

Scan Based Methodology for Reliable State Retention Power Gating Designs

Sheng Yang¹, Bashir M. Al-Hashimi¹, David Flynn² and Saqib Khursheed¹

¹School of ECS, University of Southampton, UK {sy08r, bmah, ssk06r}@ecs.soton.ac.uk

²ARM, Cambridge, UK (david.flynn@arm.com)

Abstract—Power gating is an effective technique for reducing leakage power which involves powering off idle circuits through power switches, but those power-gated circuits which need to retain their states store their data in state retention registers. When power-gated circuits are switched from sleep to active mode, sudden rush of current has the potential of corrupting the stored data in the state retention registers which could be a reliability problem. This paper presents a methodology for improving the reliability of power-gated designs by protecting the integrity of state retention registers through state monitoring and correction. This is achieved by scan chain data encoding and decoding. The methodology is compatible with EDA tools design and power gating control flows. A detailed analysis of the proposed methodology's capability in detecting and correcting errors is given including the area overhead and energy consumption of the protection circuitry. The methodology is validate using FPGA and show that it is possible to correct all single errors with Hamming code and detect all multiple errors with CRC-16 code. To the best of our knowledge this is the first study in the area of reliable power gating designs through state monitoring and correction.

I. INTRODUCTION

Transistor and voltage scaling have been the driving force behind the growth of mobile electronic industry. An undesirable side-effect however is the significant increases in leakage power with technology scaling [1]. Numerous approaches have been proposed to minimize leakage power, such as high-k metal gate [2], dual-threshold standard cells [3], drowsy logic [4] and power gating [5]. Power gating is effective in reducing leakage power in sleep mode. The power gated circuits are connected to power supply through high-Vt MOS power transistors. During sleep mode the power transistors are turned off, the leakage of the power gated circuit is limited by the power transistors, for example a reduction of 95% leakage power was reported for ARM926EJ [6]. In sleep mode when the power supply is switched off, the power gated circuit's states are lost. A retention circuit called a balloon circuit was proposed in [5] to preserve the states of power gated circuits during sleep mode. This type of memory is used by foundries to include state retention enabled flip-flops in their standard cells libraries. Fig. 1 shows a state retention enabled flip-flop, where the master flip-flop is connected to Vdd through a power transistor and the slave retention latch is always powered on. The master flip-flop consists of low-Vt transistors for fast switching during active mode, whereas the slave retention latch consists of high-Vt transistors for low leakage during sleep mode. In addition to the data and clock inputs, the state

retention flip-flop has a control signal, RETAIN. Whenever the power gated circuit is switched to sleep mode, RETAIN is set to '1' to transfer data from the master flip-flop to the slave retention latch, and before the power gated circuit is switched to active mode, RETAIN is set to '0' to restore data back to the master flip-flop.

The power supply rails of an integrated circuit are not perfect, wires have resistance and between the wires there is capacitance and inductance. In sleep mode, when the power transistors of the power gated design are off, the internal capacitance of the power gated circuit is discharged to ground and its leakage currents are determined by the power transistors. When the power gated circuit is reactivated, the power transistors are turned on, there is a rush current to charge up its internal capacitance. This sudden change of current induces a voltage across the wires' inductance, which can be modeled as step response of an RLC circuit [7]. The voltage fluctuation at the power supply rails may corrupt the state retention latches connected to it, which lead to potential reliability problem.

Some works have been reported to increase the reliability of power gating design in the presence of rush current and supply voltage fluctuation. In [7] the author proposed to turn on power transistors slowly by either controlling the gate-to-source voltage of the power transistors or turn-on a portion of the power transistors at one time. In [8] it was proposed to use pump capacitors to slowly turn on the power transistors and use a voltage monitor circuit to detect the end of the activation process. Although the general theme of the work in [7, 8] and this paper appear similar - to protect the state integrity of power gated circuits, there is fundamental difference. In this work the state integrity of power gated design is protected using state monitoring and recovery achieved by scan chain encoding and decoding. In [7, 8] the impact of supply voltage fluctuation on the state integrity is reduced through rush

