An Effective Design-for-Iddq-Testing Approach for Embedded Cores Based System-on-Chip

John Sunwoo (Author), Jonathan Harris (Reviewer)

Dept. of Electrical and Computer Engineering
Auburn University
Auburn, AL 36849-5201
phone: (334)844-1843
john@johnsunwoo.com

ABSTRACT - Papers concerning Iddq testing issues have been reviewed and summarized analytically. The most interesting topic among the Iddq testing areas is to apply testing on the Very Large Scaled Integrated (VLSI) circuit as technologies become more complex and smaller in size. Modern System-on-Chips (SoCs) are mostly in deep submicron technology and it presents a major challenge in the implementation of Iddq testing. [1] The problems are increased leakage current resulting from increased number of gates as well as result of increased sub-threshold leakage on the individual transistors [1]. There are two main topics that are summarized and analyzed; a design-for-test concept to address the issue of high leakage due to the large size of SoC design and the list of suggestions on some design rules that are necessary to make SoC design suitable for Iddq testing [1]. Mostly, in this paper, one of the design methodologies is presented that implements the Iddq testing by controlling power-supply of the individual cores through JTAG boundary scan allowing Iddq testing on one core at a time. The significance of this approach is that it does not require any dedicated control pin and it has a negligible area overhead profile [1].[1]

INTRODUCTION

Motivation of most research and design approach is to solve the issue of increased leakage current due to enormous size of the modern SoC design. As the technology advanced, VLSI CAD technologies developed into the deep submicron level. Therefore, testing different issues arose. As an example, a particle of size 0.5um causing extra metal was relatively unimportant when metal pitch was 2.0um, the same defect in a technology with 0.25um metal pitch most likely will cause a catastrophic short between metal lines which is commonly referred to as a bridging fault.

Most test engineers consider Iddq testing an integral part of the overall IC testing. This is because it can detect and identify the root cause of a fault without taking relatively expensive cost. All the factors in the test cost, such as additional design effort and area, test generation effort, simulation time and test application time, are relatively small when compared to testing in voltage environment [1]. For example, it will double the test cost to increase the stuck-at fault coverage from 80% to 90% in voltage environment while adding a small Iddq test set will provide equivalent or better benefits with in relatively inexpensive cost [1]. Furthermore, effectiveness of using Iddq test is shown by the fault coverage that can quickly be raised, approaching 100% by adding a small Iddq test set to the functional and stuck-at fault test set [1].

The major advantage of Iddq testing is because of its clear and relatively simple observability, thus, the test generation effort is very low compared to logic testing. Unlike logical stuck-at-fault testing, Iddq testing does not require the fault propagation effort. It only requires activating the fault site since the fault effect can be observed through the power supply. This leads to a very high detectability capability of each Iddq vector.

While Iddq testing is a very desirable test method, its implementation on SoC design is considerably challenging which is due to the large size of the SoC design. The problem of Iddq testing on an SoC device is that it holds overly large net leakage current (Ioff) of all transistors that makes it hard to distinguish between faulty and fault-free chips [1]. Another way to look at this problem is through the size of the today's embedded cores which are in the range of 20K-to-200K gates [1]. This leads to the points that when 4-5 cores on-chip are powered by a power supply, then the power supply loading becomes 100K-to-1M gates. The combined leakage of these 4-5 cores makes it almost impossible to perform Iddq testing.

2. PROBLEMS ON SOC IDDQ TEST

An analysis of Iddq testing is based on the estimation of fault-free current in the circuit. Defining a limit of current is called Iddq threshold which is a parameter that determines if the circuit is faulty or not by observing if the current goes above the threshold or not. Practical Iddq thresholds, in industry range from 10uA to 100uA, but the threshold is very design dependent. Typically, for digital circuits in 0.5um, 0.35um, and 0.25um technologies, a number close to

---

[1] Term Paper on Iddq Testing [VLSI-TESTING, Spring/2004]

1uA is considered as fault-free [1]. The Gaussian distribution is used to analyze the measured current over large number of ICs [5]. As shown in Figure 1, using statistical variations, ICs up to (mean+3σ) are considered fault-free. Anything above the value considered to be over the threshold is faulty [5].

