Efficient testing and diagnosis of faulty power switches in SOCs

S.K. Goel, M. Meijer and J. Pineda de Gyvez

Abstract: The use of power switches in modern system chips (SOCs) is inevitable as they allow for efficient on-chip static power management. Leakage is one of the main hurdles in low-power applications. Power switches enable power gating functionality, that is one or more parts of the SOC can be powered-off during standby mode thus leading to savings in the SOC's overall power consumption. To this end, a circuit and a method to test power switch is presented. The proposed method allows for the testing of on/off functionality. In case of segmented power switches, individual failing segments can be identified by using the proposed test strategy. The method only requires a small number of test patterns that are easy to generate. Furthermore, the proposed method is very scalable with the number of power switches and has a very small area-overhead.

1 Introduction

The ongoing miniaturisation of the integrated circuit feature size has had a significant impact on the chip's size, performance, and power consumption. With every technology node, the circuit performance is improved because of shorter transistor channel length, lower threshold voltage, and reduced gate-oxide thickness. This, however, will lead to increased leakage power because of increased sub-threshold leakage and gate oxide tunnelling current. As feature sizes shrink below 100 nm, leakage power becomes as important as dynamic switching power in many applications [1–3]. To minimise both dynamic switching power as well as leakage power dissipation, modern system chips (SOCs) require efficient power management. For that purpose, it is becoming common practice to provide the whole SOC, or some parts, with power management modes. In general, two modes can be distinguished: (1) active mode, and (2) standby mode. In active mode, the SOC (or part of it) is able to perform the function for which it was designed, while in standby mode, SOC (or part of it) is idle.

To minimise standby power consumption, power supply gating is used. To enable power gating functionality, different parts of a SOC are equipped with one or more power switches [4–6]. For example, in a core-based SOC [7, 8], every individual core can be equipped with one or more power switches. Based on the activity in the SOC and the data transactions between different cores, a core can be individually turned-off through the power switch. In this way, the leakage power is minimised for the core (which is turned-off), thus leading to savings in the over-all power consumption.

doi:10.1049/iet-cdt:20060147

Paper first received 8th September 2006 and in revised form 8th January 2007

S.K. Goel is with NXP Semiconductors, Corporate Research–Design Methods & Solutions, 5656 AE Eindhoven, The Netherlands

M. Meijer and J. Pineda de Gyvez are with NXP Semiconductors, Corporate Research–Systems & Circuits, 5656 AE Eindhoven, The Netherlands

S.K. Goel is now a Member of Consulting Staff at Magma Design Automation, San Jose, CA

E-mail: sgoel@magma-da.com

Fig. 1*a* shows a conceptual example of a core-based SOC with four cores, which are connected to the power supply (VDD) via power switches. To minimise the risk of not working and ease of manufacturing, instead of a large power switch, segmented power switches [5] are fabricated in practice. A segment can contain one or more transistors. Therefore, each core in a SOC is connected to a number of small power switches. Fig. 1*b* shows an example implementation of such a power switch.

Fault-free operation of power switches is critical for the functional operation of the SOC. In the case of faulty switches, the SOC can suffer from a performance loss or may not even operate in the worst case. Therefore, the testing of these switches for manufacturing defects is very important. In this paper, we present a simple yet effective design-for-test (DfT) circuitry together with a method to test power switches. The proposed method also enables the identification of individual failing segments in the case of segmented power switches. The area-overhead for the proposed DfT circuitry is very small.

2 Prior work

Multiple supply voltages together with power switches are increasingly used to minimise both dynamic and static power consumption. Lackey *et al.* [9] proposed a system architecture called 'voltage islands'. In a voltage islands architecture, the complete chip is divided among different islands, where every island contains one or more logic blocks and has a unique power characteristic. In other words, different islands use different on-chip voltage supplies. Furthermore, every island can be equipped with one or more power switches to enable power gating functionality. Similarly, Puri *et al.* [10, 11] described techniques to minimise the power consumption by using different supply and threshold voltages. Power Wise Interface (PWI) Open Standard [12] defines a serial interface between a SOC power supply controller and an external voltage regulation system. The external voltage regulation system allows dynamic adjustment of the on-chip supply voltage and the threshold voltages.