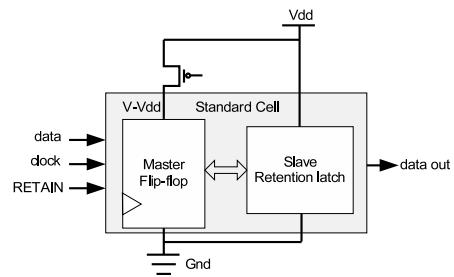


Fig. 1. State Retention Flip-Flop

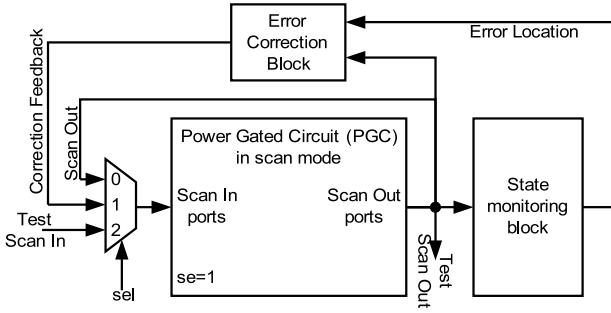


Fig. 2. State monitoring and recovery of state retention power gated design through scan chain encoding and decoding

current reduction circuit design techniques as outlined above.

In this paper a methodology is introduced which protects the state integrity of power gated design in deep sleep mode through scan chain encoding and decoding. Using this methodology, various error detection and correction codes can be applied to monitor the state of power gated design and correct any corrupted states if necessary. Rush current reduction methods [7, 8] are effective but if the state of a power gated design is corrupted they can not correct them. Through the case presented here, it is believed that an approach grounded in state monitoring and corrections, is the right step towards reliable state retention power gating designs.

The rest of the paper is organized as follows. Section II describes the proposed methodology for protecting state integrity of power gated design through state monitoring realized by scan chain encoding and decoding. Section III shows how to reuse already available scan chains for improving in-field reliability of the device without affecting manufacturing test. Section IV describes the functional verification of the proposed methodology using FPGA. Section V shows the detection and correction cost trade-offs of the proposed methodology when implementing two type of detection and correction codes. Section VI concludes the paper.

II. PROPOSED METHODOLOGY

Error detection and correction coding has been used extensively to improve the reliability of memory circuits. This is achieved by generating parity bits when writing the data into memory and checking the data against saved parity bits when reading data out of memory. However in a normal chip layout, flip-flops are not as structured as memory blocks, instead they are scattered physically. These flip-flops do not have a unified input and output channel which is required to access their data and generate parity bits like memories. Scan chains which connect flip-flops into long shift registers for performing manufacturing test can provide the channel for parity bits generation. Scan chains insertion is normally automated by EDA tools, and involves replacing the system flip-flops with scan enabled flip-flops and create scan-in, scan-out ports and a scan enable signal without affecting the functionality of the original design. Only when the scan enable signal is active the flip-flops are reconnected in a daisy chain, and the scan-in and scan-out ports are the input and output of these chains.

In this work the scan chains of a power gated design are exploited for the purpose of protecting its state integrity. This is achieved by monitoring the state through scan chain encoding and decoding. The methodology has two stages detection and correction for improving the reliability of power gated design in the presence of rush current and voltage fluctuation consists of three main parts: an architecture which monitors the states of the power gated design and recovers its states if errors are detected; a design flow for reliable power gated design; the state monitoring control that is integrated into power gating control.

A. Architecture

The architecture is shown in Fig. 2 which consists of three blocks: the power gated circuit (PGC) to be protected, the state monitoring block and the error correction block. The state monitoring block encodes the PGC states before the PGC is powered off, where the control signal 'sel' is '0' and scan enable signal 'se' is '1'. The PGC is in scan mode, its scan-out ports are connected to its scan-in ports and the state monitoring block. Assume that each scan chain contains ' l ' flip-flops, by circulating the scan chains for ' l ' clock cycles, the state monitoring block generates and stores the parity bits of the PGC states. The state monitor decodes the states after the PGC is powered on, where control signal 'sel' is '1' and scan enable signal 'se' is '1'. The PGC is in scan mode, its scan-in ports are connected to the output of the error correction block, PGC scan-out ports are connected to the state monitoring block and the error correction block. The state monitoring block checks the states of the power gated circuit against the stored parity bits. When errors are detected state monitoring block sends the error location to the error correction block which corrects the corrupted states and feeds back to the circuit. In manufacturing test mode, the control signal 'sel' is '2' and scan enable signal 'se' is '1', the PGC's scan-in ports are connected to the test scan-in ports and its scan-out ports are connected to the test scan-out ports. This means the proposed architecture for improving the reliability of state retention power gated design has no impact on the scan chains during manufacturing test mode, which is discussed in section III. There is no impact on power gated circuits' performance (critical path) in normal operation. This is because all state monitoring is done in scan mode.

