Leakage Current Reduction in Sequential Circuits by Modifying the Scan Chains

Afshin Abdollahi
University of Southern California
(310) 592-3886
afshin@usc.edu

Farzan Fallah
Fujitsu Laboratories of America
(408) 530-4544
farzan@fla.fujitsu.com

Massoud Pedram
University of Southern California
(213) 740-4458
pedram@ceng.usc.edu

Abstract - Input vector control is an effective technique for reducing the leakage current of combinational VLSI circuits when these circuits are in the sleep mode. In this paper a design technique for applying the minimum leakage input to a sequential circuit is proposed. Our method uses the built-in scan-chain in a VLSI circuit to drive it with the minimum leakage vector when it enters the sleep mode. Using these scan registers eliminates the area and delay overhead of the additional circuitry that would otherwise be needed to apply the minimum leakage vector to the circuit. We show how the proposed technique can be used for several different scan-chain architectures and present the experimental results on the MCNC91 benchmark circuits.

1. Introduction

As technology scales down, the supply voltage must be reduced to prevent the gate insulator break down. Voltage reduction has the added benefit of reducing the dynamic power consumption in a VLSI circuit. However, voltage down-scaling results in a linear increase in the propagation delay of the logic gates. Therefore, the threshold voltage of the transistors must be lowered to maintain the circuit speed. This reduction in V_{th} results in a significant increase in the leakage current, which increases the static power consumption in the circuit.

There are three main sources for leakage current:

1. Source/drain junction leakage current
2. Gate direct tunneling leakage
3. Sub-threshold leakage through the channel of an OFF transistor

The junction leakage occurs from the source or drain to the substrate through the reverse-biased diodes when a transistor is OFF. The magnitude of the diode's leakage current depends on the area of the drain diffusion and the leakage current density, which is in turn determined by the process technology.

The gate direct tunneling leakage flows from the gate thru the "leaky" oxide insulation to the substrate. Its magnitude increases exponentially with the gate oxide thickness T_{ox} and supply voltage V_{DD}. According to the 2001 International Technology Roadmap for Semiconductors, high-K gate dielectric reduced direct tunneling current is required to control this component of the leakage current for low standby power devices.The sub-threshold current is the drain-source current of an OFF transistor. This is due to the diffusion current of the minority carriers in the channel for a MOS device operating in the weak inversion mode (i.e., the sub-threshold region.) For instance, in the case of an inverter with a low input voltage, the NMOS is turned OFF and the output voltage is high. Even when VGS is 0V, there is still a current flowing in the channel of the OFF NMOS transistor due to the VDD potential of the VDS. The magnitude of the sub-threshold current is a function of the temperature, supply voltage, device size, and the process parameters out of which the threshold voltage (V_{th}) plays a dominant role.

In current CMOS technologies, the sub-threshold leakage current is much larger than the other leakage current components. This current can be calculated by using the following equation:

$$I_{DS} = K\left(1 - e^{-\frac{V_{DS}}{V_T}}\right) e^{\frac{(V_{GS} - V_T + \eta V_{DS})}{nV_T}}$$

where K and n are functions of the technology, and η is the drain-induced barrier lowering coefficient. Clearly, decreasing the threshold voltage increases the leakage current exponentially. In fact decreasing the threshold voltage by 100 mv increases the leakage current by a factor of 10. Decreasing the length of transistors increases the leakage current as well. Therefore, in a chip, transistors that have smaller threshold voltage and/or length due to process variation contribute more to the overall leakage. Although previously the leakage current was important only in systems with long inactive periods (e.g., pagers and networks of sensors), it has become a critical design concern in any system in today's designs.

In the recent past, many researchers have proposed techniques for leakage power reduction in VLSI circuits. These techniques range from process technology-based solutions to circuit-level and even architectural solutions. [1-7] In this paper, we propose a new technique based on controlling the input vector to a circuit when it enters the sleep mode. Our proposed technique is applicable to both combinational and sequential circuits. For the latter type of circuits, which are the focus of the present paper, our method requires only modification of the

Proceedings of the Fourth International Symposium on Quality Electronic Design (ISQED'03)
0-7695-1881-8/03 $17.00 © 2003 IEEE

scan-chains that are already put into the circuit in order to allow efficient testing of the circuit functionality. No other change to the circuit in question is required. So from a designer's perspective, the cost of reducing leakage in a standby circuit is minimal.

