Asynchronous Test Hardware for Null Convention Logic

Nastaran Nemati, Mark C. Reed, Michael R. Frater
School of Engineering and Information Technology, University of New South Wales, Canberra, Australia
Email: nastaran.nemati@student.unsw.edu.au, mark.reed@unsw.edu.au, m.frater@adfa.edu.au

Abstract—**Asynchronous design is predicted to have a significant place in the future due to benefits of speed, power consumption, and design. Null Convention Logic (NCL) is a subcategory of asynchronous design that results in the most reliable and low-power asynchronous hardware. However, test strategies are not adopted to match the characteristics of this important asynchronous method, and the existing test methods for NCL use synchronous test hardware. In this paper asynchronous test hardware is proposed for testing NCL-based circuits to avoid problems of synchronous design as well as the need for asynchronous-synchronous interfaces and clock trees. The proposed test hardware is implemented in HDL and synthesised to define design characteristic and improvements over previous work.**

I. Introduction

Ever-increasing clock frequency and transistor shrinkage necessitate more complex clock tree distribution, and increase power consumption, electromagnetic interference and hot spots per chip. Clock-less or asynchronous design has been shown to solve these problems [1] and nowadays is gaining more attention where it is predicted in International Technology Roadmap for Semiconductors (ITRS) reports to be the majority share of fabricated electronics circuits (54%) by 2026 [2]. Little attention, however, has been given to this design method till now. For this reason, asynchronous design lacks commercialised design and test strategies, which has resulted in a lot of resistance against the wide usage of this beneficial method among designers. ITRS test reports [3] mention the lack of proper test and testability methods for asynchronous design and the necessity of investments in this area. Among the various subclasses of asynchronous design, Null Convention Logic (NCL) [4] has been shown to provide the most robust and low-power designs [1]. For this reason, this work focuses on test strategies for this subcategory of clock-less design.

Existing test strategies for NCL-based systems use synchronous methods [5], [6], [7] which may defeat the purpose of asynchronous design: the clock tree is being eliminated from design, but then added to it for testing purposes. However, this is not the only reason for seeking asynchronous test strategies. In several design for test (DfT) techniques the original design and test hardware are merged such that in some cases they are not distinguishable in functional mode [8]. When using synchronous DfT elements for asynchronous designs, these common DfT techniques are blindly eliminated from the list of options. Moreover, homogenously asynchronous design and test hardware enables reconfigurable DfT methods. Finally, homogenous asynchronous design and test hardware eliminates the need for massive synchronous-asynchronous interfaces that could cause metastability issues and increase the power consumption of the whole chip.

This paper is organized as follows. Firstly, Section II gives a brief introduction to the subclass of asynchronous design, NCL. Section III then discusses the limitations of the previous work while Section IV describes the proposed method. Section V provides the experimental results of this work, and finally Section VI concludes this paper.

II. Background: Null Convention Logic (NCL)

In NCL [4], to prevent overlap between consecutive data, each combinational block is forced to take one NULL or No-DATA value between each two data values. To force the circuit to take and maintain full-DATA and full-NULL until a new full-NULL or full-DATA arrives, a new type of gate with state holding behaviour, named m-of-n threshold (thmn) gates, is used (Fig. 1). Here n is the number of inputs of the gate, and at least m of these n inputs must transition from NULL to DATA to make the output of the gate transition from NULL to DATA.

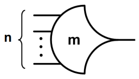


Fig. 1. NCL threshold m-of-n gate

In order to maintain the monotonic transition of combinational blocks between NULL and DATA, two other elements are required: NCL registration and completion detection circuits. The registration stage is made of th22 gates to hold the value of the upcoming DATA or NULL wavefront. When full-NULL or full-DATA is received in the registration stage, the completion detection circuit detects it and asks the previous functional block to send new DATA or NULL values, respectively.

1) Complexity Added to Testing: Although asynchronous method has many benefits, it also increases the level of complexity of testing. Asynchronous systems have non-deterministic behaviour in producing their results, and it is not clear when the results could be checked. Another factor of complexity is all the feedbacks and loops added to the circuit for handshaking. Finally, as with synchronous circuits, an asynchronous circuit can include functional loops which are important concerns when designing test hardware.