B. State Monitoring Control

Fig. 3 (a) shows the conventional power gating control flow. The power gated circuit starts from active mode. When signal 'sleep' is '1', it starts the sleep sequence, including saving the circuit's states and turning off the power transistors, then the circuit enters the sleep mode. When signal 'sleep' is '0' it starts the wake-up sequence, including turning on the power transistors and restoring the circuit's states when the power supply become stable, then the power gated circuit enters active mode. The state monitoring control can be integrated into power gating control. The state monitoring block (Fig. 2) generates and stores parity bits before the sleep sequence, and

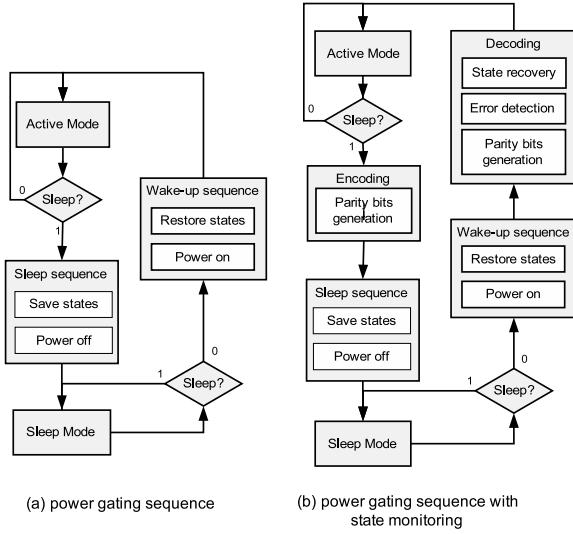


Fig. 3. Control sequence for power gating (a) conventional (b) proposed

checks the circuit's states against the stored parity bits after the wake-up sequence. Fig. 3 (b) shows the power gating control flow with state monitoring. The power gated circuit starts from active mode. When the signal 'sleep' is '1', it first starts the encoding sequence, state monitoring block generates and stores the parity bits of the circuit's state. Then follows the sleep sequence including saving the circuit's states to retention registers and switching off the power transistors. Finally the circuit goes into sleep mode. In sleep mode when signal 'sleep' is '0', it starts the wake-up sequence including switching on power transistors and restoring the circuit's states. The decoding sequence starts after the wake-up sequence, where another set of parity bits are generated and compared to the stored parity bits, if they are the same the circuit goes back to active mode, otherwise it raises an appropriate error code (detection or correction) in case of an error. This is how the state monitoring block (Fig. 2) detects and corrects error states in the circuit.

C. Design Synthesis Flow

Fig. 4 shows the reliability-aware design synthesis flow for incorporating the proposed state monitoring methodology into a conventional power gated design. There are three inputs to the flow: The conventional power gated design; the configuration file for providing the quality solutions in terms of area, power, latency and energy¹; the templates of state monitoring block and the (proposed) power gating controller (whose control sequence is shown in Fig. 3(b)). The proposed reliability-aware synthesizer consists of four main steps: it first inserts scan chains into the power gated circuit, then generates state monitoring and error correction logic, configures the proposed power gating controller to incorporate state monitoring and recovery, and finally synthesizes the design. To test a working design flow synopsys DFT Compiler and Design Compiler are used. The output of the reliability-aware synthesizer is

¹In section V, we discuss various trade-offs including area overhead, encoding and decoding power and latency with different scan chain configurations.

