ISSN 23499842(Online), Volume 1, Special Issue 2(ICITET 15), March 2015
International Journal of Innovative Trends and Emerging Technologies

AN EFFICIENT APPROACH FOR REDUCED POWER CONSUMPTION AND DIAGNOSIS OF FAULTY POWER SWITCHES USING A RANDOM TEST PATTERN GENERATION

S. SATHIYAPRIYA[1], S.LAVANYA[2]
PG Student, VLSI Design, Department of ECE, Surya Group Of Institutions, India[1].
Assistant Professor, Department of ECE, Surya Group of Institutions, India[2]

Abstract*:*
The proposed diagonsis method utilizes the recently proposed RTPG for efficient testing of power switches.Each power switches can be tested to diagnosis faulty power switches and power consumption is reduced. The role of random testing is to detect whether something went wrong and the role of diagnosis is to determine exactly what went wrong. Framed testing is used to produce time sequence for bit generation.

I. INTRODUCTION

An externally switched power supply is very basic form of power gating to achieve long term leakage power reduction. To shutoff the block for small interval of time internal power gating is suitable. CMOS switches that provide power to the circuitry are controlled by power gating controllers. Output of the power gated block discharge slowly. Hence output voltage levels spend more time in threshold voltage level. This can lead to larger short circuit current. Power gating uses low-leakage PMOS transistors as header switches to shut off power supplies to parts of a design in standby or sleep mode. NMOS footer switches can also be used as sleep transistors. Inserting the sleep transistors splits the chip's power network into a permanent power network connected to the power supply and a virtual power network that drives the cells and can be turned off. The quality of this complex power network is critical to the success of a power-gating design. Two of the most critical parameters are the IR-drop and the penalties in silicon area and routing resources. Power gating can be implemented using cell- or cluster-based (or fine grain) approaches or a distributed coarse-grained approach. Power gating implementation has additional considerations than the normal timing closure implementation. The following parameters need to be considered and their values carefully chosen for a successful implementation of this methodology.

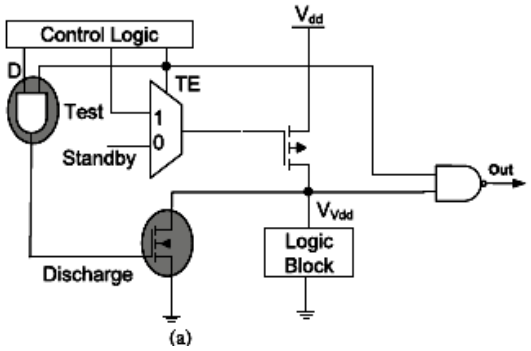


Fig 1 (a) Fine-grain design

The power gate size must be selected to handle the amount of switching current at any given time. The gate must be bigger such that there is no measurable voltage (IR) drop due to the gate. Generally we use 3X the switching capacitance for the gate size as a rule of thumb. Designers can also choose between header (P-MOS) or footer (N-MOS) gate. Usually footer gates tend to be smaller in area for the same switching current. Dynamic power analysis tools can accurately measure the switching current and also predict the size for the power gate**.**

II. RELATED WORK AND CONTRIBUTIONS

To achieve higher density and performance and lower power consumption, CMOS devices have been scaled. High leakage current in deep-submicrometer regimes is becoming a significant contributor to power dissipation of CMOS circuits. The goal is to optimize the channel profile to minimize the off-state leakage while maximizing the linear and saturated drive currents. Supersteep retrograde wells and halo implants have been used as a means to scale the channel length and increase the transistor drive current without causing an increase in the off-state leakage current. It explores different circuit techniques to reduce the leakage power consumption. 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. A simple yet effective design-for-test (DfT) circuitry together with a method to test power switches was presented. It also enables the identification of individual failing segments in the case of segmented power switches.Power gating is a low-power design technique to reduced leakage power. Power switches are used as part of power-gating technique to reduce leakage power of a design. It takes long discharge time when power

switches are turned-off, when testing power switches using available DFT solutions. The detailed analysis of available power gating DFT solutions and a simple and effective DFT solution (together with test vectors) for testing power switches to reduce the discharge time when they are turned-off is presented.
A multi-Vdd design has a set of discrete supply voltage settings it can switch between depending on the current workload and power saving mode. The testing for resistive bridging faults in such designs requires more than one voltage setting for 100% fault coverage. An effective gate sizing technique for reducing test cost of multi-Vdd designs with bridge defects was presented.