978-1-4799-3432-4/14/$31.00 ©2014 IEEE

III. Limitations of Previous Work

As mentioned, the existing test methodologies for NCL designs are based on synchronous test hardware. Among the previous work, [6] changes the internal structure of NCL registrations by adding one two-phase clocked latch for each bit of data. This approach does not consider the faults on the Gate Internal Feedbacks (GIFs), does not support adding test points (TPs) to the internal nodes of combinational blocks, and, most importantly, brings back clocking to the clock-less design. Reference [7] detects faults on GIFs by changing the internal structure of threshold gates and adding test enable input to all of them, which adds a significant amount of hardware overhead (average 13%). This study also adds clocked flip-flops to all the n+1 lines of n-bit registers after or before hard-to-test combinational blocks. This method too imposes a lot of hardware overhead to the DUT, does not address the internal hard-to-test parts of combinational blocks and adds clocking to the clock-less design.

Aside from adding a clocked structure to the clock-less design, the previous methods result in a heterogeneous design and test environment, which prevents the merging of the original hardware with test hardware and makes it hard or even impossible to have configurable DfT. This work is devoted to design and implementation of NCL-based DfT elements. The benefits of the existing test methods and the proposed work are compared in Table 1. In this table GIF shows whether the test method detects faults on gate internal feedbacks, FL represents the detection of faults on functional loops, and TP shows the ability of the test method to insert test points.

TABLE I
Comparison of DfT methods for NCL-based circuits

Test method	GIF	FL	TP	Clock-less	Configurable
[6]	×	✓	×	×	×
[7]	✓	✓	✓	×	×
Proposed	×	✓	✓	✓	✓

Based on this table, our proposed method only lacks the detection of the faults on gate internal feedbacks which will be addressed in future work. The next section describes the detail of the proposed asynchronous DfT elements.

IV. Proposed Method: Asynchronous DfT for NCL-based Design

Physical testing is the process of checking for physical defects which occur during or after fabrication. For simple circuits, testing can be done externally and by just applying test vectors to inputs and checking if the outputs match the expected values. Testing complex systems that have many registers and feedbacks and loops, however, is a harder and more time consuming task; applied test vectors may take a non-deterministic long time to set a stable internal state for the circuit. So, testing such systems may require modifications to the internal parts of the DUT. This concept is referred to as design for testability or DfT.

A. Hard-to-Test Parts of the Combinational Circuits

An internal node of the DUT whose value is difficult to adjust using the primary inputs is called hard to control; an internal node whose value is hard to propagate to a primary output is called hard-to-observe. Such nodes decrease the testability of the DUT and increase the cost of testing.

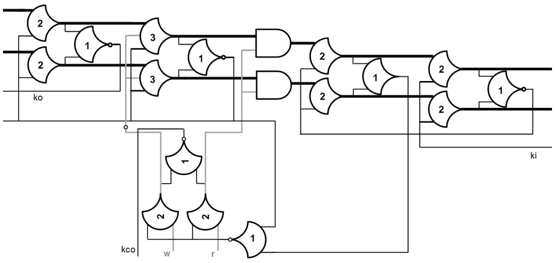


Fig. 2. NCL Read/Write Memory [4]

It is a common practice to insert controllability points (CP) and observability points (OP) into hard-to-control and hard-to-observe nodes, respectively. For synchronous design, CP insertion needs a flip-flop and a multiplexer at the desired node. OP insertion in clocked systems just needs a flip-flop whose output is directly connected to an extra primary output.

These clocked DfT elements are parts of the existing DfT hardware used for NCL-based design which, as discussed before, are desired to be replaced with their asynchronous counterparts. Fig. 3 depicts the Read/Write memory design in NCL [4] which could be used to design NCL DfT elements. This memory is made by separate read and write cycles, which allow controlling the storage and consumption of data passing through the NCL circuit wherever necessary.

1) Controllability Point: NCL-based CP could be designed by modifying the R/W memory of Fig. 3. The timing of each CP is controlled by the handshaking signals of the combinational circuit of the same stage. The designed NCL CP is shown in Fig. 4. The grey logic elements are the glue logic added to convert the read/write memory into a CP. The grey boxes marked as THXor0 are of a type of NCL threshold gate and function like a multiplexer. In functional mode these THXor0 gates pass through the data generated by combinational cloud ‘A’ to ‘B’, and in test mode they bypass the circuit’s original value (generated by ‘A’) and push the test data into combinational cloud ‘B’.

