On Testing Prebond Dies with Incomplete Clock Networks in a 3D IC using DLLs

Michael Buttrick and Sandip Kundu
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst Massachusetts, USA
{mbuttric, kundu}@ecs.umass.edu

***Abstract*—3D integration of ICs is an emerging technology where multiple silicon dies are stacked vertically. The manufacturing itself is based on wafer-to-wafer bonding, die-to-wafer bonding or die-to-die bonding. Wafer-to-wafer bonding has the lowest yield as a good die may be stacked against a bad die, resulting in a wasted good die. Thus the latter two options are preferred to keep yield high and manufacturing costs low. However, these methods require dies to be tested separately before they are stacked. A problem with testing dies separately is that the clock network of a prebond die may be incomplete before stacking. In this paper we present a solution to address this problem. The solution is based on on-die DLL implementations that are only activated during testing prebond unstacked dies to synchronize disconnected clock regions. A problem with using DLLs in testing is that they cannot be turned on or off within a single cycle. Since scan-based testing requires that test patterns be scanned in at a slow clock frequency before fast capture clocks are applied [1], on-product clock generation (OPCG) must be used. The proposed solution addresses the above problems. Furthermore, we show that a higher-speed DLL is better suited to not only high frequency system clocks, but lower power as well due to a smaller variable delay line.**

Keywords-3D integrated circuit testing, delay lock loops, low power testing, on-product clock generation

I. INTRODUCTION

As today's designs become increasingly complex and require more area, vertical stacking of silicon dies (3D integration) allows for larger designs with lower power. This is accomplished by reducing the average interconnect length by partitioning the design among a few dies and stacking them such that units with high connectivity are now much closer than in a 2D design. The connections between the dies are made using Through Silicon Vias (TSVs).

In order to take advantage of the 3D organization, a low-power 3D clock distribution must also be implemented. To function properly, the clock distribution on stacked dies should operate synchronously with the lowest possible skew across the die-stack. There are two possible implementation choices here. The first choice is to implement a separate clock distribution network in each die and when the dies are bonded together, TSVs are used to make redundant connection between the clock networks to reduce skew across the stack.

The main benefit of this approach is that each prebond die can be tested separately without any concern about stacking. The disadvantage of this approach is that it adds significantly to the area, capacitance and power. Since the clock distribution network in a 2D processor can account for well over 25% of the total power [2], it is essential that the clock distribution network is not replicated on all dies. An example of a power-saving 3D clock distribution network is shown in Fig. 1 where one die has a complete network, but additional dies only have local connections. This need not be the case; in fact, PLLs may be fabricated one die in a more mature process while high-level buffers are on another die in a newer process that provides higher drive current. While such clock networks demand less power during operation, they have the drawback that each individual die may not have a complete clock distribution network before bonding. This, of course, leads to test problems as the entire die is not functional and individual TSVs are too small and dense to probe using conventional probe cards [3].

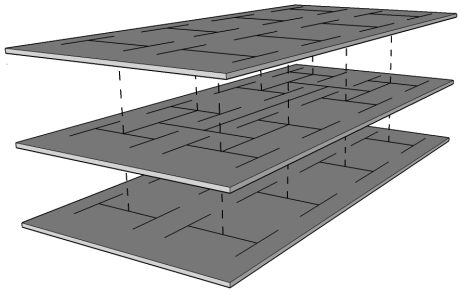

Fig. 1: Three dies in a 3D stack, the top and bottom of which have incomplete clock networks to save power. Dashed lines represent TSV connections

Because a formidable roadblock facing the widespread adoption of 3D integrated circuits is the less-than-adequate methods of testing prebond dies, a solution is needed to so that pre-bond dies can be tested before bonding.

A promising solution is one whereby TSVs that represent cut points in clock distribution networks are driven by Delay Lock Loops (DLLs) to ensure that the disparate local clock networks fed by each clock probe site are in sync and operate with low skew during testing. While this solves a basic clocking problem, it lacks the clock period adjustments needed for at-speed tests. Free running oscillators do not permit clock period modulation.

