random sequences of data. These sequences, when fed to the logic circuits in question, create outputs which can be compared to expected output values. Where these are differences between the expected and actual output values, such differences are associated with defects in the digital circuit.

When ATPG techniques are used to create sequences for reconfigurable logic circuits, a problem with this method can arise when inverting feedback loops are created by the random sequence of configuration data. Such inverting feedback loops are created when part of the logic device is configured in such a way that a signal can loop back on itself. If there is an even number of inverting elements in that loop, the state of all of the elements will find a steady state after the signal has completed the loop. On the other hand, if the loop contains an odd number of inverting elements, the loop will continue to oscillate infinitely.

This oscillation can draw a very large amount of current, which current, which, at best, is costly in terms of power consumption, and at worst, can damage, or even destroy, a circuit.

In order to remedy this, some prior reconfigurable devices have found new ways of generating test patterns. These solutions however are specifically designed for particular reconfigurable devices having particular architectures. Accordingly, the test patterns generated by these solutions do not always conform to the ATPG used by the other components on a System on Chip (SoC).

These devices are therefore not only costly in terms of computing resources, but also add significant complexity to the process for testing the larger Integrated Circuit (IC) of which the reconfigurable device forms a part of.
Accordingly, there is a need for an improved system and method for eliminating the risk of inverting feedback loops in reconfigurable devices during testing using standard Automatic Test Pattern Generators (ATPG).

In order to solve the problems associated with the prior art, the present invention provides a method of configuring an area of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the method comprises the steps of:

- identifying all inverting and potentially-inverting logic units in the area;
- grouping each inverting logic unit and potentially-inverting logic unit into at least one chain comprising inverting logic units and/or appropriately-configured potentially-inverting logic units, which chains form non-inverting paths through the area;
- overriding the reconfigurable area to implement the chains and the appropriate configurations of the potentially-inverting logic units during testing.

Preferably, the at least one chain comprises a plurality of potentially inverting logic units and/or inverting logic units, and the at least one chain further comprises at least one routing unit for interconnecting the plurality of potentially inverting units and/or inverting logic units, and
- the overriding step further comprises the step of overriding the at least one routing unit to implement the chains.

Preferably, the grouping and overriding steps are physically performed using the reconfigurable logic of the reconfigurable array.

Preferably, the above method further comprising the steps of:

- providing a register to the output of at least one inverting logic unit; and
- providing bypass means for selectively bypassing the register, such that the inverting logic unit can be used as a potentially-inverting logic unit by controlling the bypass means.

Preferably, the grouping and overriding steps are implemented in software by altering of the data output from the Automatic Test Pattern Generation (ATPG) tool.

The present invention also provides a method of configuring an array of a reconfigurable array of logic units in order to avoid inverting feedback loops during
testing using an Automatic Test Pattern Generation (ATPG) tool, the method comprises the steps of:

partitioning the array into a plurality of areas; and
executing the above method for each area.

The present invention also provides a system for configuring an area of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the system comprises:

identifying means arranged to identify all inverting and potentially-inverting logic units in the area;

grouping means, arranged to group each inverting logic unit and potentially-inverting logic unit into at least one chain comprising inverting logic units and/or appropriately-configured potentially-inverting logic units, which chains form non-inverting paths through the area;

overriding means arranged to override the reconfigurable area to implement the chains and the appropriate configurations of the potentially-inverting logic units during testing.

Preferably, the at least one chain comprises a plurality of potentially inverting logic units and/or inverting logic units, and the at least one chain further comprises at least one routing unit for interconnecting the plurality of potentially inverting units and/or inverting logic units, and

the overriding means further comprises routing unit overriding means arranged to override the at least one routing unit to implement the chains.

Preferably, the grouping and overriding means are implemented in hardware by using the reconfigurable logic of the reconfigurable array.