Mostly, these papers assume implicit testing of power switches when the chip is powered-on. However, this type

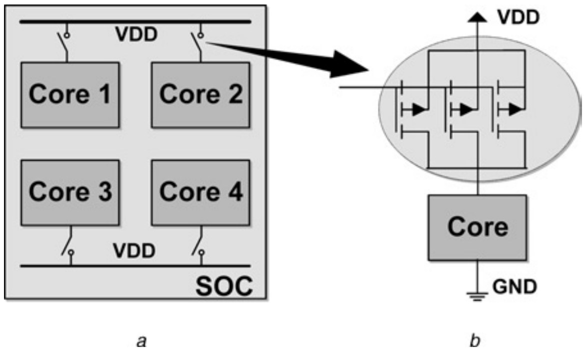

Fig. 1 *Example of core-based SOC equipped with power switches*

a Core-based SOC with four cores
b Implementation of the power switch

of testing is not sufficient when one needs to find out whether all power switches are working correctly or not. To understand this, let us consider an example chip with one switch partitioned in to n segments out of which, j are defective. Let us also assume that during power-on, the chip still works. Observe, however, that the correct functional operation of the chip does not necessary mean that all n segments are functional. In fact, the j faulty segments will escape fail-detection. If this is the case, the remaining $n-j$ switches will draw more current and it is possible that with time, some of them will breakdown and cause the chip to malfunction. Moreover, the SOC's performance can be hampered during active-mode operation because of the additional voltage drop across the switch due to the failing segments. Therefore, it is very important to identify individual failing-power switches. To the best of our knowledge, there are no papers available in the literature that describe the testing and diagnosis of power switches in SOCs. In this paper, we present a simple, yet effective, circuit and method to test and diagnose power switches.

3 Types of power switches

Several techniques for power switch realisation are presented in [4–6]. These techniques describe the use of one or more PMOS or NMOS transistors as power switches. Based on the requirement, these transistors can be controlled individually or by a common input signal.

Fig. 2 shows the three types of power switch configurations that are widely used. Fig. 2*a* shows an example of a power switch known as 'header switch'. A header switch is a PMOS transistor that is controlled by a dedicated control signal *stand by*. When the *stand by* signal is active (e.g. logic '1'), the header switch does not conduct. In this situation, the power supply to the core is gated and the core does not operate functionally. When the *stand by* signal is inactive (e.g. logic '0'), the header switch conducts and the core is able to operate. An alternative to the header switch is the so-called 'footer switch', which is an NMOS transistor and is used to control the ground line (GND). Fig. 2*b* shows an example of a symmetric header and footer switch. In this case, there are two control signals *hstand by* and *fstand by*. However, the control signal *fstand by* is usually the inverse of the signal *hstand by*. The operation of this switch can be easily derived from the operation of the header switch.

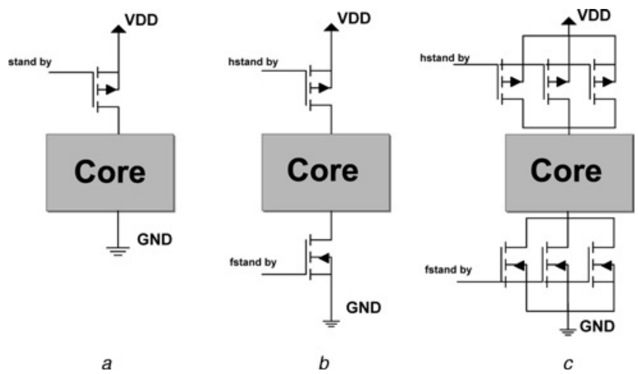

Fig. 2 *Examples of power switches*

a Header switch
b Symmetric header and footer switch
c Segmented symmetric header and footer switch

Another popular way to implement power switches is via a number of transistor segments, where every segment can contain one or more transistors. All transistors in a segment share the drain, gate, source and bulk material. Again, they can be implemented as a header, or a header and footer switch. Fig. 2*c* shows an example of a segmented symmetric header and footer switch. In this figure, there is only one segment and it contains three transistors. Here, the core is able to operate if at least a pre-determined number of segments are turned on simultaneously. One of the reasons behind such an implementation is to improve on design-for-manufacturability (DfM). Other reasons can be seen as the gradual on/off switching of the power switch to minimise the VDD/GND bounce [5], or the physical placement of the segmented power switches around the core.

Fig. 3 shows an example placement of power switches used in the ANGEL chip developed in Philips Research, Eindhoven. In the example shown, power switches are placed around a core such that the reliability and the electrical robustness (voltage and current characteristics) of the core's power grid are guaranteed. The efficient placement of power switches in a chip is itself a well-known problem [13] and will not be addressed in this paper.