III PROPOSED OF DIAGNOSIS ALGORITHM

i) Fault Random Number
a) Objectives Of Proposed System
To propose a diagnosis method to identify the location and number of faulty power switches in a design. By using a random testing is more efficient than the coarse grain design style.
b) Techniques Used In This Method
Random testing is a testing technique where programs are tested by generating random, independent inputs. Results of the output are compared against software specifications to verify that the test output is pass or fail. In case of absence of specifications the exceptions of the language are used which means if an exception arises during test execution then it means there is a fault in the program.Types Of Random Testing With Respect To The Input Random input sequence generation (i.e. a sequence of method calls).Random sequence of data inputs (sometimes called stochastic testing) - f.ex. a random sequence of method calls.Random data selection from existing database
C) Guided Vs. Unguided
undirected random test generation - with no heuristics to guide its search.directed random test generation - f.ex. "feedback-directed random test generation" or "adaptive random testing".
d) Random Number:
Random sequences typically exhibit statistical randomness while being generated by an entirely deterministic causal process. Such a process is easier to produce than a genuinely random one, and has the benefit that it can be used again and again to produce exactly the same numbers - useful for testing and fixing software. A Random variable is a variable which is created by a deterministic procedure (often a computer program or subroutine) which (generally) takes random bits as input. The random string will typically be longer than the original random string, but less random .This can be useful for randomized algorithms. Random number generators are widely used in such applications as computer modeling (e.g., Markov chains), statistics, experimental design, etc. A Random number generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values, called the PRNG's seed (which may include truly random values). Although sequences that are closer to truly random can be generated using hardware random number generators, pseudorandom number generators are important in practice for their speed in number generation and their reproducibility.

A PRNG can be started from an arbitrary initial state using a seed state. It will always produce the same sequence when initialized with that state. The period of a PRNG is defined thus: the maximum, over all starting states, of the length of the repetition-free prefix of the sequence. The period is bounded by the number of the states, usually measured in bits. However, since the length of the period potentially doubles with each bit of "state" added, it is easy to build PRNGs with periods long enough for many practical applications.

*e) LFSR:*A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven by the XOR of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences.

f) Framed testing: This testing is performed by generating a time sequence in which for that particular time the random tests are applied and verified. Likewise different time pulses are generated and applied for tests and the outputs are analysed for variations according to the framed tests given. Framed tests are nothing but the tests which are framed for a specified time period. These tests are repeated for a specific time and the random tests are applied for this specified period.

Again this is repeating and the output switches acts according to that.

ii) Fault Modeling And Fault DaignosisSoft errors in VLSI: A soft error is a type of error where a signal or datum is wrong. Errors may be caused by a defect, usually understood either to be a mistake in design or construction, or a broken component. A soft error is also a signal or datum which is wrong, but is not assumed to imply such a mistake or breakage. After observing a soft error, there is no implication that the system is any less reliable than before. In the spacecraft industry this kind of error is called a single-event upset.

a) Chip-level soft error:These errors occur when the radioactive atoms in the chip's material decay and release alpha particles into the chip. Because an alpha particle contains a positive charge and kinetic energy, the particle can hit a memory cell and cause the cell to change state to a different value. The atomic reaction is so tiny that it does not damage the actual structure of the chip. If detected, a soft error may be corrected by rewriting correct data in place of erroneous data. Highly reliable systems use error correction to correct soft errors on the fly. However, in many systems, it may be impossible to determine the correct data, or even to discover that an error is present at all. Soft errors involve changes to data - the electrons in a storage circuit, for example -but not changes to the physical circuit itself, the atoms. If the data is rewritten, the circuit will work perfectly again. Soft errors can occur on transmission lines, in digital logic, analog circuits, magnetic storage, and elsewhere, but are most commonly known in semiconductor storage.

b) Detecting soft errors: There has been work addressing soft errors in processor and memory resources using both hardware and software techniques. Several research efforts addressed soft errors by proposing error detection and recovery via hardware-based redundant multi-threading. These approaches used special hardware to replicate an application execution to identify errors in the output, which increased hardware design complexity and cost including high performance overhead.

c) Correcting soft errors: Typically, a semiconductor memory design might use forward error correction, incorporating redundant data into each word to create an error correcting code. Alternatively, roll-back error correction can be used, detecting the soft error with an error-detecting code such as parity, and rewriting correct data from another source. This technique is often used for write-through cache memories. Soft errors in logic circuits are sometimes detected and corrected using the techniques of fault tolerant design. These often include the use of redundant circuitry or computation of data, and typically come at the cost of circuit area, decreased performance, and/or higher power consumption. The concept of triple modular redundancy (TMR) can be employed to ensure very high soft-error reliability in logic circuits. In this technique, three identical copies of a circuit compute on the same data in parallel and outputs are fed into majority voting logic, returning the value that occurred in at least two of three cases. In this way, the failure of one circuit due to soft error is discarded assuming the other two circuits operated correctly. In practice, however, few designers can afford the greater than 200% circuit area and power overhead required, so it is usually only selectively applied. Another common concept to correct soft errors in logic circuits is temporal (or time) redundancy, in which one circuit operates on the same data multiple times and compares subsequent evaluations for consistency. This approach, however, often incurs performance overhead, area overhead (if copies of latches are used to store data), and power overhead, though is considerably more area-efficient than modular redundancy.