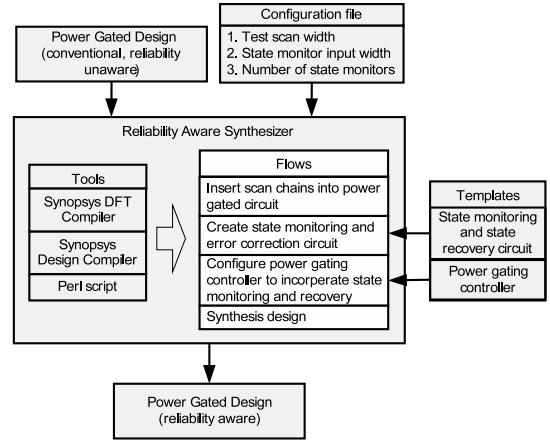


Fig. 4. Proposed design flow for reliable state retention power gating design

the power gated design that incorporate states monitoring and error corrections circuitry, which improves reliability in the presence of rush current and supply voltage fluctuation.

III. SCAN CHAIN CONFIGURATION

The main purpose of scan chains is for manufacturing test. The available scan chains are reused for state monitoring and Fig. 5 shows that scan chains can be configured to satisfy the requirement of both manufacturing test and state monitoring. However the configuration does have a cost in terms of area overhead, wake-up latency and energy consumption associated with the proposed state monitoring methodology through scan chain encoding and decoding. For the state monitoring block to generate parity bits, the power gated circuit's states need to be circulated through the scan chains. Assume there are ' l ' registers in each scan chain and clock period is ' T ', the encoding and decoding time is ' $l \times T$ '. If ' l ' is large, each encoding and decoding cycle can take a long time and therefore consume a significant amount of energy. To reduce latency and energy consumption, shorter scan chains are needed. To make each scan chain shorter, the number of scan chains can be increased.

The scan chains can be configured to reduce encoding and decoding time without affecting manufacturing test. Assume the test scan width (I/O width for manufacturing test) is 4 bits and the state monitoring block employs Hamming (7,4) code to monitor power gated circuit's states, with input width of 4 bits per state monitoring block. Assume that originally, a

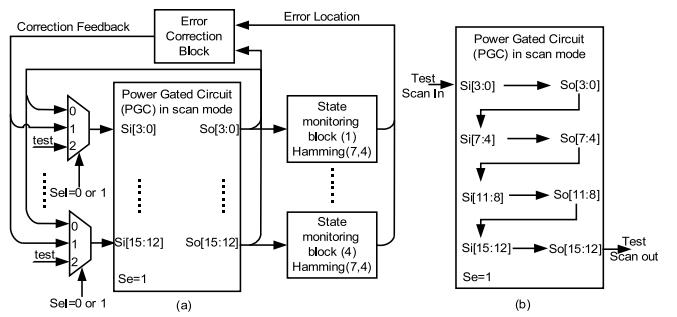


Fig. 5. Scan chain configurations (a) state monitoring (b) manufacturing test

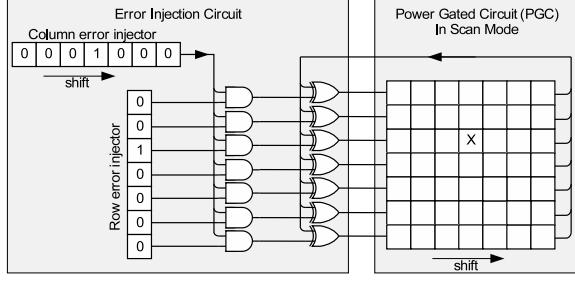


Fig. 6. Error injection

power gated circuit had 128 flip-flops and 4 scan chains in the circuit (1 test input bit per scan chain) and since state monitoring block's width is 4 bits, it will take $128 \div 4 = 32$ clock cycles for encoding and decoding the data. Next, consider that 128 flip-flops are re-ordered into 16 scan chains allowing 4 state monitoring blocks to work in parallel. The number of encoding and decoding clock cycles will then be $128 \div 16 = 8$, resulting in 4x speed-up. The latter 16 scan chain configuration is shown in Fig. 5 (a), as can be seen, 4 state monitoring blocks are operating in parallel taking data from 128 flip-flops configured in 16 scan chains. Fig. 5 (b) shows how this configuration can be re-used for 4 bit scan chain operation during manufacturing test, as can be seen the output of So[3:0] is fed back to Si[7:4] and so on, until test data is scanned out through So[15:12].