978-3-9810801-7-9/DATE11/©2011 EDAA

In modern designs that are tested using a scan methodology, it is necessary to perform scan operations at a slower scan speed to stay within a test power envelope. After the patterns are scanned in, typically, an at-speed clock is applied for system capture to expose any timing issues brought about by manufacturing defects. Since a DLL requires several cycles to lock onto a reference frequency, starting and stopping the clock from a tester is not an option. Thus, while incorporating DLLs to solve the DC test problem, a new solution is still needed for the at-speed test. To address this problem, we present modifications to the DLL-based 3D IC testing solution to allow for on-chip generation of scan and test clocks from a single reference clock. In evaluating the quality of our solution, we will focus primarily on the power usage of the solution, as minimizing test power is becoming an increasing concern [1].

The contribution of this work is means to use a DLL in testing while allowing for separate scan and test clock frequencies. Since a DLL cannot change operating frequencies or be turned on and off during testing, the proposed work is essential to the testing of incomplete clock networks in 3D ICs using DLLs.

The rest of this paper is organized as follows: in Section II we review previous work. Section III gives insight into how DLLs can be used in 3D testing while Section IV details the implementation of two methods of scan and test clock generation. In Section V we will examine results of the two methods as well as present a case study in test overhead. Finally, Section VI concludes the paper.

II. Previous Work

There have been many published works regarding On-Product Clock Generation (OPCG) in recent years, and as such a full exploration of the topic will not be possible. However, a short survey of relevant papers is necessary to motivate our work presented here. In an early paper, von Kaenel *et al* present a solution dealing with on-chip clock generation using a Phase Lock Loop (PLL) in [10]. Manufactured in a 0.35µm process, this PLL operates at a reduced supply voltage to decrease power consumption. While providing a quality solution to generation a low-jitter clock on-chip from a slower input clock, this result of this work is not suited to scan-based design testing.

Beck *et al* present a hardware implementation to generate a fast, at-speed clock from the slower scan clock, and allow both to be passed to the clock distribution network to be used in scan-based testing in [5]. To help with design, Uzzaman, Li, Keller, and Snethen propose a methodology to aid in the specification and design of the OPCG using an automated flow and standard cell libraries in [1]. It is taken another step further Pei, Li, and Li in [7] by providing a means by which the test clock frequency can be specified in the test patterns that are shifted in, providing for more flexibility.

While all the implementations above are concerned with PLLs, Farjad-Rad *et al* use a DLL for on-product clock generation in [6]. The solution in this work is intended for high-speed serial data links where low-jitter clock and data recovery (CDR) is necessary. However, this solution has no provisions for test. The work is touted to be low-power, but in a 0.18µm process, its 18mW power consumption is outdated and could be greatly improved upon.

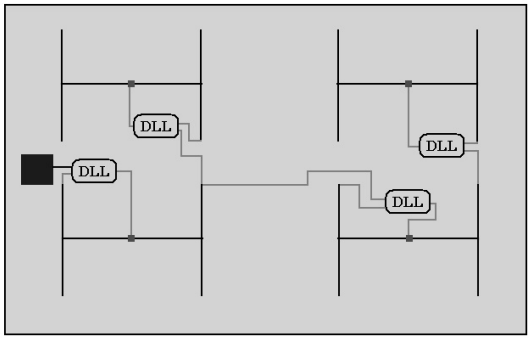

Fig. 2: Using DLLs to synchronize disconnected clock networks

So far we have seen no works that allow for OPCG using a DLL that will generate scan and test clock frequencies. Since our solution to test dies with incomplete clock distribution networks in 3D ICs uses DLLs, this is a critical issue we have addressed and will present here. Two solutions will be shown; one that follows a methodology similar to [1][5][6] and [7], then another method which accomplishes the same goal while using much less power.

III. 3D Testing of Incomplete Clock Networks

The core problem involves testing pre-bond dies with several disjoint local clock networks. One solution is to connect them using central and regional clock buffers and distribution networks. We rule out this solution as it is too expensive in terms of area and power. Another solution is to drive TSVs from testers. Probing issues prevent such a solution. A small TSV microcontact cannot be probed repeatedly by a probe pin which has to be dimensionally similar as it will not have the mechanical strength, nor reliability or electrical fidelity. Another solution is to daisy chain the clock distribution networks by taking the clock from the closet local leaf node as the driver for the adjacent local clock tree. In this case only one pin needs to be driven from tester and the routing requirements are minimal. Ordinarily, this solution will be ruled out due to excessive clock skew. However, this is the solution we focus on.