Preferably, the register providing means arranged to provide a register to the output of at least one inverting logic unit; and

bypass providing means arranged to provide bypass means for selectively bypassing the register, such that the inverting logic unit can be used as a potentially-inverting logic unit by controlling the bypass means.
Preferably, the grouping and overriding means are implemented in software by alteration of the data output from the Automatic Test Pattern Generation (ATPG) tool.

The present invention also provides a system for configuring a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the system comprising:

- partitioning means arranged to partition the array into a plurality of areas;

and

- a system in accordance with the above arranged to configure each of the area partitioned by the partitioning means.

The present invention also provides a reconfigurable logic device configured in accordance with the above method.

As will be appreciated, the present invention provides several advantages over the prior art. For example, because the present invention has the potential to modify the circuitry of the reconfigurable device, it is possible to use standard Automatic Test Pattern Generators (ATPG) to test the reconfigurable device.

Specific embodiments of the present invention will now be described with reference to the attached drawings, in which:

- Figure 1 represents a reconfigurable device/fabric which can be used in conjunction with the present invention; the reconfigurable device/fabric is made up of a plurality of user programmable logic tiles;

- Figure 2 represents a closer view of the reconfigurable device of Figure 1, including Arithmetic Logic Units and switch boxes;

- Figure 3 is functional block diagram of a portion of a reconfigurable device susceptible to the presence of inverting feedback loops during testing;

- Figure 4 is a flow chart representing the steps in a method of identifying override scenarios in accordance with one embodiment of the present invention;
Figure 5 is a functional block diagram of the reconfigurable device of Figure 3, configured in accordance with a first scenario determined using the method of Figure 4;

Figure 6 is a functional block diagram of the reconfigurable device of Figure 3, configured in accordance with a second scenario determined using the method of Figure 5;

Figure 7 is functional block diagram of another portion of a reconfigurable device susceptible to the presence of inverting feedback loops during testing;

Figure 8 is a functional block diagram of the reconfigurable device of Figure 7, configured in accordance with a first scenario determined using the method of Figure 4;

Figure 9 is a functional block diagram of the reconfigurable device of Figure 7, configured in accordance with a second scenario determined using the method of Figure 4;

Figure 10 is a functional representation of a single Logic Unit of the reconfigurable device connected to a configuration memory;

Figure 11 is a functional representation override circuitry added to the reconfigurable device of Figure 3 in accordance with the present invention;

Figure 12 is a functional representation override circuitry added to the reconfigurable device of Figure 4 in accordance with the present invention; and

Figure 13 is a functional representation of a single Logic Unit in accordance with a preferable embodiment of the present invention.

Reconfigurable devices/fabrics, such as D-Fabrix (disclosed in, for example, US6353841, US6252792, US2002/0157066) are commonly made up of a plurality of interconnected user programmable logic blocks or tiles, the fundamental building blocks of the system. This arrangement facilitates the use of an easily scalable configuration mechanism of equally regular structure.

Each user programmable tile is connected to a programmable routing network which can implement arbitrary connectivity between the tiles. Because each tile is connected to the routing network in the same way, the resulting device has a high degree of homogeneity. That is to say that, the way in which a specific subset of tiles of an array can be used and interconnected will be the
same, regardless of where on the array that subset is located. Such a homogeneous array can be found disclosed in, for example, US6252792.

Figure 1 shows a diagram representing a programmable fabric comprising a plurality of Logic Units (LUs) interconnected by way of a plurality of switching sections. As shown in Figure 1, each tile 20 is divided into four areas. A two-by-two group of LUs and switching sections forms a tile 20, which is the basic building block of the fabric, and is shown bounded by a thick line in Figures 1 and 2. Two of the areas, which are diagonally opposed in the tile 20, provide locations for a pair of LUs. The other two circuit areas, which are also diagonally opposed in the tile, provide the locations for a pair of switchboxes.

Each LU can perform standard arithmetic operations (such as ADD, SUBTRACT) as well as standard logic operations (such as AND, NAND, OR, NOR) on a set number of bits. As will be appreciated by the skilled reader, some of the above functions will logically invert a signal, whilst other will not.