IV. METHODOLOGY VALIDATION

The validation of the proposed state monitoring and recovery for reliable power gated design consists of two stages: error injection and functional verifications.

Fig. 6 shows an error injection circuit used for injecting random single errors (Fig. 7 (a)) and multiple errors (Fig. 7 (b)). The error injection circuit consist of a column error injector and a row error injector which indicates the error injection location. Each fault injection cycle consist of two stages: it first generates random errors by setting the column and row injector using linear feedback shift registers; then the injection circuit injects errors through scan chains by flipping the scan-out data and this is fed back into the scan-in ports. For example to inject a single error into the flip-flop in 3rd row and 4th column shown in Fig. 6, we set the row injector to '0010000' (top → down), set the column injector to '0001000' (left → right) and set the circuit in scan mode. The column fault injector shifts in the same direction (to the right) as the scan chains. When the column injector's output is '0' the fault injection is disabled by the 'AND' gates. After three clock cycles the column injector's output is shifted to '1', it enables the fault injection of the 4th column. Then the row injector will flip the 3rd bit of the column using 'XOR' gates. In the 4th clock cycle the error is latched into the circuit.

The second part of the validation is the functional verification of a reliable power gated circuit with state monitoring and recovery implemented in Xilinx VirtexII-Pro FPGA. Although there is no power gating and scan chain insertion in FPGAs,

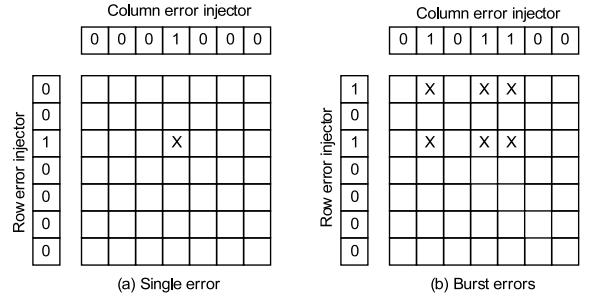


Fig. 7. Error injection pattern

the reliable power gating control sequence (Fig. 3) is emulated and the scan chain insertion is done in RTL using Perl script. To verify the proposed methodology we created a 32x32 bit FIFO circuit (as a case study) because it has high density of flip-flops and no error masking. 80 scan chains (selected for demonstration purpose) are created in the FIFO with 13 flip-flops in each scan chain. The state monitoring block (Fig. 2) uses both Hamming code and CRC code, they are chosen because of their effectiveness in improving memory circuit's reliability. The testbench setup is shown in Fig. 8. There are 5 components: FIFO_A consists of a FIFO module using proposed reliable power gated design and an error injection circuit; FIFO_B is the error-free reference FIFO module; "Stimulus" generates and writes random data to both FIFO_A and FIFO_B; "Comparator" reads the data from both FIFO_A and FIFO_B and compares them. "Counter" records each event when the errors are reported by FIFO_A and when the mismatches are reported by comparator. The test bench *test sequence* has 5 stages: 1. Reset both FIFO_A and FIFO_B to ensure they start in the same state; 2. Write to both FIFO_A and FIFO_B with the same random data; 3. Send sleep signal to FIFO_A; 4. Wait until FIFO_A is in sleep mode, then send wake-up signal to FIFO_A; 5. Read both FIFO_A and FIFO_B and compare the output. Repeat the test sequence for a required number of times before sending the result to a "PC" through the RS-232 serial port.