DLLs are used to ensure that local clocks do not have skews resulting from such chaining. This can be seen in Fig. 2. The large pad is a probe point where a clock signal will come directly from the tester. The small squares in the middle of the small trees represent TSVs. Since a DLL is used in the local network directly connected to the clock probe, the clock sinks will be locked to the clock from the tester. Subsequent local networks receive their clock signal from the closest live

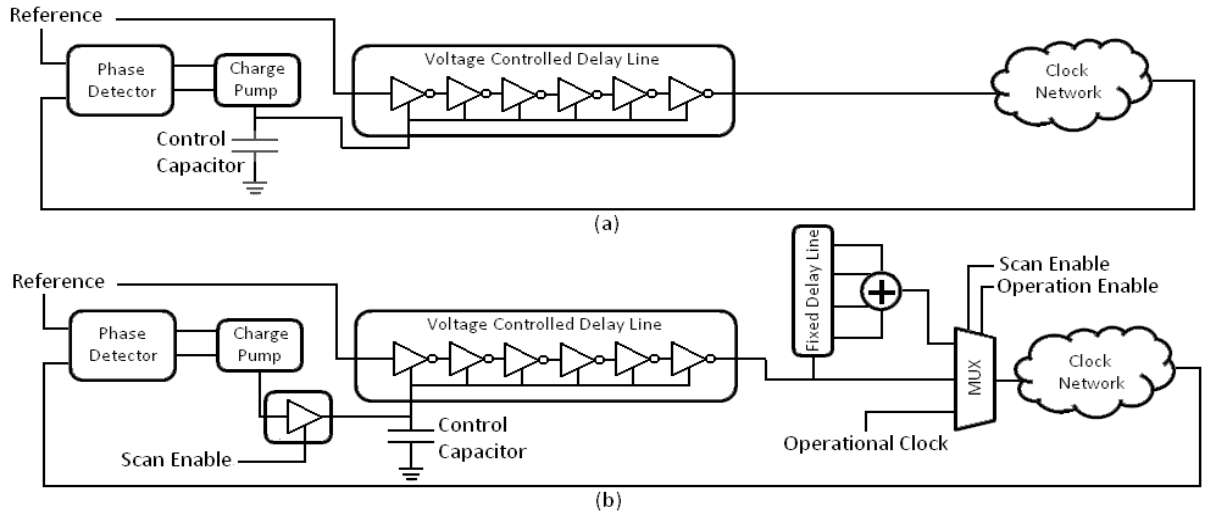

Fig. 3: Block diagrams of typical DLL (a) and DLL with circuitry to generate and apply test clock from supplies scan clock (b)

network and use a DLL to ensure its clock sinks are also aligned with the test clock.

The use of DLLs in testing is problematic because, as they require several clock cycles to lock, they cannot be turned on and off as needed for scan-based testing. As such, the contribution of this work is a means by which the DLLs can be always on and be used to supply the appropriate clock frequencies for test.

IV. DLL-Based Clock Generation for Test

Using DLLs succeeds in connecting the disconnected trees with low skew. However, because of the nature of scan-based testing and the fact that it takes several cycles for a DLL to lock, we need to examine suitable on-product clock generation (OPCG).

A. Need for Scan and Test Clocks

Phase Lock Loops (PLLs) are widely used for OPCG. Generally, a clock much slower than the operating frequency of a design is supplied to the chip where a PLL multiplies it and feeds it to the clock distribution network. However, we have already used DLLs to synchronize disconnected clock trees in a 3D die rather than PLLs because of their stability, smaller area/power requirements, ease of implementation, and that they do not accumulate jitter [6]. These DLLs can be used along with some logic to generate the necessary clock rate for scan-based testing as well as operation.

The basic block diagram of a DLL is shown in Fig. 3a. The phase of a reference clock coming from the tester through the nearest probe site is compared to that of the clock signal at a leaf node of the clock network driven by the DLL. Based on the phase error between the two, the charge pump will either increase or decrease the voltage on the control capacitor to decrease or increase the delay of the Voltage Controlled Delay Line (VCDL), respectively. In this way, it is ensured that the clock signal arriving at the clock syncs is in phase with the reference clock.