Scan-based testing is the dominant method for testing VLSI chips [8-9]. We modify scan-chains so they can be used to drive the circuit with the *minimum leakage vector* (*MLV*). This reduces the leakage current of the circuit while it is in the sleep mode. All proposed input vector control methods [3-5,10] require modification of the circuit and adding some multiplexers and/or gates to drive the circuit with the *MLV*s. Modifying the circuit increases the delay of its critical paths. Therefore, there is a delay penalty associated with the existing MLV-based methods. In contrast, our proposed method does not affect the delay of the critical paths of the circuit. Therefore, there is no delay penalty associated with our method.

The rest of this paper is organized as follows. Section 2 describes the input vector control method for decreasing the leakage current of a combinational circuit. In Section 3 scan-based testing is described. Our method for modifying the scan-chain of a sequential circuit to decrease its leakage current is presented in Section 4. Experimental results are presented in Section 5, while Section 6 gives the conclusion.

2. Input Vector Control Method

The leakage current of a logic gate is a strong function of its input values. The reason is that the input values affect the number of OFF transistors in the NMOS and PMOS networks of a logic gate. For example, the minimum leakage current of a two-input NAND gate corresponds to the case when both its inputs are zero. In this case, both NMOS transistors in the NMOS network are off, while both PMOS transistors are on. The effective resistance between the supply and the ground is the resistance of two OFF NMOS transistors in series. This is the maximum possible resistance. If one of the inputs is zero and the other is one, the effective resistance will be the same as the resistance of one OFF NMOS transistor. This is clearly smaller than the previous case. If both inputs are one, both NMOS transistors will be on. On the other hand, the PMOS transistors will be off. The effective resistance in this case is the resistance of two OFF PMOS transistors in parallel. Clearly, this resistance is smaller than the other cases. There is also the "stack effect" i.e., the phenomenon whereby the leakage current through a stack of two OFF transistors of W/L ratios each is lower than that of a single OFF transistor with a W/2L ratio. This is mainly because of the body effect, which causes an increase in the effective resistance of the two-transistor chain compared to that of a single transistor.

In summary, logic gates exhibit widely varying leakage currents as a function of the applied input pattern. As a result, the leakage current of a circuit is a strong function of values of its primary input and outputs of the flip-flops. Abdollahi et al. [10] used this fact to reduce the leakage current in purely combinational circuits. They formulate the problem of finding the MLV using a series of Boolean Satisfiability problems. Using this vector to drive the circuit while in the STANDBY state, they reduce the circuit leakage by as much as 35% .

Having found the minimum leakage pattern, one can use this vector to drive the circuit while in the sleep mode. This requires the addition of some multiplexers at the primary inputs of the circuit. The multiplexers are controlled using a sleep signal. In this paper, we assume that the sleep signal is provided externally or is generated by an on-chip power management unit, which is independent of the realization of the circuit in question. In practice, because one input of each multiplexer is a constant *0* or *1*, the multiplexers can be simplified to an *AND* or *OR* gate.

Figure 1 shows the input driver for two bits $\{a_1, a_0\}$ assuming the required *MLV* is *{1, 0}*.

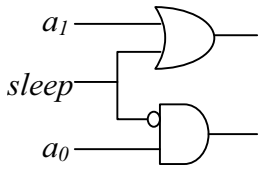

Figure 1. Input driver for *MLV {1, 0}*.

Notice that such a technique can reduce the total power consumption of the circuit (dynamic plus leakage) only for long periods of circuit sleep time. Therefore, the sleep signal should be activated only if the circuit sleep period is longer than a specified threshold.

3. Scan-Based Testing

In Figure 2, we consider a sequential circuit comprised of a combinational circuit and a set of flip-flops.