Figure 2 shows a closer view of the fabric. Each tile 20 contains two LUs and two switching sections. Each switching section comprises a plurality of switches which are arranged to selectively connect a horizontal bus to a vertical bus at their intersection point. The horizontal and vertical buses can be any number of bits wide. Some switches, which are shown as black squares in Figure 2, are used for locally connecting the LUs to the switching sections. Other switches, which are shown as striped squares in Figure 2, are used for longer distance connections (e.g. between switch sections).

As can be seen from Figure 1 and Figure 2, the fabric has a high degree of homogeneity in that a particular tile can be used (i.e. configured or interconnected) in the exact same way as every other tile in the array.

Figure 3 is a functional diagram of a portion of a reconfigurable device which can be used with the present invention. As will be appreciated by the skilled reader, Figure 3 is a simplified diagram representing a portion of a reconfigurable device which may be part of an inverting feedback loop, and is merely used to illustrate the operation of the present invention. The structure and functionality of
the circuits shown in the figures are not limiting to the functionality of the present invention.

The portion of the reconfigurable device of Figure 3 comprises logic units LU1 - LU12 and routing units R1 - R12. All of these elements are interconnected as shown by the arrows. The skilled reader will appreciate that other interconnections may be possible. In Figure 3, the centre of each logic unit LU comprises pictorial information relating to the logical inverting effect of that logic unit LU (i.e. whether a signal input to the logic unit will be logically inverter at that output of that logic unit LU). The alignment of this pictorial information has no bearing on the function of a particular logic unit.

For example, the centre of LU8 comprises an inverter, which indicates that, apart from other possible operations on an input signal, LU8 acts as an inverter, regardless of the function LU8 is configured to perform. Accordingly, a signal input to logic unit LU8 from routing unit R8, and then subsequently output to routing unit R10 will necessarily be inverted. Thus, a route through the portion of the reconfigurable device consisting of the elements R1, R4, R6, R8, LU8, R10 and R12, and including the feedback loop, will result in an inverting loop.

The pictorial information at the centre of LU3 shows that LU3 is potentially inverting. That is to say that, at least one function of LU3 will invert its input signal, and at least one other function of LU3 will not invert its input signal. Accordingly, if configured one way, the LU3 will have an inverting effect on its input signal and, if configured another way, it will not have an inverting effect on its input signal.

Thus, a route through the portion of the reconfigurable device consisting of the elements LU2, R4, LU3, R5, R8, R9, R12 and including the feedback loop, will result in an inverting feedback loop if one or the other (but not both) of LU2 and LU3 are configured to invert. If both of LU2 and LU3 are configured to invert, these inversion will cancel each other out and the chain LU2, R4, LU3, R5, R8, R9, R12 will be non-inverting. Moreover, a route through the portion of the reconfigurable device consisting of the elements R2, LU4, R6, LU8, R10, R12 and including the feedback loop, will result in an inverting feedback loop if LU4 is configured to not invert, and will result in a non-inverting feedback loop is LU4 is configured to invert, as the inversion of the inversion of LU4 and the inversion of LU8 will cancel each other out.
Finally, the absence of any pictorial information at the centre of LU1 shows that logic unit LU1 is non-inverting, regardless of the function LU1 is configured to perform. Thus, a route through the portion of the reconfigurable device consisting of the elements LU1, R3, R5, LU7, R9, LU10 and R12 and including the feedback loop, will result in a non-inverting loop, regardless of which functions the logic units in that route are configured to perform.

The circuit of Figure 3 comprises a plurality of inverting, non-inverting and potentially inverting logic units. Accordingly, if configured in specific ways, the logic units and routing units of the portion of the reconfigurable fabric of Figure 3 could implement a plurality of inverting feedback loops. As mentioned above, the oscillation cause by inverting feedback loops can draw a very large amount of current, which current, which, at best, is costly in terms of power consumption, and at worst, can damage, or even destroy, a circuit.