4 Problem definition

Power gating functionality is enabled by a power switch using two distinct operation modes, for example (1) the switch is conducting, and (2) the switch is not conducting. From a test point-of-view, power gating functionality has to be verified to prevent failures due to manufacturing defects. For example, a short between the source and the drain of a header switch will cause the switch to be permanently on. Similarly, an open between a header switch and the core to connected to it, will cause the core to be permanently off. Therefore, it is very important to test the power gating functionality. Furthermore, if there are a number of segments in a power switch, then it is important to ensure that all segments are working correctly. This relates to the

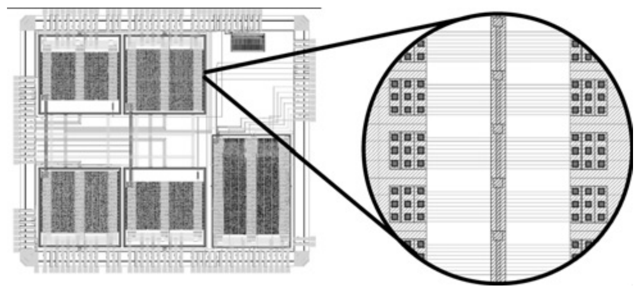

Fig. 3 *Wrapped-around placement of power switches for a core*

life-time of the switch. In this paper, we present a test methodology and DfT circuit required to allow the testing of power switches. The problem addressed in the paper can be formally defined as follows.

4.1 Problem (testing of power switches)

Instance: A power switch is given with a number of segments m, and for each segment s, the number of transistors, t_s and the segment on-resistance, R_s are given. Furthermore, the supply voltage VDD and a number k, which represents the minimum number of segments that needs to be turned on simultaneously so that the core connected to the power switch can operate functionally are given.
Objective: To determine a test circuitry and a pattern set P_m, such that all segments are tested for on/off functionality and each individual failing segment can be identified.

For a single header switch, both m and k are equal to 1. For multi-segment power switches, the number of segments k that need to be turned on simultaneously is determined at design time and should be selected such that they are sufficient to provide a value of at least $VDD_{\min}$ at V_{core}, where V_{core} is the voltage at the core's power supply pin.

4.2 Selection of number of segments

Although in the problem definition described above, we assume that k is given, it is important to know how the value of k is selected for a given power switch. The number of segments, k, is selected such that (1) the on-resistance of the k segments does not exceed the resistance of the core, and (2) V_{core} is at least $VDD_{\min}$, that is

$$\frac{1}{\sum_{s=1}^{k} \frac{1}{R_s}} \le R_{\text{core}} \qquad \text{for} \quad 1 \le k \le m$$

where R_s is the on-resistance of a segment s, R_{core} is the equivalent core resistance ($\sim 1/\alpha Cf$). A value of $VDD_{\min}$ at V_{core}, will ensure that the core can still operate functionally. The value of V_{core} can be determined as follows

$$V_{\text{core}} = \frac{R_{\text{core}} \times VDD}{R_{\text{core}} + 1/\sum_{s=1}^{k} \frac{1}{R_s}}$$

In order to have $V_{\text{core}} \ge VDD_{\min}$, the following should hold true

$$V_{\min} = \frac{R_{\text{core}} \times VDD}{R_{\text{core}} + 1/\sum\limits_{s=1}^{k} \frac{1}{R_s}}$$

Let us assume that $\sigma (0 < \sigma \le 1)$ represents the ratio between $VDD_{\min}$ and VDD, that is $\sigma = VDD_{\min}/VDD$. If all segments have the same on-resistance, that is $R_{s1} = R_{s2} = R_s$, the above-mentioned equation can be written in the following form to calculate the value of k

$$k \ge \frac{R_s \times \sigma}{R_{\text{core}} \times (1 - \sigma)}$$

Note that k should be an integer. Consider an example of a logic block with a switching circuit capacitance of 0.6 nF, a maximum operating frequency of 200 MHz, a nominal power supply (VDD) of 1.2 V and $VDD_{\min}$ as 0.6 V ($\sigma = 0.5$). Suppose that the power switch is sized such that the voltage drop across the switch is less than 1% of VDD when the block is running at its maximum performance.

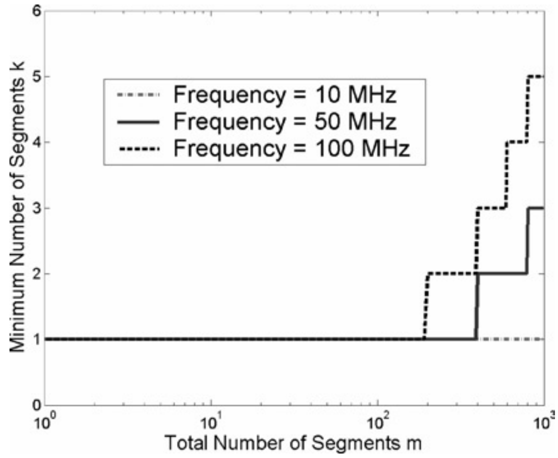

Fig. 4 *Selection of minimum number of segments*

Fig. 4 shows k as a function of m and different test clock frequencies. From the figure, one can see that the value of k does not increase linearly with m. Depending on the operating frequency, the value of k varies between 1 and 5 even for a power switch with 1000 segments. Therefore, for a large set of practical power switches, the value of k generally does not exceed five.