Two experiments are performed using 100 million test sequences, with each test sequence conducting fault injection. In the first experiment, a single error is injected per test sequence while multiple errors are injected in the second experiment. In the first experiment, the error correction circuitry detected and corrected all single errors per test sequence and therefore

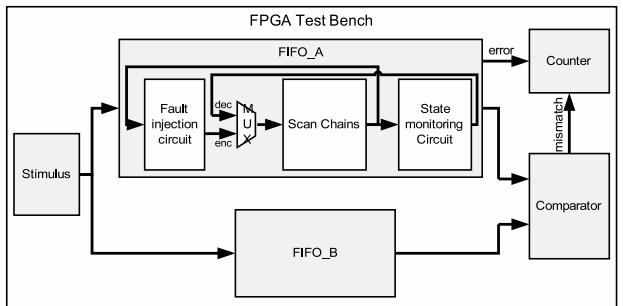


Fig. 8. FPGA test bench for functional verification of the methodology

TABLE I

ENCODING AND DECODING CIRCUIT AREA OVERHEAD, POWER, LATENCY AND ENERGY CONSUMPTION FOR CRC-16 CODE WITH DIFFERENT SCAN CHAIN CONFIGURATIONS

32x32 FIFO, CRC-16 code, 120nm, clock=100MHz									
W	l	Area		power(mW)		t(ns)		E(nJ)	
		μm^2	%	enc	dec	enc	dec	enc	dec
4	260	73658	2.8	4.99	4.99	2600	12.97	12.97	
8	130	73928	3.2	4.96	4.97	1300	6.45	6.46	
16	65	74614	4.2	4.96	4.98	650	3.22	3.24	
40	26	75762	5.8	5.13	5.17	260	1.33	1.34	
80	13	78208	9.2	5.14	5.25	130	0.67	0.68	

no error was reported by FIFO_A. This is further verified by comparing the outputs of FIFO_A and FIFO_B using the “Comparator” shown in Fig. 8. On the other hand, during the second experiment with multiple errors injection, none of the errors were corrected by the error correction circuitry. This is because burst errors occur randomly through out the test sequence and they are closely clustered, while Hamming code can correct only limited number of errors if they do not occur close to each other. However all these errors were accurately detected and reported to the counter; this is further validated by comparing the outputs of two FIFO blocks using comparator. These two experiments show that all injected single errors are corrected and all multiple errors are accurately detected.

V. TRADE OFF ANALYSIS

The proposed error detection and correction methodology is achieved through scan chain encoding and decoding. For detection CRC-16 and Hamming code are investigated. For correction there are two possible approaches: hardware error correction and software state recovery. Software recovery generally has higher latency than hardware correction. The target application is high performance design where low latency is often preferred so hardware error correction is studied. In this section, the trade-offs of the state monitoring circuit’s area overhead, encoding and decoding time and power related to the implementation of two type of coding (Hamming code and CRC code) with different scan chain configurations are discussed. The terms latency and encoding and decoding times are used interchangeably. The 32x32 bits ‘FIFO’ was used as a test circuit. The design is synthesized using STmicroelectronics 120nm technology. The area is generated from Synopsys Design Compiler. The gate level netlist of the power gated design is simulated in a Cadence simulator, and the encoding and decoding power is calculated by Synopsys Prime Time PX, the circuit is clocked at 100MHz for demonstration purpose.

Table I shows the area, power, latency and energy when implementing the reliable power gated FIFO using CRC-16 code. The 1st column shows the number of scan chains, the 2nd column shows the scan chain length, followed by the area of FIFO circuit and state monitoring logics overhead, the 5th and 6th columns shows the power consumption, the 7th column shows the timing performance, and finally last two columns show energy consumption. As the number of scan chains W increases from 4 to 80, the length of scan chain

TABLE II

ENCODING AND DECODING CIRCUIT AREA OVERHEAD, POWER, LATENCY AND ENERGY CONSUMPTION FOR HAMMING (7,4) CODE WITH DIFFERENT SCAN CHAIN CONFIGURATIONS

32x32 FIFO, Hamming (7,4) code, 120nm, clock=100MHz									
W	l	Area		power(mW)		t(ns)		E(nJ)	
		μm^2	%	enc	dec	enc	dec	enc	dec
4	260	120594	68.4	6.76	6.72	2600	17.58	17.47	
8	130	121552	69.7	6.91	6.86	1300	8.98	8.92	
16	65	123303	72.1	7.11	7.00	650	4.62	4.55	
40	26	126811	77.0	7.72	7.45	260	2.00	1.94	
80	13	134141	87.3	8.43	8.05	130	1.08	1.05	