During normal operation of the circuit, it is relatively easy for a skilled person to configure the routing and logic units of this portion of the reconfigurable device to avoid inverting feedback loops. When the same portion of the circuit is tested using an Automatic Test Pattern Generation (ATPG) tool however, the routing units and logic units will be configured randomly. Thus, the routing units RU can be configured to implement any chain of logic units, and each of the potentially inverting logic units can either be configured to invert or not. This raises the problem of accidentally creating inverting feedback loops during testing.

In order to address this problem, the present invention forces specific chains of the logic units LUs and routing units Rs to perform in specific functions during testing. In order to allow each logic unit LU and each routing unit R in the chain to be tested, different scenarios are created, each of which comprises one or more override chains, thereby allowing each routing unit R and each logic unit LU to be tested whilst performing each of its functions.

Figure 4 is a flow chart representing the steps in a method of identifying override scenarios in accordance with one embodiment of the present invention. This method can be performed on a computer or, alternatively, by a local processor on, for example, a system-on-chip (SOC) device. The first step in the method (i.e. step 101) is to identify all of the logic units in the portion of the reconfigurable device and identifying which are inverting (INV), potentially inverting (PINV) and
non-inverting (NINV). This embodiment of the present invention assumes that all routing units \( R \) in the portion of the reconfigurable device have the same function and are homogenously spread throughout the portion. For example, each routing unit \( R \) in the portion of Figure 3 will route a single input to a single output, and all routing units \( R \) are equally distributed throughout the device. In other embodiments of the present invention, when the location and the function of the logic units LU are being determined in step 101 of the method, the position and function of the routing units is also determined.

Once step 101 is complete, the scenario counter is set to 1 in step 102. The use of the scenario counter will be further explained below.

Then, in step 103, a determination is made as to whether it is possible to determine a chain comprising inverting LUs and non-tested potentially-inverting LUs configured to invert, in which 1) the sum of inverting LUs and potentially-inverting LUs configured to invert in the chain is an even number, and 2) the chain comprises all inverting LUs in the portion. As should be appreciated by the skilled reader, non-tested (or untested) potentially-inverting LUs configured to invert are potentially-inverting LUs which have not previously formed part of a chain whilst being configured to invert. If the answer to the above is "yes", then step 104 is performed. If the answer to the above is "no", then step 110 is performed.

In step 104, the method determines the longest chain of inverting LUs and potentially inverting LUs set to invert, where the sum of the inverting LUs and potentially inverting LUs is an even number. Then, in step 105, the routing cost of each routing unit \( R \) in the chain found in step 104 is increased by a certain amount. When using this embodiment of the present invention, the determination of a chain through the portion of the reconfigurable device is done by minimising the total routing cost of any one chain. By increasing the routing cost of each routing unit \( R \) in a chain, it is possible to avoid using the same routing units in two different chains, or, at the very least, it is possible to avoid using the same routing unit configured in the same way in two different chains, thereby allowing all routing units to be tested in all possible configurations.

Then, in step 106, all potentially-inverting LUs in the chain are set to invert (i.e. \( \text{PINV}(i) \)), and in step 107 all potentially-inverting LUs not part of the chain are set to not invert (i.e. \( \text{PINV}(n) \)). At this point, the arrangement which has been
created is a non-inverting scenario in which the LUs and Rs in the portion of the reconfigurable device can be tested. Each potentially-inverting LU in the scenario can only be tested in one of its two possible states (i.e. either PINV(i) or PINV(n)), depending on how these elements have been configured in the scenario. This scenario (i.e. the location and value of configuration bits necessary to implement this scenario) is saved for future reference.

In step 108, the scenario number is increased by 1 unit. Then, in step 109, if any remaining potentially-inverting LUs set to invert in the last scenario remain untested, step 110 is performed. If not, the step 103 is performed again. In step 110, if any potentially-inverting LUs set to not invert in the last scenario remain untested, step 111 is performed. If on the other hand, no potentially-inverting LUs set to not invert remain untested, the method is brought to an end.