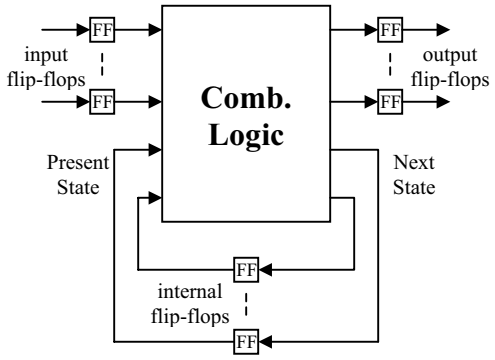

Figure 2. A general model of a sequential circuit.

In the scan-based designs, the flip-flops are connected in such a way that they enable two modes of operation: **normal** mode and **test** mode. In the normal mode, the flip-flops are connected as shown in Figure 2. At each clock cycle, the next state is stored in the flip-flops. In the test mode, the flip-flops are reconfigured and form one or more shift registers, called scan registers or scan chains. At each clock cycle the values of the flip-flops are shifted. The values can be observed through the output of the last flip-flop of the scan chain. Furthermore, the values can be shifted into the scan-chain through the input of the first flip-flop in the chain.

In this paper, we assume that all internal and external (input and output) flip-flops are included in the scan chain. This type

of circuit is called full-scan. Full scan chains convert the problem of testing a sequential circuit to that of a combinational one. In other words, the input and internal flip-flops can be treated as primary inputs of the circuit, whereas the output and internal flip-flops are considered as the primary outputs. In order to test a circuit, the circuit is first switched to the test mode and the present state value is shifted into the flip-flops. After that the circuit is switched to the normal mode and operates for one or more cycles under the externally provided input values. In the next step, the circuit is switched back to the test mode and the next state value is shifted out.

As mentioned before, the scan-based test methodology requires the modification of the circuit and addition of a test mode in which the flip-flops are configured as one or more scan chains. For this reason, the flip-flop design must be modified. One way to add the new functionality into the flip-flops is through the addition of a multiplexer with inputs D and D_S, as shown in Figure 3.

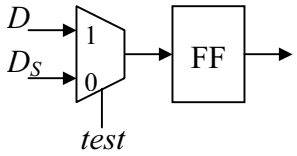

Figure 3. A multiplexed-input scan flip-flop.

The control input of the multiplexer is controlled by the test signal. This design is referred to as a **multiplexed-input scan** flip-flop. Each flip-flop in the circuit may be replaced by such a flip-flop where its D input is connected to the corresponding state output in the circuit and its D_S input is connected to the output of another flip-flop, which is designated as the predecessor of the current flip-flop in the scan chain. Input D_S of the first flip-flop in a chain is the scan chain input and is denoted by *ScanIn*, while the output of the last flip-flop in the chain is the output of the scan chain and is denoted by *ScanOut*. The input and the output of a chain are connected to an input and an output pin of the chip, respectively.

Figure 4 shows details of a scan chain design. In the Figure, the flip-flops are configured as a single chain.

The use of scan allows the desired value to be shifted into each flip-flop, or **scanned in**, using the test mode and scan chains. Hence, present state of the sequential circuit can be directly controlled. This increases the controllability. After applying a test vector, the values at state outputs are captured into the flip-flops by configuring them in their normal mode. The captured values are shifted out or **scanned out**, using the test mode and observed at the corresponding scan output pin, *ScanOut*. This means the next state of the sequential circuit becomes observable. This increases the observability.

Assuming the flip-flops are configured as a single chain, the following steps are used to apply a test vector.

1. The circuit is set into test mode by setting *test=0*.
2. Shift the test vector into flip-flops via *ScanIn* pin by applying $m+k$ clocks, where m and k are the number of input and internal flip-flops, respectively. This causes the test vector be applied to the primary inputs (including present state) of the circuit.
3. The circuit is configured in its normal mode by setting *test=1* and one clock is applied. This causes the response at the primary outputs (including next state) of the circuit be captured in the corresponding flip-flops.
4. The state response captured in the scan flip-flops is scanned out and observed at the ScanOut pin by setting *test=0* and applying $k+n$ clocks, where n is the number of output flip-flops.