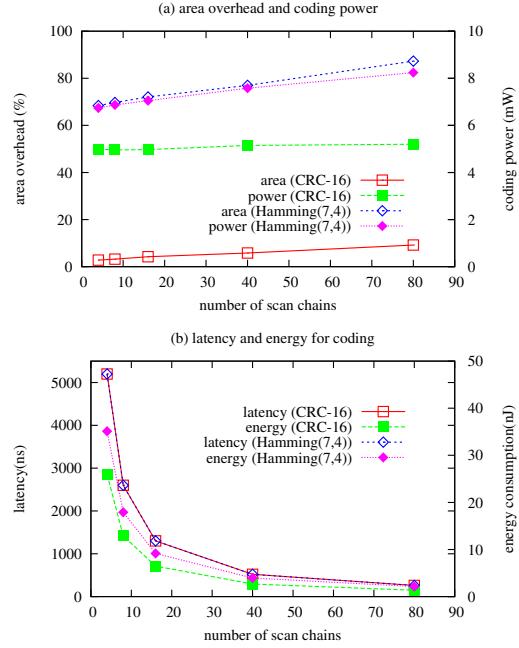


Fig. 9. Implementation trade-offs related to state monitoring and recovery for CRC-16 code and Hamming code

l decreases from 260 to 13, the encoding and decoding time decreases from 2600 ns to 130 ns. This is because encoding and decoding time is equal to the product of the scan chain length and clock period (Section. III). The increase in the number of scan chains (from 4 to 80) results in area overhead of 2.8% to 9.2%, this is because higher number of scan chains require additional state monitoring blocks (Fig. 5 (a)) for encoding and decoding. Power consumption increases slightly (from 4.99 mW to 5.14 mW) with increase in area. The encoding and decoding energy decreases (from 12.97 nJ to 0.67 nJ) with the increase in the number of scan chains, because energy is the product of power and time. With the increase of scan chains the power increase only by 3% while latency decreases by 95%, which results in overall reduction in energy consumption. Similarly Table II shows the area, power, latency and energy by using Hamming (7,4) code on the same power gated FIFO. It shows a similar trend in terms of different scan chains configurations.

Fig. 9 (a) shows the area overhead and encoding and decoding power trade offs for CRC-16 code and Hamming (7,4) code implementation. CRC-16 code have small area

overhead starting from 2.8% with 4 scan chains and increase to 9.2% with 80 scan chains. The area overhead of Hamming (7,4) code varies from 68% with 4 scan chains to 87% with 80 scan chains. Error detection and correction code requires more redundancy than error detection code. Despite the higher area overhead, the encoding and decoding power of Hamming (7,4) code is only between 20% and 40% higher than CRC-16 code. This is because the majority of the encoding and decoding power is due to scan chains switching which is common in both implementations. Fig. 9 (b) shows the encoding and decoding time and energy trade-offs of CRC-16 code and the Hamming (7,4) code. The encoding and decoding time for both codes is the same because latency is only affected by the scan chains length. The encoding and decoding of Hamming (7,4) code consumes around 20% to 40% more energy than CRC-16 code. Fig. 9 (b) also shows for both codes that by increasing the number of scan chains, the encoding and decoding time and energy reduces significantly at the cost of relatively small increase in area and power as shown in Fig. 9 (a).

There are other Hamming codes with lower area overhead than the one shown in Table II. Table III shows the area overhead and power consumption of different Hamming codes. The 1st column specifies implemented Hamming code, the 2nd column shows the number of scan chains inserted, the 3rd column shows the area overhead, the 4th column shows the power consumption and the last column shows the maximum error correction capability of each implementation. The redundancy of Hamming (n,k) code is equal to the ratio of the parity bits to the information bits: $\frac{n-k}{k}$. Higher redundancy correspond to higher area overhead and higher error correction capability. As can be seen, the area overhead is minimum with Hamming (63,57) code which has least error correcting ability (1.59%). Overall, the area overhead can be reduced from 84.8% to 15.9% using different Hamming codes at the cost of error correction ability that decreases from 14.3% to 1.59%. If large area overhead is not acceptable then the approach of CRC error detection with software recovery may be considered.