5 Testing of power switches

We first start with a test solution for a single-header power switch. The solution also applies to a single-footer power switch. Later, the proposed circuitry is extended for the symmetric header and footer switch and the segmented power switches.

5.1 Header switch

For a core with a header power switch shown in Fig. 5*a*, on/off functionality of the power switch can be tested by means of the additional circuitry shown in Fig. 5*b*.

The basic idea is to use a comparator to compare the logic level value of the core's power supply V_{core} with the logic value of the *stand by* signal. The comparator compares the two signals and confirms the opposite nature of the two signals. In the case of the signals having the same logic values, the comparator output can be used to indicate that something is wrong with the power switch. The comparator can be implemented as an EX-OR gate as shown in Fig. 5*b*. The comparator and the other added circuitry

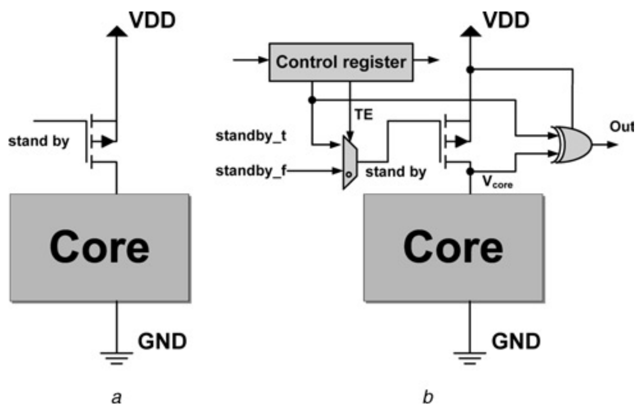

Fig. 5 *Test circuitry for a header switch*

a Header power switch
b Additional circuitry for testing on/off functionality of power switch

are powered from the undisrupted power supply *VDD* to allow operation when the controlled power supply V_{core} is turned-off. The signal *standby_f* is the functional signal to control the operation of the power switch, while the signal *standby_t* is the test signal for the same operation. The signal *standby_t* can be provided by means of a shift register, which can be programmed and controlled via the IC level IEEE 11491.1 TAP controller.

To observe the value of the comparator output (*Out*), the output can be connected to a scanable flip-flop, which can be scanned out. This also makes the proposed circuitry compliant with existing DfT approaches such as scan test for logic circuits.

5.1.1 Power gating functionality: To test the power switch with the proposed circuitry, the following two patterns are applied. In the test mode, the signal *TE* is set to 1 so that the multiplexer in front of the power switch selects the signal *standby_t*.

Pattern 1: *TE* = 1 and *standby_t* = '1'

In this case, the power switch should be turned-off and hence V_{core} should be much lower than *VDD*. Please note that the leakage should be constrained by the power switch. Ideally V_{core} should be zero. Therefore, the EX-OR output should be equal to '1' (i.e. *Out* = '1') for a correct operating power switch. If *Out* is equal to logic '0' instead of '1', this indicates that the power switch is not working correctly and there may be a short between V_{core} and *VDD*.

Pattern 2: TE = 1, and *standby_t* = '0'

In this case, the power switch should be turned-on and hence V_{core} should be equal to *VDD*. Therefore, the EX-OR output should be equal to '1' (i.e. *Out* = '1') for a correct operating power switch. If *Out* is equal to logic '0' instead of '1', this indicates that the power switch is not working correctly and that there may either be an open in the power switch or there exists a short between V_{core} and *GND*.

Therefore, by observing the value of the *Out* signal, one can check whether the switch is working correctly or not. It is important to note here that switching threshold voltage for the EX-OR gate should be VDD_{min}. As the value of VDD_{min} can be different for different types of blocks (logic, memory and so on), we assume that a library of different EX-OR cells with different threshold voltages is available. Such a library can be designed easily by changing the size of the driving-stage transistors in the EX-OR gate. Depending on the requirements of a block, a user can select the appropriate EX-OR gate and use it to test the power switch connected to the block.

5.1.2 Ordering of test patterns: The order in which the two test patterns are applied is very important for the correct testing of the power switch. If pattern 2 is applied first, node V_{core} will be charged to *VDD* and hence sufficient time must be allowed for the complete discharge of node V_{core} before applying pattern 1. Otherwise, the output of the EX-OR gate will be '0' irrespective of the power switch functionality, which can also mean a false fault behaviour.