Pushing test data into the CPs requires an extra primary input for each CP. Since adding pins to the original circuit is very expensive, it is common to cascade all the added test points together like a chain and just add one serial input and one serial output for all of them. The HDL pseudo-code of the test controller for three cascaded CPs added to a design is shown in Fig. 5. Based on this pseudo-code, the test controller takes a two-phase procedure. Phase one is for shift-in, in which test data is pushed and shifted into the chain of the cascaded CPs. Two shift enablers are used which each are connected to the read port of one CP cell and to the write port of its two adjacent cells. This interleaving arrangement is to allow the test data travel through the cascaded cells from left to right without losing any data. Phase two is the test mode, in which

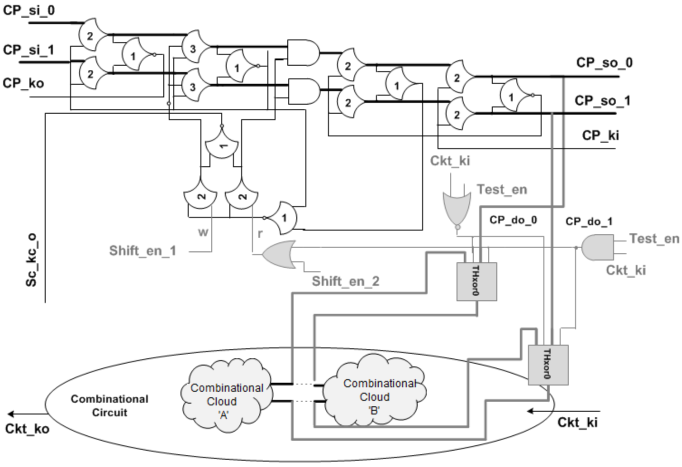
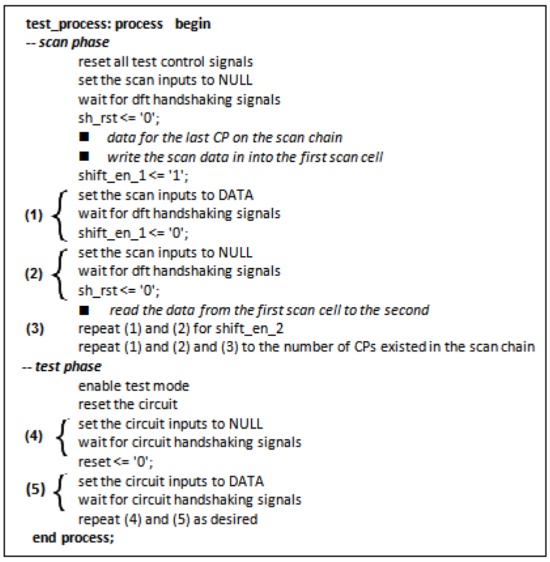


Fig. 3. NCL Controllability Point

the test data stored in the CP chain is applied to the circuit to control the value of the hard-to-control parts of the DUT.



```
test_process: process   begin
-- scan phase
          reset all test control signals
          set the scan inputs to NULL
          wait for dft handshaking signals
          sh_rst <= '0';
          ■   data for the last CP on the scan chain
          ■   write the scan data in into the first scan cell
          shift_en_1 <= '1';
(1)  {    set the scan inputs to DATA
          wait for dft handshaking signals
          shift_en_1 <= '0';
(2)  {    set the scan inputs to NULL
          wait for dft handshaking signals
          sh_rst <= '0';
          ■   read the data from the first scan cell to the second
(3)       repeat (1) and (2) for shift_en_2
          repeat (1) and (2) and (3) to the number of CPs existed in the scan chain
-- test phase
          enable test mode
          reset the circuit
(4)  {    set the circuit inputs to NULL
          wait for circuit handshaking signals
          reset <= '0';
(5)  {    set the circuit inputs to DATA
          wait for circuit handshaking signals
          repeat (4) and (5) as desired
 end process;
```

Fig. 4. HDL pseudo-code for cascaded controllability point

The proposed DfT elements are implemented in VHDL and evaluated by simulation using Mentor Graphics ModelSim$^{\text{TM}}$. The waveform for the discussed pseudo-code is represented in Fig. 6. Three CPs are added to a 4x4 multiplier. When the test enable signal, t_e, is zero the shift-in phase is done by setting the two shift enable signals. Then when t_e is set to one, the test data is applied to the circuit.

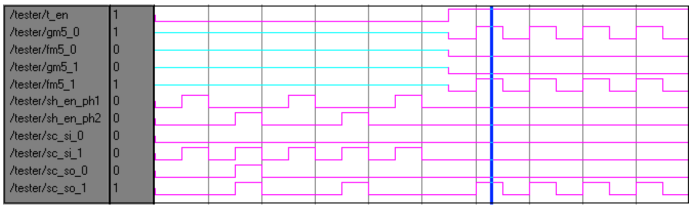


Fig. 5. Controllability point insertion (three cascaded CPs)- waveform

For the sake of comparison, one CP-inserted and one original circuit are instantiated. Value DATA1 (or 10) is inserted for two internal nodes and the effect is shown for the fifth bit of the outputs (comparing *fm* and *gm* signals for the DUT and the original circuits, respectively).