However, in scan based testing, it is necessary to scan in test patterns at a slow clock frequency (usually around 100MHz) then apply two clock pulses (a launch and a capture pulse) at the intended operating frequency of the design.

The captured result is then scanned out at the scan frequency as a new pattern is scanned in. The result is then compared to the expected result and a determination is made as to whether the design performed as expected.

Because a DLL needs several cycles to lock, it would be impossible to switch it on and off while patterns were being scanned in. So, we develop a means of OPCG that takes a signal from the tester and generates the appropriate scan and test clocks. Following the example of [1][5][7], we will first examine the case in which the tester is supplying the scan clock frequency. We will be referring to this as Case 1.

B. Test Clock Generation from Scan Clock (Case 1)

This first case is where the scan clock is effectively multiplied to generate the test clock. Fig. 3b shows how this can be accomplished.

The multiplexer whose output feeds the clock network is really two cascaded multiplexers similar to the scheme found in [7], with the exception being that the Operation Enable signal controls the second MUX, so that when in operation mode, the Operational Clock is fed to the clock network, and the value of Scan Enable is inconsequential. While scanning in and out patterns, the DLL is operating at the scan frequency as supplied by the tester, Scan Enable is asserted, Operation Enable is deasserted, and the scan clock is fed to the clock network.

When the pattern has been scanned in and an at-speed clock is to be applied (which we will refer to as the test clock), Scan Enable and Operation Enable are deasserted, and the clock network is fed a signal that comes from the Exclusive-Or of four taps from a Fixed Delay Line (FDL). These taps are chosen so that the delay between each tap is one-half the period of the desired test clock frequency. When these taps are fed through an XOR tree, the output will be a launch and capture pulse at the test clock frequency

Furthermore, this XOR output can be passed through an AND gate with the output from Tap 1 to ensure that only one set of launch and capture pulses are generated during one period of the scan clock. Also note the buffer located between the charge pump and the control voltage capacitor in Fig. 3b. This can either be an AND gate or a transmission gate that will ensure that the control voltage will not change when the test clock is being applied. Because the signal that is sent to the clock network will not correspond to the reference clock, this gate is needed to keep the control voltage from changing while the test clock is being applied, which would cause the DLL to unlock.

While we see that this is a valid approach for using a DLL with some addition circuitry for OPCG, it turns out that this solution has some undesirable features that make it not ideal for our target application which requires low on-chip power dissipation. In previous works examined, it made sense to generate a faster clock on chip because they used a PLL to provide their faster clock. To generate a faster oscillation in a PLL, less delay and thus fewer cells are needed in the Voltage Controlled Oscillator (VCO). However, in requiring our DLL to run at a low frequency, a larger delay range is needed in the VCDL. This is because in order for the DLL to be able to lock for all phase shifts possible in the clock network, the delay range must be equal to one full period of the reference clock. If the DLL is required to run at 125 MHz (representing a typical scan clock) the range of delay for the VCDL must be at least 8ns. This corresponds to a very large number of variable delay cells. Even though lower switching frequencies generally correspond to lower power, because of the hardware overhead needed to run at such a low frequency, this solution will consume more power than running at a faster frequency. Add to that the power used by the FDL, and it becomes clear that when using a DLL for OPCG, generating the test clock from the scan clock may not be the ideal solution. With that in mind, we must now explore another option.

C. Scan Clock Generation from Test Clock (Case 2)

Unlike Case 1, here we examine the case where the scan clock is generated by dividing the output of the DLL, which is running at the test clock frequency.

In previous works, it was attractive to supply the chip with a slow clock used for scan-based testing and generating a faster test clock on chip using a PLL. It was often stated that this would allow for simpler test equipment. However, when using a DLL to sync a clock signal between disconnected

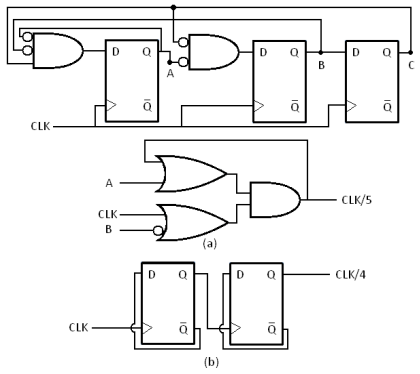

Fig. 4: Circuits to (a) divide clock by 5 [4]and (b) 4 with 50% duty cycles

clock networks in the case of a prebond 3D IC, supplying a die with a scan clock and generating a test clock leads to high on-chip power dissipation. For this reason, we will explore the situation where the die is supplied with a clock signal running at the test frequency (which is also the intended operational frequency, if not slightly higher) and generating the scan clock on-chip. We will refer to this as Case 2.