iii) Soft errors in combinational logic :The three natural masking effects in combinational logic that determine whether a single event upset (SEU) will propagate to become a soft error are electrical masking, logical masking, and temporal (or timing-window) masking. An SEU is logically masked if its propagation is blocked from reaching an output latch because off-path gate inputs prevent a logical transition of that gate's output. An SEU is electrically masked if the signal is attenuated by the electrical properties of gates on its propagation path such that the resulting pulse is of insufficient magnitude to be reliably latched. An SEU is temporally masked if the erroneous pulse reaches an output latch, but it does not occur close enough to when the latch is actually triggered to hold. If all three masking effects fail to occur, the propagated pulse becomes latched and the output of the logic circuit will be an erroneous value. In the context of circuit operation, this erroneous output value may be considered a soft error event.

iv) Designing Of Testing Module Description

Module 1: Design and analysis of power switch module

In this module, we are designing the power switch which is the basic module for the project. This power switch is nothing but a cell which acts like a switch. This power switch is used for reducing leakage power consumption. When needed the switch will be in “ON” condition,

otherwise it will be in "OFF" condition. Thus the power switch is designed and the faults are detected by applying testing.

Module 2: Design and analysis of testing module

In this module, we are designing the test patterns which will be applied to the power switch circuit. These test patterns generated will be in random order and applied randomly to the circuit. By doing so, the faults in the power switches are exactly identified. The framed testing is also performed in which the sequences of the test patterns are framed. Only framed set of patterns are given to test the power switch circuit.

Module 3: Design and analysis of fault detection module

In this module, the faults are identified in the circuit. The test patterns are applied randomly and framed order and this outputs the fault. The fault is considered here as an open circuit and short circuit fault in the power switches. The switch is open instead of short circuit, thereby producing undesired error in the output. So these are all identified here.

Module 4: Design and analysis of integration module In this module, we are integrating all the sub modules and analyzing the output using Modelsim Simulator.

v. Block Diagram Of Proposed System

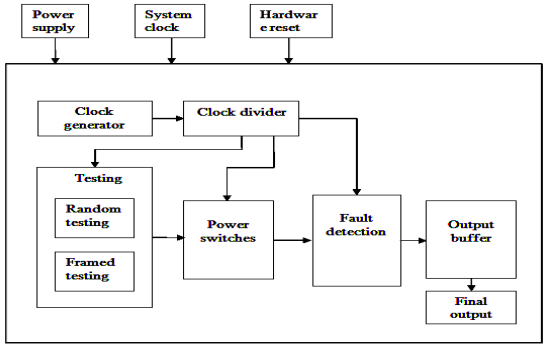


Fig 1. Block diagram of Fault modeling

From the clock generator, the input clock pulse gets generated. These clock pulses are then divided using a clock divider for uncorrelation of data. The input is given and the testing process for the power switches is applied and performed in the testing block. The error is detected in the fault detection block. Thus the output is taken by applying these test patterns to the power switches. The power supply is the input supply given to the hardware. The system clock is the basic clock pulse given to the hardware and the hardware reset is the initial condition for resetting of the hardware.

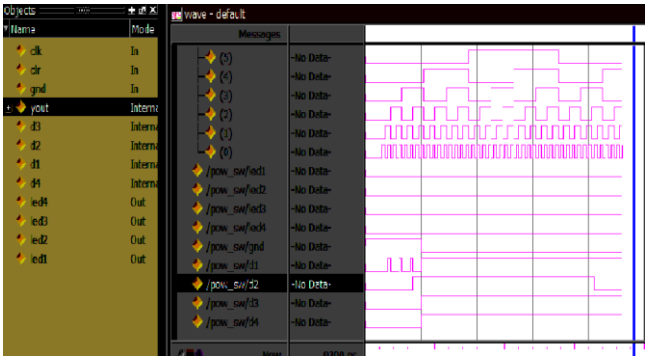
TABLE I

Comparison table

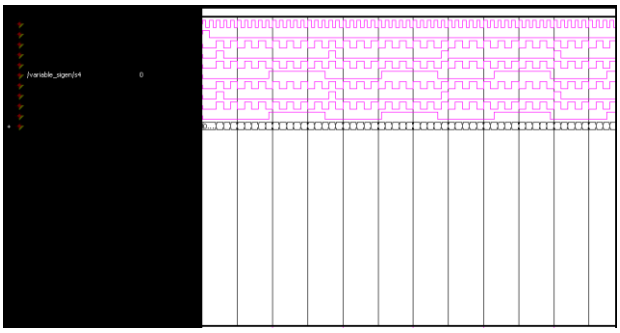
S.NO	Parameters	Existing system	Proposed system
1	Process variation	4.8v	1.2v
2	Logic elements	7segments(35elements)	43 logic elements