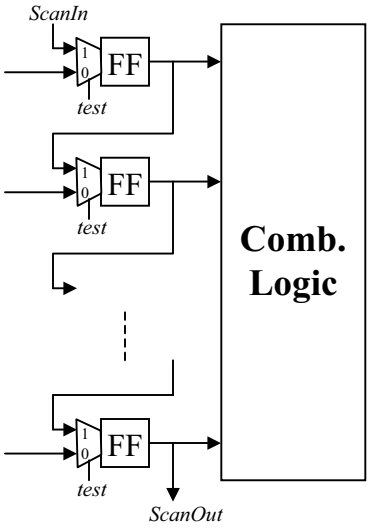

Figure 4. A generic scan chain structure.

4. Using the Scan Chain for Leakage Reduction

In this section we describe how scan chains can be modified to allow us to apply the *MLV* to a sequential circuit when it is in the sleep mode. Because scan-chains provide an easy way to control the values of flip flops, they can be used to drive the standby circuit with the MLV.

A simple way is to shift in the *MLV*, from a memory ($m+k$ bit shift register) into the first $m+k$ flip-flops via the *ScanIn* pin by setting the circuit into the test mode and applying $m+k$ clocks. For this reason the sleep signal, generated by the power management unit, is combined with the test signal to construct the new control input of the multiplexed flip-flops. After shifting in the MLV, the clock signal can be disabled to avoid power dissipation in the flip-flops as depicted in Figure 5.

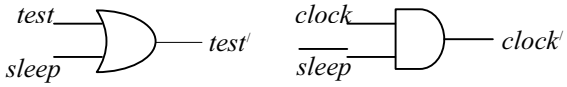

Figure 5. New test and clock signals.

With such a method, the previous state of the circuit is over written by the *MLV*. If the next state or output of the circuit, while switching back to the active mode, is a function of the previous state, then this method will obviously change the functionality of the circuit.

There are many cases in which it is not necessary to know the previous state of the machine upon backer-entering the active mode of operation. As an example, consider the floating-point unit of a microprocessor. After executing a floa

instruction, the unit can be switched back to the idle mode if there are no more floating-point instructions. Upon encountering a floating-point instruction, the unit can be switched back to the active mode. In this case it is not necessary to know the previous state of the unit and the circuit will function properly. On the other hand, there are cases where it is necessary to save the state of the circuit and restore it upon switching back to the active mode. To address this requirement, we propose to add a circuit loop comprised of the input and internal flip-flops and an (*m+k)*-bit shift register as depicted in Figure 6.

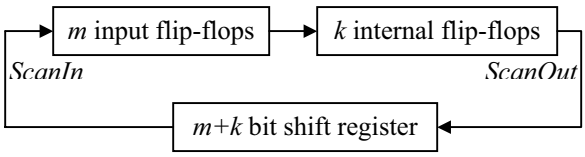

Figure 6. Configuration of the scan chain in the sleep mode.

In this way, the state of the circuit can be saved by shifting out the values of the flip-flops via the output of the $(m+k)^{th}$ flip-flop (i.e., the last internal flip flop) in the chain, which can be considered as a *ScanOut* pin, to memory. This memory can be the same (*m+k)*-bit shift register that is used for storing the *MLV*. Shifting in the state can be done at the same time that the *MLV* is shifted out. Before switching back to the active mode, we need to shift in the previous state saved in the memory to the internal flip-flops via the *ScanIn* pin by applying *m+k* clocks. Simultaneously, the *MLV* captured in the flip-flops of the circuit is shifted into the memory to be used in the next sleep period.

The performance penalty associated with this method is *m+k* clock cycles, if the length of the sleep period, *t*, is larger than *m+k* clock cycles (because it takes m+k clock cycles to load the saved state from the shift register into the flip-flop;) otherwise the performance penalty is *2(m+k)-t* clock cycles (because we need to return the state values to the flip-flops via the loop.) If we use separate memories (*m+k* bit shift register for the *MLV* and *k* bit shift register for the state values,) the performance penalty can be reduced to *k* clock cycles, if the sleep period is more than *m+k* clock cycles; otherwise, the performance penalty is *(m+2k)-t* clock cycles due to similar reasons.