While this means running the DLL at higher frequencies, significant hardware savings make this a lower power solution. As mentioned previously, the delay range of the VCDL must be equal to one full period of the intended operating frequency of the DLL in order for it to be able to lock for all possible delays. Running the DLL at four times the frequency roughly means four times fewer delay cells in the delay line. Also, the divider circuit at the output of the DLL to generate the scan clock can be rather straightforward. If the scan frequency is one-quarter the test frequency, just two flip-flops are needed to divide the clock. Similarly, a division of 8 would require three flip-flops. Slightly more complicated is the case where one would want to divide by a number that is not a power of two, but these circuits are well known and incur less overhead than needed to implement the fixed delay line needed in Fig. 3b. A few simple dividers are shown in Fig. 4. To divide a clock by 10, for example, the divide by 5 circuit could be followed by a single flip-flop to divide again by 2.

The full visual of this solution can be seen in Fig. 5. While it may not appear on the surface to be much smaller than the implementation in Fig. 3b, the VCDL can be much shorter because of the higher running frequency and the dividing circuitry requires many fewer transistors than the FDL in Case 1. This reduced circuitry results in lower power consumption despite the higher frequency.

Note that in this case, after the DLL is locked, the charge pump will be disconnected from the control voltage capacitor during the scan operation. When the Scan Enable signal is lowered and the launch and capture pulses are applied to the logic, the DLL will adjust the control voltage if it has drifted.

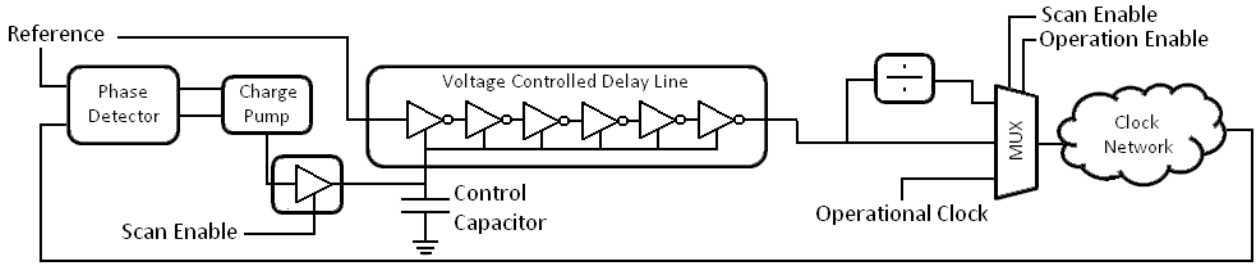

Fig. 5: Block diagram of DLL with provisions to generate Scan clock from Test clock