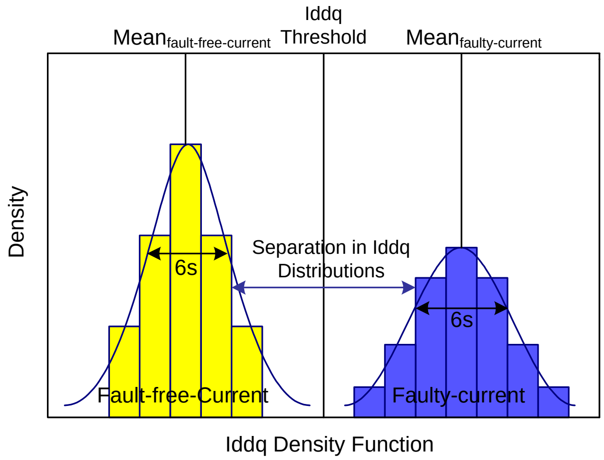

Figure 1. Representation of fault-free and faulty Iddq density functions.

To characterize the difference between the faulty and fault-free current data, a statistical method called the Separation is used and its equation is shown in [1] and is as follows:

$$\mathrm{Iddq}_{\mathrm{fault}} - \mathrm{Iddq}_{\mathrm{fault\text{-}free}} = (\mathrm{Average\ Iddq}_{\mathrm{fault}} - 3\sigma_{\mathrm{Iddqfault}}) - (\mathrm{Average\ Iddq}_{\mathrm{fault\text{-}free}} + 3\sigma_{\mathrm{Iddqfault\text{-}free}})$$

The density functions of faulty and fault-free current will be used to determine the good and the bad IC, however, as the technology advances, the clear distinction is hard to make because of the two main factors; the increased sub-threshold leakage (Ioff) per transistor and increased number of gates in the circuit. These two factors will flatten the distribution of the fault-free current data and cause it to be located closely to the distribution of the faulty current data. When the overlap occurs between these two distribution profiles, it is impossible to determine if the circuit passes the Iddq test or not. The definition of sub-threshold leakage (Ioff) is a current that represents steady state leakage of a transistor. The net Iddq in a circuit can be found by adding all the leakage as follows [1]:

$$\mathrm{Iddq} = (\mathrm{Normalized\ number\ of\ nMOS} * \mathrm{n\text{-}Ioff/um}) + (\mathrm{Normalized\ number\ of\ pMOS} * \mathrm{p\text{-}Ioff/um})$$

The equation above is the example that shows the relationship between Iddq and Ioff, number of transistors. Table 1 indicates one of the relationships. It shows that Ioff increased 4~5 orders of magnitude when changing from 0.8um to 0.18um technology.

Tech (um)	Vdd (V)	Tox (Å)	Vt (V)	Ioff (pA/um)	
				NMOS	PMOS
0.8	5	150-100	0.8-0.7	.01-.05	.005-.02
0.6	5-3.3	100-80	.75-.65	.05-0.5	.01-0.2
0.5	5-2.5	90-70	0.7-0.6	.1-2	.1-1
0.35	3.3-2.5	X0-60	.65-.55	.5-10	.1-10
0.25	3.3-1.8	70-50	0.6-0.5	6.0-60	0.5-24
0.18	2.5-1.8	55-35	.55-.45	40-600	20-300

Table 1. Characteristics of various technologies and range of Ioff data.

In summary, increased profiles in Ioff and the number of gates in the circuit, the distribution of the fault-free current flattens due to increased σ and it is more likely to reach the threshold limit. This will be a big obstacle in distinguishing whether the circuit is fault-free or not by using its distribution profile. The main topic discussed is about a design-for-test approach that attempts to solve this problem. Unlike the traditional methods such as partitioning the circuit physically, the new approach provides a virtual partition by controlling the power-supply of each individual core in SoC [1]. The advantages of not using the physical level partitioning are the avoidance of area overhead and performance penalty [6].