This method takes advantage of the built in scan structures in the circuit and does not require any modification to the circuit. Therefore, *there is no delay penalty while the circuit is in the active mode.* The fact that this method does not require any changes in the gates of the circuit or any process technology modification makes it very easy to use. On the other hand, it takes several clock cycles to switch between the active and the sleep modes.

Now we describe some modification to the scan chain in order to apply the *MLV* to the circuit in one cycle. For this reason *m+k* new multiplexers are inserted in the scan chain, in such a way that each output of a flip-flop in the scan chain is multiplexed with the corresponding minimum leakage value and the output of the multiplexer is connected to the D_S input of the next multiplexed-input flip-flop as depicted in Figure 7.

The test signal needs to be set to *one* whenever the circuit enters the sleep mode, which can be done by using the circuit in Figure 5. The added multiplexers can be simplified since one of their inputs is always the minimum leakage value, which is a constant number as shown in Figure 1.

This method over writes the previous state of the circuit with the *MLV*. To solve this problem we add *m+k* flip-flops and multiplexers controlled by the sleep signal to the circuit, which are used to save the *MLV* in the active mode and the previous state in the sleep mode. For this reason we construct a local loop corresponding to each input as shown in Figure 8.

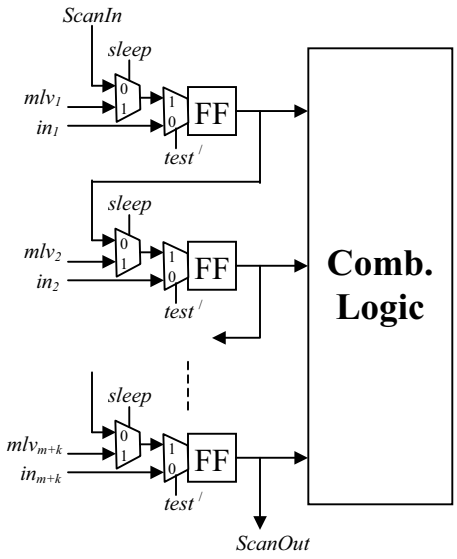

Figure 7. Modified scan chain for applying *MLV* in one cycle.

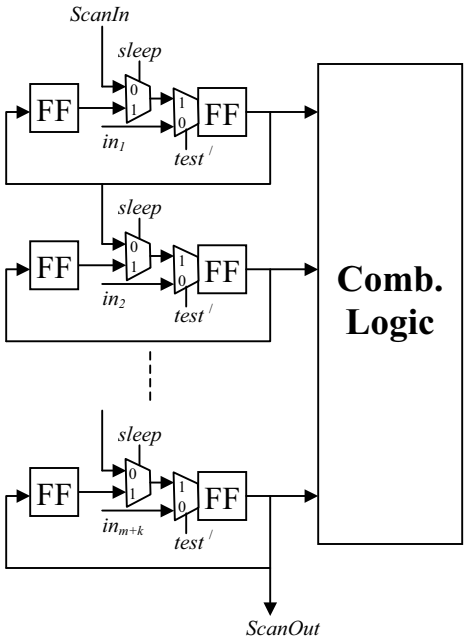

Figure 8. Adding extra flip-flops for state recovery.

Disabling the clock as shown in Figure 5 may not lead to correct results. For correct functionality, the clock needs to be disabled one cycle after entering the sleep mode and it needs to be enabled one cycle before entering the active mode. Figure 9 shows the appropriate timing of the circuit.

In this timing diagram V_1 shows the values captured in the multiplexed-input flip-flops in the scan chain and V_2 shows the values captured in the additional flip-flops. It can be seen that when the sleep signal is high, the current state will be saved in the added flip-flops; at the same time the *MLV* is loaded into the multiplexed-input flip-flops driving the inputs of the combinational circuit. Additionally, before switching to the active mode the previous state is captured in the multiplexed-input flip-flops and the *MLV* is captured in the additional flip-flops concurrently.