2) Observability Point: NCL-based OP is also designed by modifying the R/W memory. The timing of each OP is controlled by the handshaking signals of the combinational circuit of the next stage. The designed NCL OP is shown in Fig. 7. The grey logic elements are the glue logic added to convert the read/write memory into an OP. The THXor0 gates function as multiplexers and either collect the data to be observed from outputs of combinational cloud 'A' (in test mode) or pass the collected data through the cascaded OP cells (in shift mode).

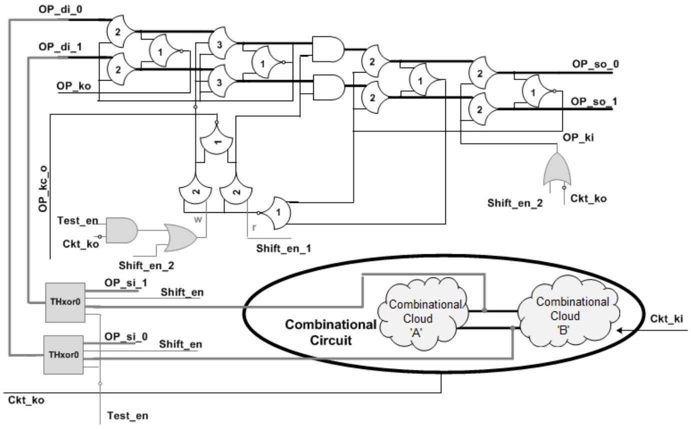


Fig. 6. NCL Obseravbility Point

Test controller for OP is very similar to the pseudo-code shown in Fig. 5. However, for OP, phase one is test phase and phase two is for shift-out. Also one of the shift enables is connected to the read port of all OPs and the other one to the write ports. This difference is to make the collected data shift right out from the last OP cell to the first one (to avoid losing the collected data).

The flow of test data through three cascaded OPs is shown in the timing diagram of Fig. 8. The vertical axis shows the flow of data in the time domain and the horizontal axis shows the spatial domain (data shifting through the cascaded OPs). The data stored in OP1 is highlighted with a grey border to make the flow of this data conspicuous as an example.

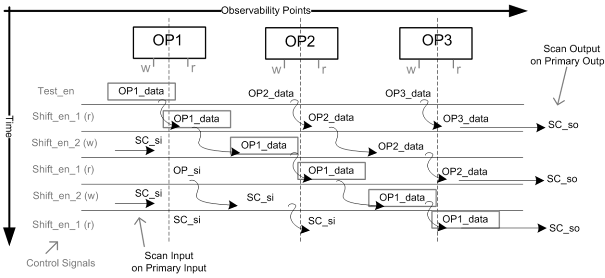


Fig. 7. Observability point insertion (three cascaded OPs)- timing diagram

B. Functional Loops

The general schematic of functional loops (FL) in NCL-based designs is depicted in Fig. 9. The most common way to deal with the complexity that FL add to testing is to break the FL by replacing the registers of the loop with scan chain registers. In scan chain technique all or some of the registers of the circuit are connected together, like a chain, so they could all be accessed serially from one primary input and through one primary output. In such cases, the scan chain sets the required state for a sequential system and then by applying one data to input, one data will be available at the output of the combinational circuit for collection.

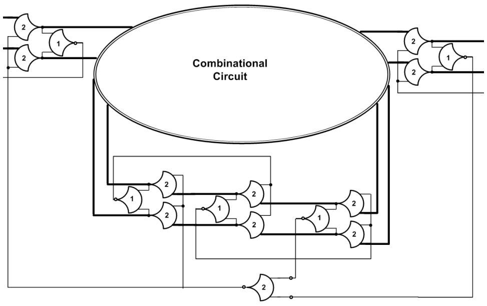


Fig. 8. Functional loop in NCL design

By adding some extra logic to the control lines of the R/M memory of Fig. 3 it could be turned into a scan register element to replace the registers at the functional loop of Fig. 10 and break this loop at test time. For increasing the testability of a register and the combinational circuit after it, scan chain cell could replace it.