3. IDDQ TESTING FRIENDLY DESIGN

The concept of Design-for-Iddq-testing is presented through maneuvering the control over power supply so that it allows Iddq testing on one core at a time. This eliminates the problem due to large number of gates in the design. An approach such as presented in [3, 7] proposed using a global control signal to switch-off static current dissipating logic. This requires a special buffer and a dedicated pin to control this global signal during the Iddq testing as well as in the regular modes of operation. As illustrated in Figure 2, the specially assigned pin will control the power states of embedded core inside of the SoC device. It is always at a logic value '1', keeping the pull-up/pull-down transistors ON at the signal pins. When Iddq testing phase starts, the signal pin will apply a logic value '0' to unnecessary partitions or embedded cores which switches-off all pull-up/pull-down transistors on the signal pins. Such an approach has negligible overhead, however, requiring one dedicated pin to control the global signal is the limitation on this approach.

The study here is to overcome the requirement of dedicated pin and there is a solution that solves the problem by utilizing JTAG Boundary Scan proposed in [1]. JTAG Boundary Scan can give access and control over the power-supply. Thus, there would be a flip-flop that is used to control the logic of global power-down control signal. For instance, the flip-flop holds logic value '1' during the normal modes of operation; while in the Iddq test mode, the flip-flop holds logic value '0' [1]. After the test is done, the flip-flop value changes to '1' and the circuit comes back to the normal operating mode. Note that with the power-down control signal being '0', the TAP controller is kept in the Run-Test/Idle state for the duration of Iddq testing [7].

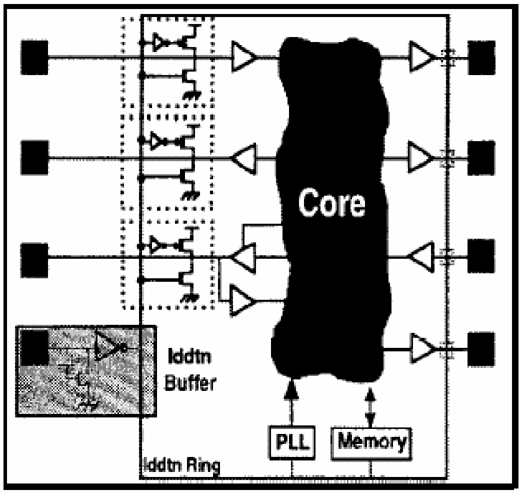

Figure 2. ASIC design that allows to switch-off static current dissipating logic during Iddq testing [3]

The concept behind this approach is illustrated in Figure 3, which has 4 total cores including a large embedded memory core. Using instructions that can be utilized through JTAG Scan, any core can be powered up selectively without any static current dissipating logic.

With the instruction that processes the power down signals, there is a decoding mechanism that gives the control to the target core to be tested. It keeps one power-down signal at '1' while all other power-down signals are set to '0'. The power-down control signals at '0' should cut-off the power supply of the respective blocks.

Finally, the other design rules are mentioned and suggested that help to avoid unwanted high Idd states in the SoC devices. They are summarized as follows [4] [8]:

1. Full circuit initialization is a fundamental requirement for Iddq testing. It means that all flip-flops (registers) must be placed in a known state, such as flip-flops within a core when core is tested or glue logic when glue logic is tested.

2. All static current dissipating logic within a core must be switched-off using power down control signals

3. All circuits including individual cores and glue logic must be stable at the strobe point.

4. All inputs and bi-directional pins of all individual cores must be at a deterministic 0 or 1.

5. All pull-up and pull-down transistors at the chip I/Os must be switched off.

6. There should be no internal bus conflict or floating nodes during Iddq testing.

7. When the primitive nets are driven by multiple drivers, nets should not be either potentially floating or have a conflict.