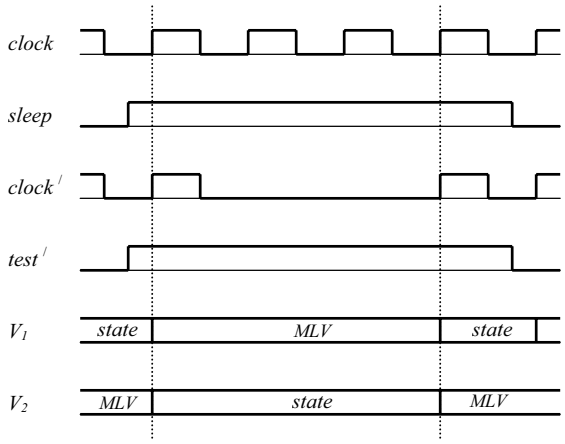

Figure 9. Timing diagram of control signals

In some sequential circuits single-latch design is used rather than flip-flop design in which a pair of latches in a master-slave configuration are used. Figure 10 illustrates the single-latch design in which two non-overlapping clocks C_1 and C_2 must be used. In such a design if there exits a combinational path from the output of a latch clocked with C_1 to the input of another latch, then that latch must be clocked by C_2.

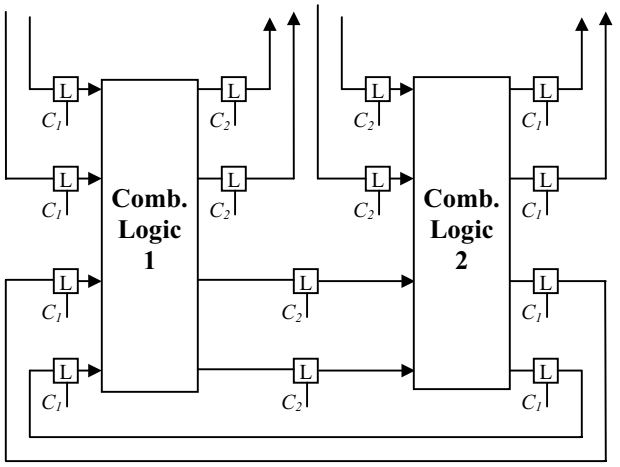

Figure 10. A single latch sequential circuit

Now we describe scan chain design for single-latch circuits. A memory element in a scan design must be capable of selecting the value from one of its two inputs, namely, the state output in the active mode and the scan output of the previous element in the chain in the test mode. Furthermore, since multiple scan elements must be connected as a shift-register, each scan element must have a functionality that is equivalent to that of a flip-flop or a master-slave latch configuration. For this reason each latch is replaced by a multiplexed input latch, similar to the previously described multiplexed input flip-flop. Furthermore, for each latch, an additional latch clocked by a different phase is added to construct the master-slave configuration in the scan chain as illustrated in Figure 11.

In the active mode extra latches hold the *MLV* and the C_3 clock is kept low. While entering the sleep mode by applying a pulse to C_2, the state is saved in *L'* latches.

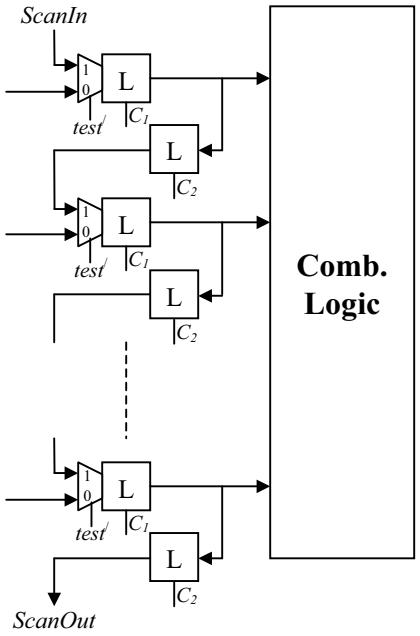

Figure 11. Scan chain structure for single-latch sequential circuits

Similar to the previous case in order to apply the *MLV* in the sleep mode and recover the state when entering the active mode, for each latch, an extra latch clocked by a different clock C_3 and a multiplexer controlled by the sleep signal are added. The extra multiplexers are controlled by the sleep signal as shown in Figure 12.