5.1.3 Testability of the proposed structure: In the proposed test circuitry, the output of the EX-OR gate is always tested for logic level '1'. A close look at the circuitry also reveals that it is not possible to drive a logic level '0' at

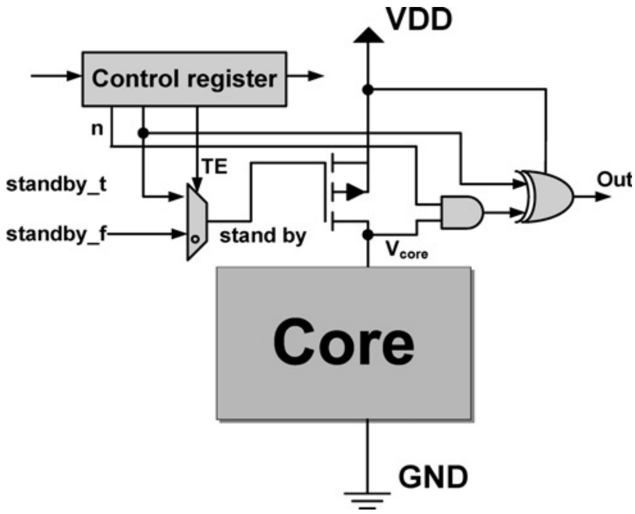

Fig. 6 *Adding a test point to the proposed circuitry*

the *Out* signal. Therefore, a stuck-at-1 fault at the *Out* signal cannot be detected. Furthermore, as the proposed test method requires *Out* to be at logic level '1', both test patterns will pass irrespective of the defects in the power switch. To circumvent such a situation, a test point can be added to the circuit so that the output of the EX-OR gate can be tested for a stuck-at-1 fault. One example implementation of such a test point is shown in Fig. 6. Here, we use a simple AND gate as a test point: however, a transparent scan flip-flop as well as other types of test points can also be used. The input *n* of the AND gate can be programmed via the control register. Therefore, by selecting signals as *TE* = 1, *n* = '0', and *standby_t* = '0', we can force a value '0' at *Out*. Now by observing the *Out* signal, we can check whether there is a stuck-at-1 fault at *Out* or not. It should be noted that all added test circuitry is powered from the undisrupted power supply *VDD*.

5.2 Header and footer switch

The above proposed circuitry and test method can be easily extended for a symmetrical power switch (header and footer switch). In this case, the proposed circuitry is added separately to both the header switch as well as to the footer switch. The resulting circuitry is shown in Fig. 7. Header

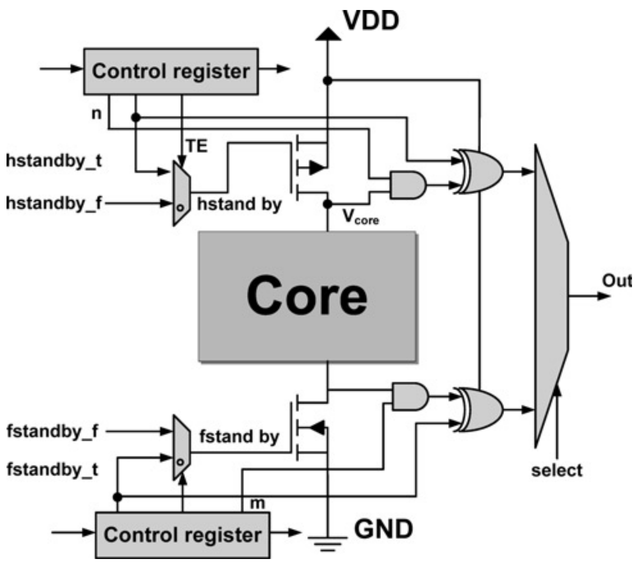

Fig. 7 *Test circuitry for a symmetrical power switch*

and footer switches are tested sequentially, for example, the header switch is tested first followed by the footer switch. The *select* signal at the output multiplexer selects the desired output node. It is important to note for clarity that, two separate control registers and two *TE* signals are shown in Fig. 7. However, a single control register can be shared between the header and the footer switch.

6 Testing and diagnosis of segmented power switch

As described earlier, one of the most popular ways of implementing power switches is via a number of transistor segments. A segment can contain one or more transistors. The number of segments m and the number of transistors t_s in each segment s are determined at design time (see Section 4). Different segments can have a different number of transistors. The above described test circuitry and method also enables the testing of segmented power switches.

Now let us consider a general case. To test and identify individual failing segments, we use the concept of a sliding window. For every test pattern, a window on w $(k \leq w \leq m)$ segments is selected. For any two consecutive patterns, the two respective windows have $w - 1$ common segments. In other words, there is an overlap q of $w - 1$ segments between two subsequent windows. The complete operation is repeated in a sliding window fashion until all segments are turned-on and off at least once and therefore tested.

Fig. 8 shows the test circuitry required for a two segment header power switch. The segments shown in the figure contain four transistors each. Please note that the positions of test bits (s1, s2, ...) in the segment control register can be optimised in order to minimise the register programming time. During the testing of segments, both *TE* and *n* are set to '1'. Let us assume $k = 1$. If we take window size $w = 1$, there cannot be any overlap between windows (i.e. $q = w - 1 = 0$). For $w = 1$, three test patterns (as shown in Table 1) are required to test the segments. The values of the fault free as well as faulty responses for these patterns are also listed in the table.