8. All nets within a core and glue logic should be checked that there is no weak value fans-in to a gate during Iddq measurement.

9. Special circuit structures should be avoided as much as possible. The examples of such structures are gate and drain/source of a transistor be driven by the same transistor group; feed back and control loops within one transistor group.

10. When design, use the components with the low power profile.

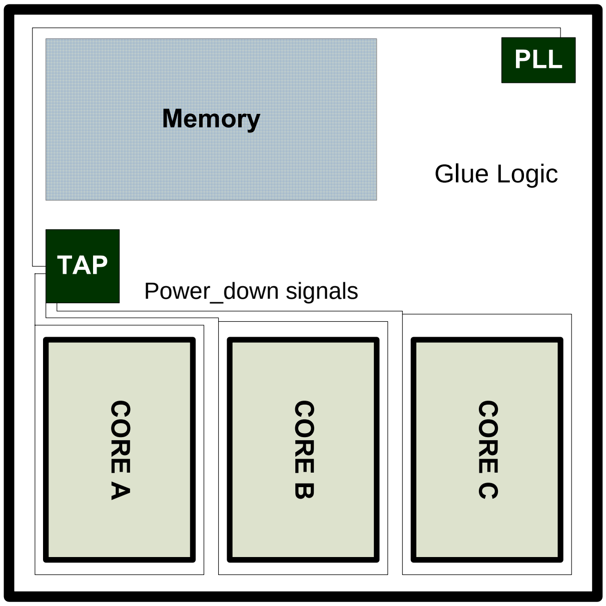

Figure 3. Implementation of power-down control signals to perform Iddq testing on embedded cores based system-on-chip.

4. CONCLUSIONS

A new implementation of a previous concept to obtain a global control at the chip level to facilitate Iddq testing has been presented. A methodology of SoC Iddq test has been summarized as well. Finally, a number of design rules that are necessary in SoC design to make it suitable for Iddq testing is suggested. The most impressive work here is to utilize JTAG boundary scan to generate chip-level power-down control signals. And the power-down control enables the selective power-off operation of the chip which allows Iddq testing on one core at a time. The importance of this work is the fact that they are using the JTAG which is well known for Design for Testability (DFT) solution for the future. Another remarkable thing is the fact that the hardware overhead in this design approach is negligible and does not require any dedicated pins to control the power-down signals. The future research on this field should be about combining the Iddq testing along with the voltage based test such as the functional and stuck-at fault and then study how to obtain the optimized/compact test set to test the IC effectively with high fault coverage level.

REFERENCES

[1] R. Rajsuman, “Design-for-Iddq-testing for embedded cores based system-on-a-chip,” IEEE Int. Workshop on Iddq Testing, pp. 69-73, 1998.

[2] A. Keshavarzi, K. Roy and C.F. Hawkins, “Intrinsic leakage in low power deep submicron CMOS ICs”, Int. Test Conference, pp. 146-155, 1997.

[3] F. Zarrinfar and R. Rajsuman, “Automated Iddq testing from CAD to manufacturing”, IEEE Int. Workshop on Iddq Testing, 1995.

[4] R. Rajsuman, “Iddq testing for CMOS VLSI”, Artech House Inc., 1995.

[5] Y. K. Malaiya, A. P. Jayasumana, Q. Tong and S. M. Menon, “Enhancement of resolution in supply current based testing for large ICs”, IEEE VLSI Test Symp., pp. 291-296, 1991.

[6] S.M. Menon, Y.K. Malaiya, A.P. Jayasumana and Q. Tong, “The effect of built-in current sensors (BICS) on operational and test performance”, Int. Conference on VLSI Design, pp. 187-190, 1994.

[7] M. Colwell, R. Rajsuman, Z. Sarkari and R. Abrishami, US patent No. 5,644,251, July I, 1997; and US patent No. 5,670,890, Sep. 23, 1997.

[8] “Iddq testing submicron system-on-a-chip”, Semiconductor International, pp. 54-56, Jan. 1996.