In step 111, the shortest chain of inverting LUs and potentially-inverting LUs set to invert is found, where the sum of the elements in the chain is even, and where the number of inverting LUs can be zero. Also, it should be noted that the potentially-inverting LUs in this chain need not be untested. Then, in step 112, any potentially-inverting LUs which are part of the chain are set to invert and any potentially-inverting LUs which are not part of the chain are set to not invert.

At this point, the arrangement which has been created is a non-inverting scenario in which the LUs and Rs in the portion of the reconfigurable device can be tested, and wherein each potentially-inverting LU in the scenario is tested in the other of its two possible states (i.e. either PINV(i) or PINV(n)). This scenario (i.e. the location and value of configuration bits necessary to implement this scenario) is also saved for future reference.

In step 112, the routing cost of each routing unit R in the chain is increased by a certain amount. Then, in step 113, the scenario number is increased by 1 unit and the method returns to step 110.

Now with reference to Figures 3, 5 and 6, an example of how the chain determination method of Figure 4 works will now be described. In step 101, all of the LUs in Figure 3 are determined. Accordingly, LU1, LU5, LU6, LU7, LU10 and LU12 are determined to be non-inverting LUs; LU2, LU3, LU4 and LU9 are determined to be potentially-inverting LUs and LU8 and LU11 are determined to be inverting LUs.
Then in step 103 it is determined that it is possible to determine a chain comprising inverting LUs and non-tested potentially-inverting LUs configured to invert, in which 1) the sum of inverting LUs and potentially-inverting LUs configured to invert in the chain is an even number, and 2) the chain comprises all inverting LUs in the portion. In step 104, the following chain is determined: LU2-R1-LU3-R4-LU4-R6-LU8-R1 0-LU1 0-R9-LU9-R1 1-LU1 1, as shown in Figure 5.

Then, in step 105, the routing costs of R1, R4, R6, R10, R9 and R11 are increased. As is also shown in Figure 5, LU2, LU3, LU4 and LU9 are set to invert, in accordance with step 106. Because the chain includes all potentially-inverting LUs in the portion of the reconfigurable device, no remaining potentially-inverting LUs will be set to non-invert in step 107. In step 108, the scenario number is increased by 1.

As shown in Figure 5, and mentioned above, no potentially-inverting LUs set to invert remain. Accordingly, step 109 is answered in the affirmative. Then, because all of the potentially-inverting LUs in the first chain are set to invert, the question as to whether the number of untested potentially-inverting LUs set to not invert is zero is answered in the negative. In step 111, the shortest chain of inverting LUs and potentially-inverting LUs set to invert is found to be LU8-R8-R9-LU1 1, as shown in Figure 6. It should be noted that the number of potentially-inverting LUs is in the chain is zero. It should also be noted that R8 and R9 are configured differently in the second chain as they are in the first chain.

Then, in step 112, the potentially inverting LUs which do not form part of the chain are set to not invert, as shown in Figure 6. The routing costs of the routing units R used in the chain is increased and the scenario number is increased by 1.

In step 110, it is found that no potentially inverting LUs set to not invert remain to be tested. Accordingly, the method is ended and the results (i.e. scenario 1 of Figure 5 and scenario 2 of Figure 6) are used to either alter the portion of the reconfigurable device of Figure 3 or alter the random data generated by the Automatic Test Pattern Generation (ATPG) tool, as described below.

Before describing how the results of the method can be used, another example of how the method of identifying non-inverting scenarios in accordance with one example of the present invention will now be described with reference to Figures 7 to 9.
Figure 7 shows a portion of a reconfigurable device which comprises a plurality of LUs. As will be appreciated, the portions shown in Figures 3 and 7 could represent different portions of the same reconfigurable device, in which case the scenarios found for each portion could be used concurrently, as described below. It should also be noted that the present invention is intended to be used with portions of reconfigurable devices comprising several thousand LUs, and the simplicity of Figure 3 and 7 are for presentation purposes only.