The first pattern $s1 = $ '0', $s2 = $ '1', enables the first segment, which is controlled by $s1$, while the second segment is disabled. As it is assumed that turning on one segment is sufficient to provide a value of at least $VDD_{\min}$ at $V_{\rm core}$, the *Out* signal should be '0'. Therefore, this pattern checks for the presence of a complete open in the first segment.

Similarly the second pattern $s1 = $ '1', $s2 = $ '0', enables second segments and disables the first segment. In this case also, the *Out* signal should be '0'. This pattern

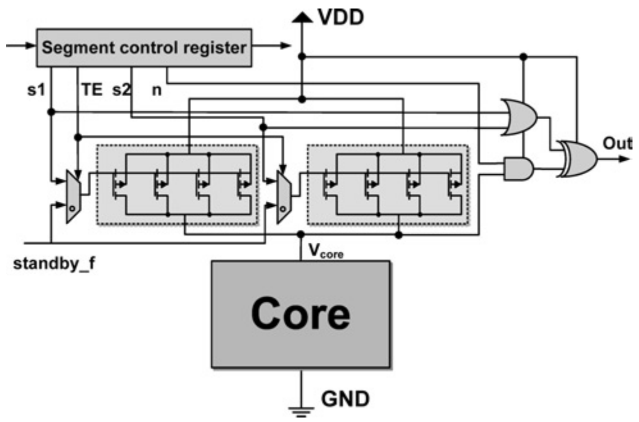

Fig. 8 *Test circuitry for segmented power switch*

Table 1: Test patterns for testing a two segment power switch

Pattern	w = 1			
	Inputs		Output	
			Out	
	s1	s2	Faulty free	Faulty
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

checks for an open in segment two. The third pattern disables both segments, and hence $V_{\rm core}$ should be ideally at ground (*Out* should be '1'). If this is not the case, then either of the two segments or both of them have a short to *VDD*. Therefore, by applying these three patterns, the two segments can be tested for on/off functionality as well as for possible manufacturing defects.

For a more generic case with $m = 5$ and $k = 3$, the sliding window concept is shown in Fig. 9. In the figure, the window size is 3, that is $w = k = 3$. For clarity, the core connected to the power switches is not shown and configurations for only a few patterns are shown. In this case, the overlap between any two consecutive windows is of two segments.

If multiple segments are tested at a time, the sliding-window concept is useful to identify in which segment the fault has occurred; however, it is not possible to pinpoint a segment's faulty transistor. The concept of sliding window is also very useful when there are a large number of segments and testing them individually might not be very economical.

It is important to note that the resistance of the core $R_{\rm core}$ should be constant during the testing of all segments, since