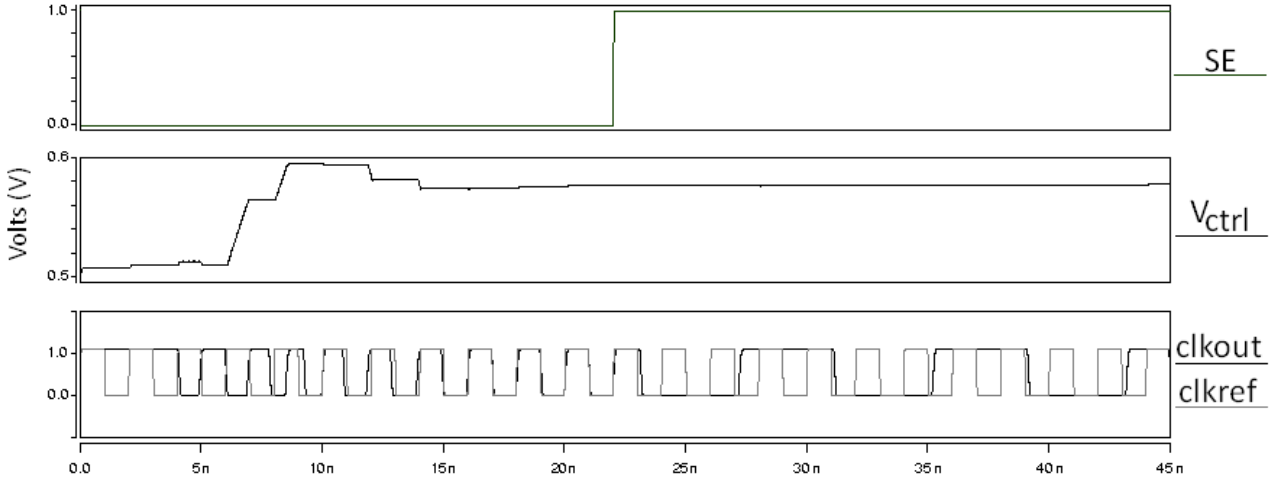

Fig. 6: SPICE waveform of DLL OPCG implementing Case 2. DLL runs at 500MHz with 125MHz scan clock applied when scan enable (SE) is high.

V. RESULTS

A. *Comparison of Case 1 and Case 2*

We will now present the results of the two cases. In each, the scan clock frequency is set at 125MHz, which is in the typical range of scan frequencies, and the test and operational frequency is set at 500MHz, which would be typical of an ASIC design.

In the first case, the DLL will run at 125MHz and a 500MHz clock will be generated by the XOR of the taps from the FDL. In order for the DLL to be able to compensate for any delay, there must be a delay range of 8ns in the VCDL, the period of the reference clock.. Since each delay cell can provide a delays in the range of 85-244ps, 51 delay cells are needed to provide a full period delay range. To generate the test clock, a fixed delay line must be constructed with 4 taps with 1ns delay between each. If the first tap is considered to be the input, this delay line will have a total delay of 3ns.

Additionally, since it is essential that the delay though this fixed delay line remain constant, the delay cells must be constructed to be insentitive to supply voltage and temperature variations. In the second case where the tester supplies the funtional clock frequency and the scan clock is generated, the DLL will operate at 500MHz and a divide by 4 circuit will be used to generate the 125MHz clock. In this case, the VCDL must have a delay range of 2ns, and thus requires 13 delay cells, not surprisingly approximately 4 times fewer than in the Case 1. The divide-by-4 counter is simply two D flip-flops connected as shown in Fig. 4.

The numbers presented in Table 1 are for all the blocks included in Fig. 3b and Fig. 5 for Cases 1 and 2, respectively. It may seem surpising that running the DLL faster would require less power, but when one considers the extra hardware that is necessary not only to construct a suitable VCDL, but the fixed delay line as well, the power numbers are understandable.

For the scan and test frequencies presented here, it is clear that when low on-chip power is required, the second case where the DLL is run at the test frequency is the best choice. These frequencies presented are representative of ASIC designs, but does the comparison yield the same result if the test frequency is higher? If the scan frequency is kept the same at 125MHz and the test frequency is increased to 2GHz, the power for Case 1 would actually likely decrease slightly. This is because while the size of the VCDL would stay the same, the FDL would shrink since at a higher frequency it would have to present a smaller total delay.

At this higher test frequency, the size of the VCDL in Case 2 would decrease and would only require 4 delay cells. In our simulations, increasing the test frequency only decreases power consumption in Case 2 because of the need for fewer voltage-controlled delay cells. This is because the VCDL is

TABLE I. HARDWARE AND POWER COMPARISON OF TECHNIQUES

	DLL Freq (MHz)	*VCDL Cells*	*Transistor Count*	*Power (mW)*
Case 1	125	51	2086	2.56
Case 2	500	13	458	0.82

constructed using current-starved inverters to provide variable delay. When the control voltage is low and considerable delay is required, the number of delay cells dominates the power consumption, at least for the range of frequencies over which the DLL is capable of operating, which tops out at approximately 2GHz. A waveform of an implementation of Case 2 is shown in Fig. 6. The DLL is running at the test frequency and is divided to generate the test clock which is applied to the clock distribution network when the ScanEnable signal is high.