In step 101, all of the LUs in Figure 7 are determined. Accordingly, LU1, LU5, LU6, LU7 and LU12 are determined to be non-inverting LUs; LU2, LU3, LU4 and LU9 are determined to be potentially-inverting LUs and LU8, LU10 and LU11 are determined to be inverting LUs.

Then in step 103 it is determined that it is possible to determine a chain comprising inverting LUs and non-tested potentially-inverting LUs configured to invert, in which 1) the sum of inverting LUs and potentially-inverting LUs configured to invert in the chain is an even number, and 2) the chain comprises all inverting LUs in the portion. In step 104, the following chain is determined: LU2-R4-LU4-R6-LU8-R1 0-LU1 0-R9-LU9-R1 1-LU1 1, as shown in Figure 8.

Then, in step 105, the routing costs of R4, R6, R10, R9 and R11 are increased. As is also shown in Figure 8, LU2, LU4 and LU9 are set to invert, in accordance with step 106. Because the chain does not includes all potentially-inverting LUs in the portion of the reconfigurable device, remaining potentially-inverting LU LU3 will be set to non-invert in step 107. In step 108, the scenario number is increased by 1.

As shown in Figure 8, and mentioned above, one potentially-inverting LU set to invert remains. Accordingly, step 109 is answered in the negative and step 103 is performed.

Then in step 103 it is determined that it is possible to determine a chain comprising inverting LUs and non-tested potentially-inverting LUs configured to invert, in which 1) the sum of inverting LUs and potentially-inverting LUs configured to invert in the chain is an even number, and 2) the chain comprises all inverting LUs in the portion. In step 104, the following chain is determined: LU3-R5-R8-LU8-R8-LU1 0-R1 2-LU1 1, as shown in Figure 9.
Then, in step 105, the routing costs of R5, R8, R10 and R12 are increased. As is also shown in Figure 9, LU3 is set to invert, in accordance with step 106. Because the chain does not include all potentially-inverting LUs in the portion of the reconfigurable device, remaining potentially-inverting LUs LU2, LU4, and LU9 will be set to non-invert in step 107. In step 108, the scenario number is increased by 1.

Then, because each potentially-inverting LU set invert and each partially-inverting LU set to not-invert will have been tested, the steps 109 and 110 will be answered in the affirmative and the method will be ended.

Now, with reference to Figures 10 to 12, the details of how a portion of a reconfigurable device can be modified based on the chains and scenarios described above will now be described.

Preferably, the information determined using the method of Figure 4 is used by a modified ATPG tool in accordance with the present invention in order to create at least one override signal OS. Using this override signal, it is possible to add custom circuitry to the LUs in the portion of the reconfigurable device to allow the scenarios to be implemented when the modified ATPG tool activates the at least one override signal.

As shown in Figure 10, the inverting aspect of a potentially-inverting LU can usually be controlled by a small number of configuration bits, which configuration bits will be subject to being set by the random data of the ATPG tool.

With reference to the example of Figure 3 and corresponding scenario of Figure 5 outlined above, Figure 11 shows that each configuration bit cLU2, cLU3, cLU4, and cLU9, in respect of LU2, LU3, LU4 and LU9, respectively, can be overridden by using an override signal OS, a test signal TS, and a small amount of dedicated circuitry, in this case, a multiplexer. The configurations of R1, R4, R6, R10, R9 and R11 are also set by OS, in order to set the connections as shown in Figure 5.

Accordingly, when the device is not in test mode (i.e. TS is HIGH), the value of the configuration bits cLU2, cLU3, cLU4 and cLU9 will determine whether LU2, LU3, LU4 and LU9 are set to invert or not.