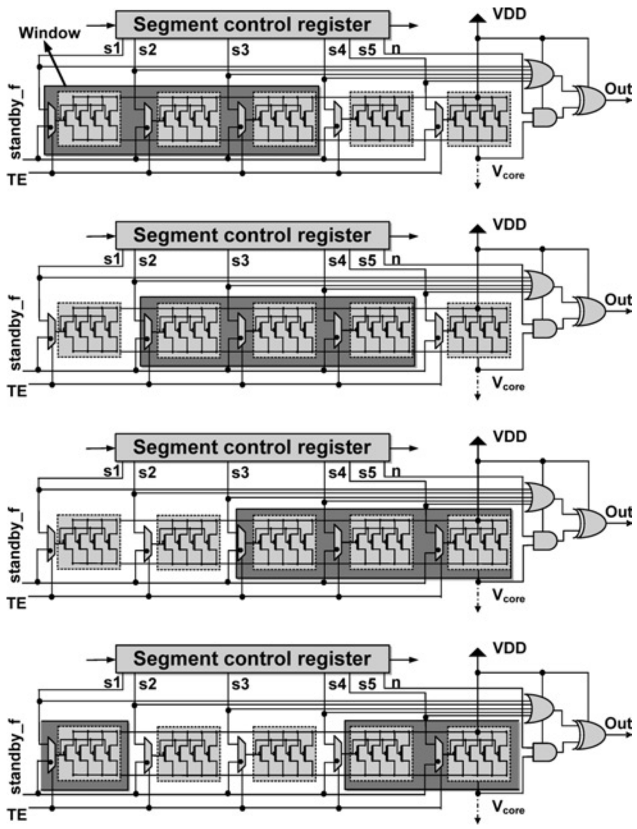

Fig. 9 *Testing of segmented switches by sliding overlapping windows*

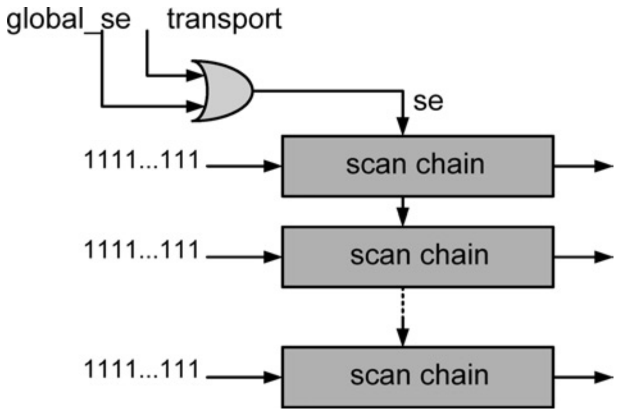

Fig. 10 *Configuring scan chains in transport mode*

the value of k is selected for a particular R_{core}. The resistance of a core can be kept constant by configuring the core in a transport mode while shifting a sequence of all 1's or all 0's through the scan chains in the core. Fig. 10 shows an example of scan chains configured in transport mode. In the figure, signal *global_se* is the top level scan enable signal, while *se* is the local scan enable signal connected to the scan chains. By using the *transport* signal, scan chains can be configured in shift mode irrespective of the state of the signal *global_se*. Here, we can safely assume that during shift cycles, states of the scan chains are not propagated through the combination logic of the core.

Therefore, if the length of the longest scan chain in the core is l_{max}, then after l_{max} clock cycles, all flip-flops in the scan chains will contain either a logic '1' or a logic '0' depending on the input sequence (in Fig. 10, a sequence of all 1's is shown). After this, the activity in the core will be just the clock activity. As the resistance of the core is directly proportional to the activity inside the core, by keeping the activity constant, the resistance is kept constant as well.

7 Test pattern generation for identification of failing segments

In the previous section, we showed how the sliding window concept can be used to test a given (m, k) segmented power switch. Just to test on/off functionality of a power switch, only two patterns, with all '1's and all '0's are sufficient. However, our objective is not just to test power switches for on/off functionality but also to identify as many individual failing segments as possible. Therefore, for maximal diagnosis of individual failing segments, the window size w should be equal to k. Based on this, the number of test patterns $|P_m|$ required for a given (m, k) power switch can be calculated as follows

$$|P_m| = m + 1 \qquad \forall k = 1 \quad \text{or} \quad k = m$$
$$|P_m| = 2m \qquad \forall 1 < k < m$$

It is important to note that the number of patterns does not depend on the overlap q between the windows for two subsequent patterns. Basically, $p1$ distinct patterns with k '0's and $(m - k)$ '1's are required to check whether there is a complete open in any of the segments, while $p2$ distinct patterns with $(k - 1)$ '0's and $(m - k + 1)$ '1's are required to check a complete short in any of the segments. As we want to test the segments using the sliding window concept, the patterns should contain uninterrupted runs of required '0's or '1's. To generate the required pattern set for a given (m, k), one needs to start with a m-bit vector with the first k bits as '0's and $(m - k)$ bits as '1's. To obtain the next vector, one needs to circularly shift the sequence of k '0' bits to the right by one position. The shift operation needs to be carried out $m - 1$ times to get $p1$ distinct test patterns. Similarly, for $p2$ patterns, one needs to start with an m-bit vector with the first $k - 1$ bits as '0's and $(m - k + 1)$ bits as '1's.

For the case with $k = 1$, $p1$ is m, while $p2$ is 1 as it corresponds to a pattern with all '1's. Similarly, for the case with $k = m$, $p1$ is 1, while $p2$ is m. Therefore, the total number of patterns for these two boundary cases is $m + 1$. For all other cases with $1 < k < m$, both $p1$ and $p2$ are m, therefore, $2m$ patterns are required to test power switches. Some examples of the generated pattern set for different values of (m, k) are given in Fig. 11.

It is important to note that based on the value of k and m, it might be impossible to pinpoint all individual failing segments. For example, in cases with $k = 1$ and $m > 1$, it is not possible to detect which of the m segments has a possible short. Similarly, for the cases with $k = m$, segments cannot be diagnosed individually for a possible open.

Multiple faulty segments: In the case of multiple faulty segments, it is still possible to identify individual faulty segments. For the cases with $1 < k \leq m/2$, segments can be identified individually for opens in the presence of maximum k faulty segments, while for the case with $m/2 < k < m$, segments can be identified individually for opens only in the presence of maximum $m - k$ fault segments. Similarly, for shorts, segments can be identified individually in the presence of maximum $k - 1$ faulty segments for the cases with $1 < k \leq m/2$, while segments can be identified individually in the presence of maximum $m - k + 1$ faulty segments for the cases with $m/2 < k < m$.