The error correction capability of 4 types of Hamming codes with error injections are investigated. Errors were randomly injected in a test sequence of 1000 bits (therefore emulating 1000 flip-flops) and upto 10 errors were injected per test sequence. In total one million test sequences were simulated. The test sequence is then passed through the 4 types of Hamming code implementation separately and the outcome is shown in Fig. 10. As can be seen, Hamming (7,4) code has best error correction capability, it corrects 98.81% errors with double errors injection and 94.14% errors with 10 errors injection. Hamming (63,57) code has least error correction capability, it corrects 88.65% of errors with double errors injection and 52.96% errors with 10 errors injection.

VI. CONCLUSION

An efficient design methodology for improving the reliability of power-gated design by protecting state integrity of state retention registers has been proposed, through state monitoring and correction. This is achieved by exploiting the available

TABLE III
ENCODING AND DECODING CIRCUIT AREA OVERHEAD, POWER, LATENCY AND ENERGY CONSUMPTION FOR DIFFERENT HAMMING CODES

32x32 FIFO, Hamming code, 120nm, clock=100MHz						
code	W	Area(μm^2)			power(mW)	cap(%)
		FIFO	total	%		
(7,4)	56	71628	132338	84.8%	8.21	7.84
(15,11)	55	71628	101681	42.0%	6.52	6.34
(31,26)	52	71628	88311	23.2%	5.89	5.82
(63,57)	57	71628	82987	15.9%	5.64	5.62
						1.59%

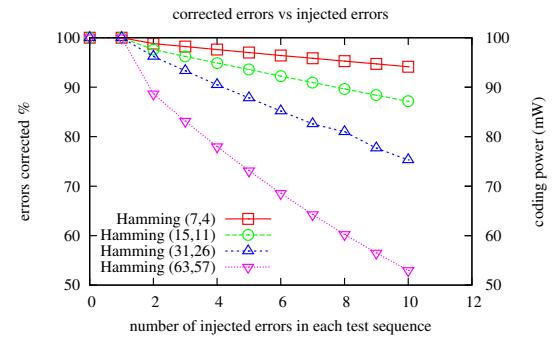


Fig. 10. The error correction ability of Hamming codes when multiple errors injected in each test sequence of 1000 flip-flops

scan chain without affecting manufacturing test and the critical paths of power gated circuits. The proposed methodology has been validated using an FPGA synthesized design and shows 100% error detection both for single error and multiple error injection, and it achieves 100% error correction in case of single errors. Using synthesized designs, it is shown that the proposed methodology can be incorporated into the power gating design flow.

ACKNOWLEDGMENT

The authors would like to thank the EPSRC-UK for funding this work in part under grant number EP/E035965/1.

REFERENCES

- [1] S. Borkar, "Design challenges of technology scaling," *Micro, IEEE*, vol. 19, no. 4, pp. 23–29, Jul-Aug 1999.
- [2] J. Robertson, "High dielectric constant gate oxides for metal oxide silicon transistors," *Reports on Progress in Physics*, vol. 69, no. 2, pp. 327–396, 2006.
- [3] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold cmos circuits," in *DAC 1998*, Jun 1998, pp. 489–494.
- [4] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: simple techniques for reducing leakage power," in *Computer Architecture, 2002. Proceedings. 29th Annual International Symposium on*, 2002, pp. 148–157.
- [5] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tanabe, and J. Yamada, "A 1-v high-speed mtcmos circuit scheme for power-down application circuits," *Solid-State Circuits, IEEE Journal of*, vol. 32, no. 6, pp. 861–869, Jun 1997.
- [6] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual for System-on-Chip Design*. Springer, 2007.
- [7] S. Kim, S. Kosonocky, and D. Knebel, "Understanding and minimizing ground bounce during mode transition of power gating structures," in *ISLPED '03*, Aug. 2003, pp. 22–25.
- [8] S. Henzler, G. Georgakos, J. Berthold, and M. Eireiner, "Activation technique for sleep-transistor circuits for reduced power supply noise," in *ESSCIRC 2006*, Sept. 2006, pp. 102–105.