B. Case Study of IA-64 microprocessor

To understand the overhead of the use of Case 2 for testing would impose on a design, we will examine the IA-64 microprocessor presented by Rusu and Singer in [9]. This implementation of the IA-64 architecture has a highly parallel execution core that achieves high performance by using an explicitly parallel instruction computing (EPIC) design. The chip is fabricated in 0.18µm process and has an operating frequency of 800MHz, which makes it a suitable candidate for our study.

Assume for simplicity that this microprocessor has been chosen to be integrated into a 3D stack containing other logic and/or memory dies. According to the design presented in [9], the clock distribution for the chip has eight regional clock networks with associated buffers. If this chip were part of a 3D stack, these eight buffers would be ideal points to feed a clock signal through a TSV from an adjacent die with a complete clock distribution network. This would eliminate the global clock network and leave just the regional networks on this die.

Since there are eight disconnected clock networks before the dies are bonded, eight of the DLLs with provisions for scan-based testing presenting in this paper will be necessary for at-speed testing. While no information is given in terms of area of the IA-64 microprocessor, it is reported that the design consists of 25.4 million transistors [9]. As shown in Table 1, a DLL suitable for OPCG during testing implemented in Case 2 running at 500MHz contains 458 transistors. Since eight DLLs are needed, the total test overhead using this method is 3,664 transistors. This accounts for a mere 0.014% overhead for test logic. Note too that this number is expected to decrease if the test frequency is increased to 800MHz to match the operating frequency of the original design as the number of delay cells in the VCDL will decrease. Thus, the expected overhead of the proposed solution is negligible.

VI. CONCLUSION

We have presented a means to perform On-Product Clock Generation (OPCG) for testing using DLL. This was necessary to manage skew between local clock networks. However, DLLs cannot be stopped and started or change frequencies as required for scan-based at-speed testing. Our modified solution addresses that problem. This work is unique because previously, only PLLs have been used for OPCG where scan-based testing is used. In one previous work on OPCG using a DLL, no provisions had been made for testing.

We examined two possible cases in which a scan and a test clock can be generated on chip. Case 1 follows previous works where the scan frequency is supplied by a tester and a fixed delay line is used to generate the test clock. Because of the amount of hardware needed for this approach, a better solution is running the DLL at the test frequency and dividing that signal to generate the scan clock. The main driving factor behind our analysis is to find the solution which requires the lowest power dissipation during testing. Our results have shown that Case 2 where the tester is run at the operating frequency and the scan clock is generated by a divider circuit consumes considerably less power regardless of the test frequency. The proposed solution enables (i) testing of pre-bonded dies of a 3D IC through use of DLLs and (ii) scan and test clock application during testing which has not been attempted previously with DLLs.

REFERENCES

[1] A. Uzzaman, B. Li, B. Keller, and T. Snethen, "Using programmable on-product clock generation (OPCG) for delay test," in *Asian Test Symposium*, 2007.

[2] E. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proceedings of the IEEE,* vol. 89, no. 5, pp. 665-692, May 2001.

[3] E. J. Marinissen and Y. Zorian, "Testing 3D chips containing through-silicon vias," in *International test conference*, 2009.

[4] M. Arora, "Clock dividers made easy," *Design Flow & Reuse*, ST Microelectronics Ltd, SNUG Boston 2002.

[5] M. Beck et al., "Logic design for on-chip test clock generation-implementation details and impact on delay test quality," in *DATE*, 2005.

[6] R. Farjad-Rad et al., "A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1804-1812, December 2002.

[7] S. Pei, H. Li, and X. Li., "An on-chip clock generation scheme for faster-than-at-speed delay testing," in *DATE*, 2010.

[8] S. Ravi, "Power-aware test: challenges and solutions," in *International Test Conference*, 2007.

[9] S. Rusu and G. Singer, "The first IA-64 microprocessor," *IEEE Journal of Solid State Circuits*, vol. 35, no. 11, pp. 1359-1544, November, 2000.

[10] V. von Kaenel, D. Aebischer, C. Piguet, and E. Dijkstra, "A 320MHz, 1.5mW at 1.35V CMOS PLL for microprocessor clock generation," in *ISSCC*, 1996.