8 Conclusion

To minimise dynamic and leakage power consumption, advanced power-aware design techniques are required for modern system chips. A very effective way to minimise leakage power consumption is to turn-off the parts of a chip that are not active. For this purpose, power switches

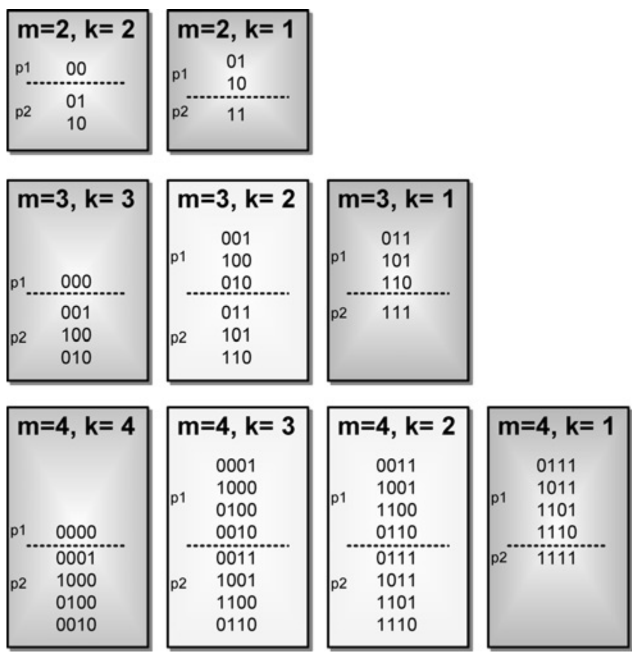

Fig. 11 *Test pattern set for a range of m and k*

are used. Power switches can be implemented in various ways such as header switch, footer switch or a segment of transistors. In this paper, we have presented a structured method to test power switches. The proposed method requires a simple circuitry and works well for all types of existing power switches. The proposed method requires a few test patterns and also enables diagnosis of faulty segments in the case of segmented power switches. As a future work, we are planning to put the proposed structure in silicon.

9 Acknowledgments

We thank our colleagues Erik Jan Marinissen and Luis Elvira Villagra for their useful comments on an earlier draft version of this paper.

10 References

1 Semiconductor Industry Association (SIA): 'International technology roadmap for semiconductors', 2005
2 Roy, K., Mukhopadhyay, S., and Mahmoodi-Meimand, H.: 'Leakage reduction mechanism and leakage reduction techniques in deep-submicrometer CMOS circuits', *Proc. IEEE*, 2003, **91**, pp. 305–327
3 Meijer, M., Pessolano, F., and Pineda de Gyvez, J.: 'Technology exploration for adaptive power and frequency scaling in 90 nm CMOS'. Proc. IEEE, ISLPED, 2004, pp. 14–19
4 Kao, J., Narendra, S., and Chrakasan, A.: 'MTCMOS hierarchical sizing based on mutual exclusive discharge patterns'. Proc. IEEE/DAC Design Automation Conference (DAC), 1998, pp. 495–500
5 Kim, S., Kosonocky, S., and Knebel, D.: 'Understanding and minimizing ground bounce during mode transition of power gating structures'. Proc. IEEE, ISLPED, 2003, pp. 22–25
6 Kosonocky, S., Immediato, M., Cottrell, P., Hook, T., Mann, R., and Brown, J.: 'Enhanced multi-threshold (MTCMOS) circuits using variable well bias'. Proc. IEEE, ISLPED, 2001, pp. 165–169
7 Zorian, Y., Marinissen, E.J., and Dey, S.: 'Testing embedded-core based system chips'. Proc. IEEE International Test Conference (ITC), 1998, pp. 130–143
8 Goel, S.K., Chiu, K., Marinissen, E.J., Nguyen, T., and Oostdijk, S.: 'Test infrastructure design for the Nexperia^{TM} home platform PNX8550 system chip'. Proc. IEEE, Design Automation and Test in Europe (DATE), 2004, pp. 108–113
9 Lackey, D., Zuchowski, P., Bednar, T., Stout, D., Gould, S., and Cohn, J.: 'Managing power and performance for system-on-chip designs using voltage islands'. Proc. IEEE Int. Conf. on Computer-Aided Design (ICCAD), 2002, pp. 195–202
10 Puri, R., Stok, L., Cohn, J., Kung, D., Pan, D., Sylvester, D., Srivastava, A., and Kulkarni, S.: 'Pushing ASIC performance in a power envelope'. Proc. IEEE/ACM Design Automation Conference (DAC), 2003, pp. 788–793
11 Puri, R., Kung, D., and Stok, L.: 'Minimizing power with flexible voltage islands'. Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), 2005, pp. 21–24
12 Power Wise Interface (PWI) Open Standard http://www.pwistandard.org
13 Kozhaya, J.N., and Bakir, L.A.: 'An electrical robust method for placing power gating switches in voltage islands'. Proc. IEEE Custom Integrated Circuits Conference (CICC), 2004, pp. 321–324