If however, the device is in test mode (i.e. TS is set HIGH), whether LU2, LU3, LU4 and LU9 will invert or not is determined by the value of OS. If OS if HIGH,
each of LU2, LU3, LU4 and LU9 will be set to not invert, and OS if LOW, each of
LU2, LU3, LU4 and LU9 will be set to invert. Moreover, it is possible to route the
same OS to each of the routing units R used in the chains found in accordance with
the method of Figure 4 and use that signal to force them to route such that they
implement the chains.

With reference with the first example described above, and the added
circuitry of Figure 11, it can be seen that, when in test mode, the chain in the
scenario of Figure 5 will be implemented when the override signal is LOW, and the
chain in the scenario of Figure 6 will be implemented when the override signal is
HIGH. Similarly and now with reference to the added circuitry of Figure 12, as well
as the example of Figures 7 to 9, it can be seen that, when in test mode, the chain
in the scenario of Figure 8 will be implemented when the override signal is LOW,
and the chain in the scenario of Figure 9 will be implemented when the override
signal is HIGH.

The routing units R which form part of the chains in the scenarios will be
arranged similarly, such that there operation can be overridden by the test and
override signals.

Accordingly, by determining non-inverting chains in various areas of the
reconfigurable device, each chain of an area being associated with an overall
scenario associated with the device, and using that information to add dedicated
hardware which employs a global override signal to implement the different
scenarios, it is possible to avoid inverting feedback loops during testing.

As will be appreciated, the examples described above could be different
areas of the same reconfigurable device/fabric. If this was the case, the present
invention would only need a single override signal to implement the chains of the
first scenario (i.e. Figure 5 in the first area and Figure 8 in the second area).

Moreover, it will also be appreciated that the present invention could be
implemented in software, where instead of adding dedicated hardware to the logic
units LU, it would be possible to replace specific portions of the ATPG data with
replacement configuration data which would force the areas to implement the non-
inverting chains.

Figure 13 shows an LU which has been modified by the addition of a
register. The effect of this modification is that it can make an inverting LU act as a
potentially-inverting LU. As will be appreciated by the skilled reader, this modification of this sort can be used in accordance with the present invention.
1. A method of configuring an area of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the method comprising the steps of:
   identifying all inverting and potentially-inverting logic units in the area;
   grouping each inverting logic unit and potentially-inverting logic unit into at least one chain comprising inverting logic units and/or appropriately-configured potentially-inverting logic units, which chains form non-inverting paths through the area;
   overriding the reconfigurable area to implement the chains and the appropriate configurations of the potentially-inverting logic units during testing.

2. The method of claim 1, wherein:
   the at least one chain comprises a plurality of potentially inverting logic units and/or inverting logic units, and the at least one chain further comprises at least one routing unit for interconnecting the plurality of potentially inverting units and/or inverting logic units, and
   the overriding step further comprises the step of overriding the at least one routing unit to implement the chains.

3. The method of any of claims 1 or 2, wherein the grouping and overriding steps are physically performed using the reconfigurable logic of the reconfigurable array.

4. The method of any of the preceding claims, further comprising the steps of:
   providing a register to the output of at least one inverting logic unit; and
   providing bypass means for selectively bypassing the register, such that the inverting logic unit can be used as a potentially-inverting logic unit by controlling the bypass means.
5. The method of any of claims 1 or 2, wherein the grouping and overriding steps are implemented in software by altering of the data output from the Automatic Test Pattern Generation (ATPG) tool.

6. A method of configuring an array of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the method comprising the steps of:
   partitioning the array into a plurality of areas; and
   executing the method of any of claims 1 to 5 for each area.

7. A system for configuring an area of a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the system comprising:
   identifying means arranged to identify all inverting and potentially-inverting logic units in the area;
   grouping means, arranged to group each inverting logic unit and potentially-inverting logic unit into at least one chain comprising inverting logic units and/or appropriately-configured potentially-inverting logic units, which chains form non-inverting paths through the area;
   overriding means arranged to override the reconfigurable area to implement the chains and the appropriate configurations of the potentially-inverting logic units during testing.