IV SIMULATION RESULT & VERIFICATIONS

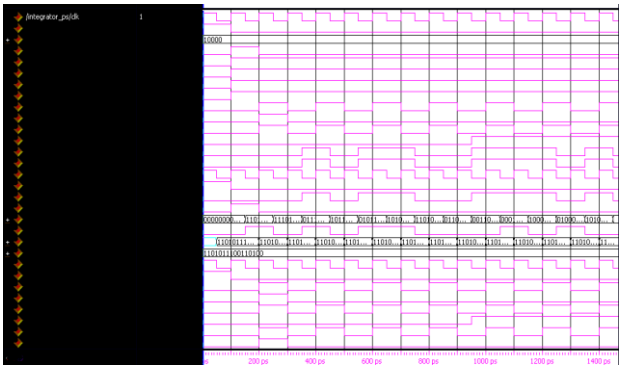
a) Simulated Output for module 1



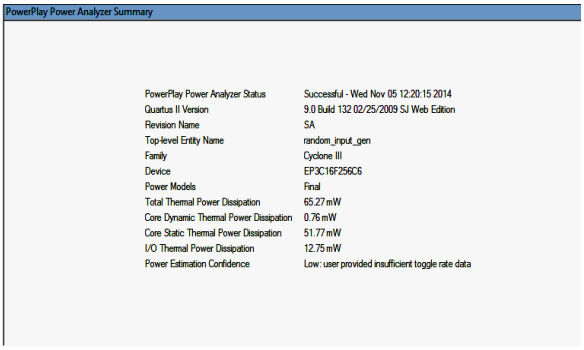
b) Simulated Output for module 2



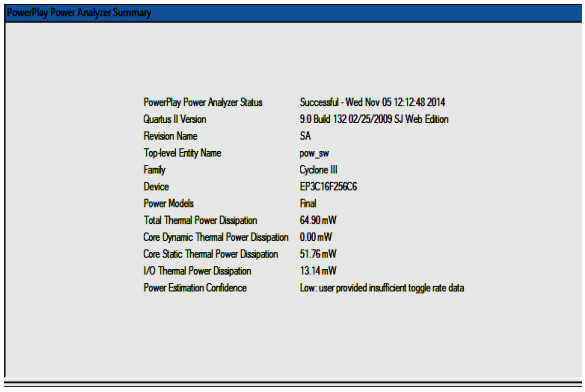
c) Simulated Output for module 3



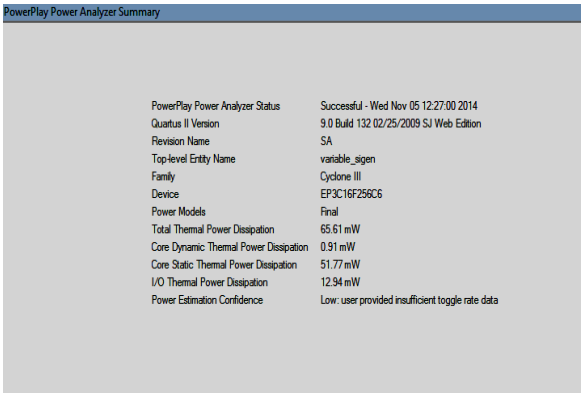
b) Power analyzer module 1



c) *Power analyzer module 2*



d) *Power analyzer module 3*



V. CONCLUSION

Thus in this, we had designed the power switch module which works according to the conditions. The random testing which was designed produces random numbers are applied to the power switch and framed testing which produces the particular time sequence are also designed. All these are applied to the power switch module and its outputs are checked. Thus all these are designed and verified successfully using Modelsim Simulator.

REFERENCES

[1] *M. Kassab and M. Tehranipoor, "Test of power management structures," in* Power-Aware Testing and Test Strategies for Low Power Devices, *P. Girard, N. Nicolici, and X. Wen, Eds. Springer, Nov. 2009, ch. 10, pp. 295–322*

[2] *M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi,* Low Power Methodology Manual for System-on-Chip Design. *Springer, 2007*

[3] *S. Kumar Goel, M. Meijer, and J. de Gyvez, "Testing and diagnosis of power switches in SOCs," in* ETS, *May 2006.*

[4] *K. Shi, Z. Lin, and Y.-M. Jiang, "A power network synthesis method for industrial power gating designs," in* ISQED, *Mar. 2007.*