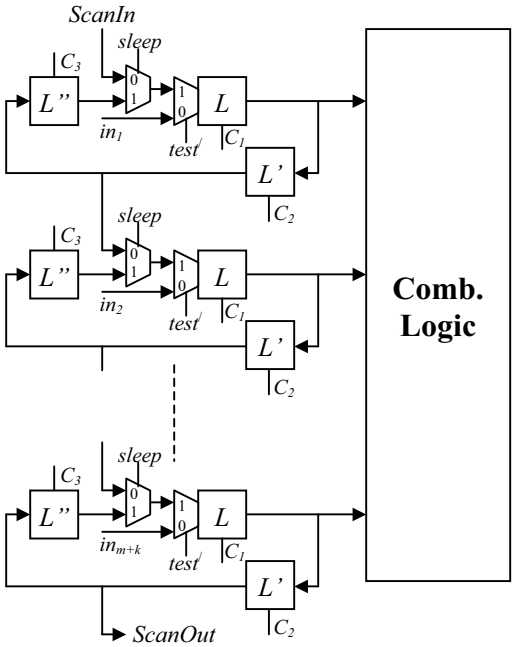

Figure 12.Adding extra latches and multiplexers for state

Then, by applying a pulse to C_1 and setting *sleep=1*, which results in *test'=1* as shown in Figure 5, the *MLV* is loaded to *L* latches driving the combinational circuit. In the next step, applying a pulse to C_3 captures the state values, saved in *L'* latches, into the *L"* latches. This way the data in *L* and *L"* latches are swapped via *L'* latches by applying appropriate pulses to C_1, C_2 and C_3. Hence, during the sleep period *L"* latches keep the previous state of the circuit. While entering the active mode, the state can be recovered in *L* latches by swapping data in *L"* and *L* latches by taking a similar approach. Figure 13 shows the timing diagram of the circuit.

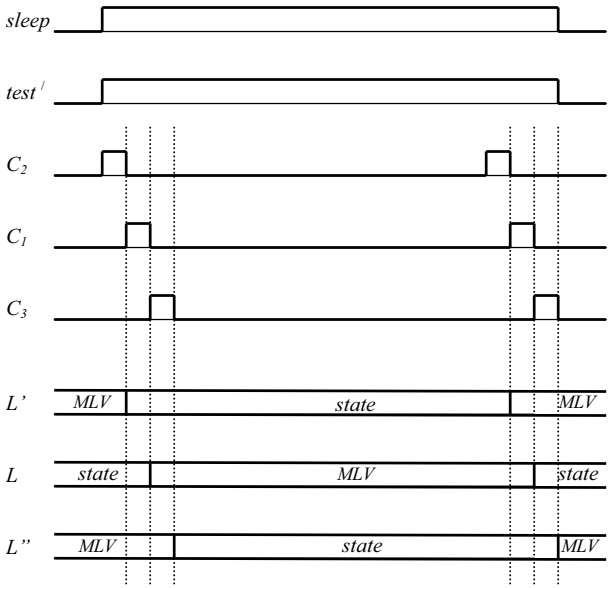

Figure 13. Timing diagram of control and clock signals

5. Experimental Results

We applied our leakage reduction methods on **ISCAS89** benchmark circuits. Each method is associated with some delay overhead. We have compared the delay overhead of our methods with the previous method, which does not modify the scan chain of circuits. Table 1 shows the leakage reduction percentage using input vector control.