8. The system of claim 6, wherein
   the at least one chain comprises a plurality of potentially inverting logic units and/or inverting logic units, and the at least one chain further comprises at least one routing unit for interconnecting the plurality of potentially inverting units and/or inverting logic units, and
   the overriding means further comprises routing unit overriding means arranged to override the at least one routing unit to implement the chains.
9. The system of any of claims 7 or 8, wherein the grouping and overriding means are implemented in hardware by using the reconfigurable logic of the reconfigurable array.

10. The system of any of claims 7 or 8, further comprising:
register providing means arranged to provide a register to the output of at least one inverting logic unit; and
bypass providing means arranged to provide bypass means for selectively bypassing the register, such that the inverting logic unit can be used as a potentially-inverting logic unit by controlling the bypass means.

11. The system of any of claims 7 or 8, wherein the grouping and overriding means are implemented in software by alteration of the data output from the Automatic Test Pattern Generation (ATPG) tool.

12. A system for configuring a reconfigurable array of logic units in order to avoid inverting feedback loops during testing using an Automatic Test Pattern Generation (ATPG) tool, the system comprising:
partitioning means arranged to partition the array into a plurality of areas; and
a system in accordance with any of claims 7 to 11 arranged to configure each of the area partitioned by the partitioning means.

13. A reconfigurable logic device configured in accordance with the method of any of claims 3 or 4.
START

Identify LUs in device and which are: NINV; PINV; INV.

Set Scenario = 1

Is it possible to determine a chain of INV and non-tested PINV(i) which has an even number of INV+PINV(i) and comprises all INV?

YES

Determine longest chain of INV+PINV(i), where INV+PINV(i) is even.

Increase routing cost of each R used in chain.

Set all PINV in chain to PINV(i).

Set any PINV not part of chain to PINV(n).

X = X + 1

NO

Non-tested PINV(i) = 0?

YES

END

NO

Non-tested PINV(n) = 0?

YES

Determine shortest chain of INV+PINV(i), where INV+PINV(i) is even.

Set any PINV not part of chain to PINV(n) and any PINV which is part of chain to PINV(i).

Increase routing cost of each R used in chain.

X = X + 1

FIGURE 4
**INTERNATIONAL SEARCH REPORT**

**PCT/EP2010/061454**

**A. CLASSIFICATION OF SUBJECT MATTER**

**INV. G01R31/3185**

According to International Patent Classification (IPC) into both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G01R

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents :

**A** document defining the general state of the art which is not considered to be of particular relevance

**E** earlier document but published on or after the international filing date

**L** document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another document, or which is cited to combine the disclosures of other documents

**O** document referring to an oral disclosure, use, exhibition or other means

**P** document published prior to the international filing date but later than the priority date claimed

**T** later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

**X** document of particular relevance; the claimed invention cannot be considered novel, or cannot be considered to involve an inventive step when the document is taken alone

**Y** document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

**Z** document member of the same patent family

Date of the actual completion of the international search 21 April l 2011

Date of mailing of the international search report 04/05/2011

Name and mailing address of the ISA:

European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk

Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016

Authorized officer

Nadal, Rafael

Form PCT/ISA/210 (second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 7 509 547 BI (HAN UI SUN [US] ET AL) 24 March 2009 (2009-03-24) abstract; claims 1-19; figures 1-6 column 1, line 61 - column 2, line 24 column 2, line 51 - column 9, line 10</td>
<td>1-13</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>US 6966020 B1</td>
<td>15-11-2005</td>
<td>NONE</td>
</tr>
<tr>
<td>US 6223314 B1</td>
<td>24-04-2001</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 60010614 T2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2001144261 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 6574761 B1</td>
</tr>
<tr>
<td>US 7509547 B1</td>
<td>24-03-2009</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Form PCT/ISA/210 (patent family annex) (April 2005)