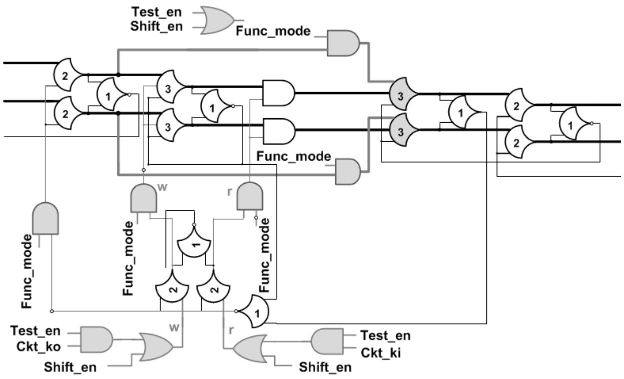


Fig. 9. NCL Scan Register breaking the functional loop

Adding DfT elements to the DUT makes it more testable and decreases the test time, but on the other hand it imposes some hardware overhead to the circuit. Since in scan chain insertion, the functional registers of the circuit are being replaced, the amount of the area overhead is smaller than that of CPs and OPs. In order to decrease the amount of hardware overhead, it is better to, whenever necessary, replace regular registers with CPs instead of scan cells, marked in table as "scan register (for Reg)". The specifications of the designed NCL-based DfT elements are tabulated in Table 2. In this table $T_{dt}(ns)$ and $T_{sh}(ns)$ are the worst case delay of each DfT element when dealing with the DUT data and when shifting the test data, respectively and are calculated using Altera Quartus$^{\text{TM}}$.

TABLE II
COMPARISON OF THE PROPOSED ASYNCHRONOUS DFT ELEMENTS

DfT module	Gate count	Tr Count	$T_{dt}(ns)$	$T_{sh}(ns)$
Single CP	22	212	0.27	0.13
Single OP	20	172	0.19	0.13
Cascaded CPs	23	214	0.31	0.16
Cascaded OPs	24	216	0.28	0.16
Scan register (for FL)	28	120	0.41	0.57
Scan register (for Reg)	19	184	0.14	0.09

This table shows that "scan register (for Reg)" is the best DfT choice, both in terms of area overhead and test time. The option with the worst specifications is "scan register (for FL)". As mentioned before, scan cells for FLs are the most frequently used test hardware for testing electronic systems. However, since they are applied just for functional loops, not many of this DfT element is required for each design under test. The overhead that this DfT imposes to the DUT is, therefore, not significant compared to other DfT elements. The area overhead and delay of CPs and OPs are somewhere between those of scan cells for registration stages and scan cells for functional loops. These test elements are to be added to the DUT based on testability measurements of the combinational blocks.

V. CONCLUSION

Asynchronous design for test techniques for NCL-based designs are missing from the literature. In this work DfT elements such as controllability point, observability point, and scan chain elements are designed and implemented in HDL to help integrate test hardware with the design hardware and increase the capability of configurable test hardware. Also eliminating clock tree and repeaters from the circuit may increase the reliability, performance, and energy saving of the circuit both in the test and functional mode.

REFERENCES

[1] J. Di and S. C. Smith, "Ultra-low power multi-threshold asynchronous circuit design," Jul. 12 2011, uS Patent 7,977,972.

[2] "ITRS Reports - Design," http://www.itrs.net/, 2011, [Online; accessed Sept., 2013].

[3] "ITRS Reports - Test," http://www.itrs.net/, 2011, [Online; accessed Sept., 2013].

[4] K. M. Fant, *Logically determined design: clockless system design with NULL convention logic*. Wiley-Interscience, 2005.

[5] V. Satagopan, B. Bhaskaran, W. Al-Assadi, S. Smith, and S. Kakarla, "DFT Techniques and Automation for Asynchronous NULL Conventional Logic Circuits," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1155–1159, 2007.

[6] A. Kondratyev, L. Sorensen, and A. Streich, "Testing of asynchronous designs by "inappropriate" means. Synchronous approach," in *Proc. IEEE Int. Symp. on Asynchronous Circuits and Systems, 2002 (ISCAS'02)*, pp. 171–180.

[7] W. Al-Assadi and S. Kakarla, "Design for Test of Asynchronous NULL Convention Logic (NCL) Circuits," in *Proc. IEEE Int. Test Conf., 2008 (ITC'08)*, pp. 1–9.

[8] N. Nemati, A. Simjour, A.-A. Ghofrani, and Z. Navabi, "Optimizing Parametric BIST Using Bio-inspired Computing Algorithms," in *Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems, 2009 (DFT'09)*, pp. 268–276.