Circuit	Leakage reduction	Circuit	Leakage reduction	Circuit	Leakage reduction	Circuit	Leakage reduction
S1196	26%	S35932	16%	S208	36%	S5378	19%
S1238	25%	S382	34%	S27	39%	S641	23%
S1423	19%	S386	27%	S298	35%	S713	31%
S1488	31%	S400	34%	S344	33%	S820	33%
S1494	32%	S510	29%	S349	31%	S838	33%

Table 1. Leakage reduction percentage using input vector control

The techniques illustrated in Figures 6 and 7 do not modify the critical paths of the circuit, therefore there is no delay overhead associated with this these methods in the active mode. However the method in Figure 6 is associated with a performance penalty and the method in Figure 7 is not able to recover the state. The method in figure 8 is associated with an area overhead and slight delay overhead because of additional capacitive load of extra flip-flops driven by multiplexed-input flip-flops. Table 2 shows the comparison of delay overhead of our method with standard input control method (using multiplexers in the primary inputs of the combinational circuit, which is on the critical path.)

6. Conclusions

In this paper we presented some techniques for reducing the leakage current of a sequential circuit using its minimum leakage vector. In our method, we modify the scan chain of the circuit and use it to drive the circuit with the minimum leakage vector while the circuit is in standby mode. This effectively eliminates the delay overhead associated with the vector-based methods. Our method results in the loss of the previous state of the sequential circuit. In order to save the state information and restore it upon switching back to the active mode, some extra latches can be added to the circuit. We presented several latch architectures to achieve this goal.

Circuit	Delay Overhead		Circuit	Delay Overhead	
	Standard method	Our method		Standard method	Our method
S1196	10%	1%	S35932	8%	0%
S1238	9%	1%	S382	14%	1.2%
S1423	4%	0%	S386	15%	1.2%
S1488	12%	1%	S400	13%	1.1%
S1494	11%	1%	S510	12%	1%
S208	15%	1.4%	S5378	11%	1%
S27	17%	1.5%	S641	10%	1%
S298	13%	1.2%	S713	9%	1%
S344	12%	1%	S820	12%	1%
S349	13%	1.1%	S838	13%	1.1%

Table 2. Comparison of delay overhead of the proposed method with standard method

References

[1] Ferre, A. and Figueras, J., "Characterization of Leakage Power in CMOS Technologies", IEEE International Conference on Electronics, Circuits and Systems, Vol. 2, 1998, pp. 85 –188.

[2] Cheng, Z., Johnson, M., Wei, L. and Roy, K., "Estimation of Standby Leakage Power in CMOS Circuits Considering Accurate Modeling of Transistor Stacks", ISLPED 98, pp. 239-244.

[3] Johnson, M., Somasekhar, D. and Roy, K., "Models and Algorithms for Bounds in CMOS Circuits", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 18, No. 6, June 1999, pp. 714-725.

[4] Ye, Y., Borkar, S., and De, V., "A New Technique for Standby Leakage Reduction in High-Performance Circuits," Symposium on VLSI Circuits, 1998, pp. 40-41.

[5] Bobba, S. and Hajj, I., "Maximum Leakage Power Estimation for CMOS Circuits", Proceedings of the IEEE Alessandro Volta Memorial Workshop on Low-Power Design, 1999, pp. 116 –124.

[6] Johnson, M., Somasekhar, D. and Roy, K., "Leakage Control With Efficient Use of Transistor Stacks in Single Threshold CMOS ", Proceedings of the 36th Design Automation Conference (DAC), June 1999, pp. 442-445.

[7] Halter J., and Najm, F., "A Gate-level Leakage Power Reduction Method for Ultra Low Power CMOS Circuits, IEEE Custom Integrated Circuits Conference, 1997, pp. 475-478.

[8] Gupta S. "Digital System Testing", to be published by Cambridge University Press

[9] Abramovici, M., Breuer, M.A., Friedman, A.D.," Digital Systems Testing and Testable Designs", Computer Science Press, New York, 1995

[10] Abdollahi, A.; Fallah, F.; Pedram, M., **"**Runtime mechanisms for leakage current reduction in CMOS VLSI circuits**"** Low Power Electronics and Design, 2002. ISLPED '02. Proceedings of the 2002 International Symposium on, 2002, Page(s): 213 -218