

Universal CMOS Gate for Parallel Test of Logic Circuits

S. Samavi* and A. Ferdowsi¹

With the increasing complexity of integrated circuits, the cost of testing is on the rise. Design for testability methodologies with the introduction of extra hardware tries to reduce different cost factors associated with the testing of a complex circuit at the expense of added chip area. In this paper, a novel universal gate is proposed for the complete testability of combinational circuits. Also, an algorithm is devised to perform the simple testing of sequential circuits. The ease of testing, using the proposed structure, is independent of the size and complexity of the circuit. Two different designs using CMOS technology are proposed.

INTRODUCTION

Since the advent of the first RISC processor in the 1980's, the importance of testing and testability has grown constantly. This is because with the advancement of microelectronics, the complexity of integrated circuits is increasing. The density of circuits has an annual growth rate of fifty percent, while the die has been increasing in size by 10 to 15% per year. Therefore, a chip complexity increase of 60 to 80% per year is present.

An integrated circuit's cost is a function of many factors. The following equation shows these general parameters:

$$\text{IC cost} = \frac{(\text{die cost} + \text{testing cost} + \text{packing cost})}{(\text{final test yield})} \quad (1)$$

With increasing circuit complexity and a relatively constant number of I/O pins, test complexity is growing. Test complexity can be converted into the costs associated with the testing process. There are several facets to this cost, such as the cost of test pattern generation, fault simulation and the generation of fault location information, test equipment and the

cost related to the testing process itself, namely the time required to detect and/or isolate a fault [1].

In order to reduce the test cost, some design methods, known as "design for testability", have been suggested [2]. Usually, these methods introduce design guidelines, which, if followed, make testing of the circuit much easier. The Built-In-Self-Test (BIST) and the Level Sensitive Scan Design (LSSD) are two well-known methods for making circuits readily testable [3,4]. These methods offer means and techniques for circuit design at the gate level. Not much work has been done to improve the testability at the transistor level.

Usually, errors in the circuits are due to a limited number of specific faults. It is essential to assume that these faults occur individually. These faults at the gate level are the Stuck At 1 (SA1) and the Stuck At 0 (SA0). For the transistors, the faults are the Stuck OPen (SOP) and the Stuck At Closed (SAC) conditions. The interconnections also have their own special fault, which is the short circuit between any two nodes [5,6].

To test the circuit, a single fault is considered at some point in the circuit. Then, a test vector is found for that specific fault. This test vector should create an output that is functionally dependent on the existence of the fault. The process of fault detection consists of "fault propagation" and "value justification" sub-processes. The goal of fault propagation is to guide the faulty value from an inner point to an accessible output port. The "value justification" is

*. Corresponding Author, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, I.R. Iran.

1. Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, I.R. Iran.

referred to as the process of assigning proper values to the primary inputs, in order to make the propagation process possible. These process are dependent on the complexity of the network connecting the gates, as well as the type of the gates that are present in the network. Therefore, in devising an algorithm, both gates and network should be considered [7].

Testing sequential logic is considerably more difficult, compared to testing combinational circuits. One of the factors that complicate the task of creating tests is the presence of memory and circuit delay [8]. In these circuits, signals must not only be logically correct, but, must also occur at the right time in respect to other signals. A number of algorithms have been suggested and used for fault modeling since the advent of sequential circuits. Among these algorithms, Iterative Test Generation (ITG) is still considered a popular method [8].

A UNIVERSAL GATE FOR COMPLETE TESTABILITY

The objective for the design of a testable gate is to make the circuit dependent on the internal structure of the gate and independent of the interconnection network. An example of a testable gate is shown in Figure 1.

In Figure 1, a structure is shown which, based on what the select-lines are set at, operates as an AND, OR, NAND or NOR gate [9]. This structure could be used to construct any digital circuit with whatever complexity the circuit might have. Figure 2 shows the design of a basic sequential circuit, namely a D flip/flop. The circuit is made entirely by standard

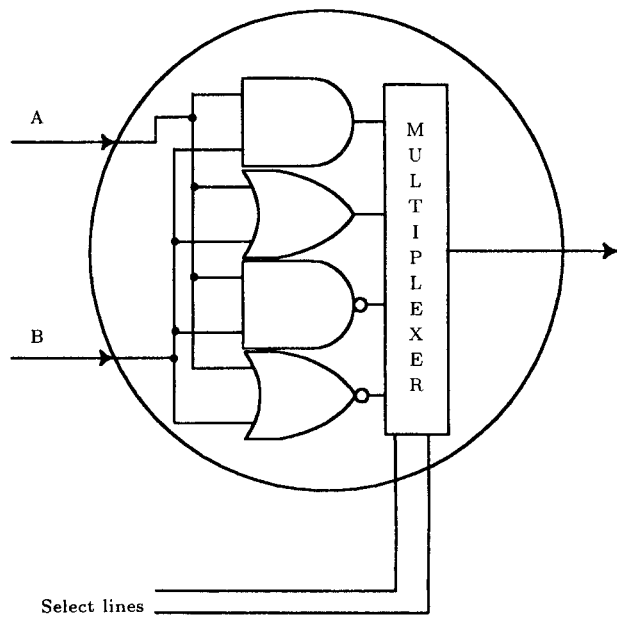


Figure 1. Block diagram of a testable gate.

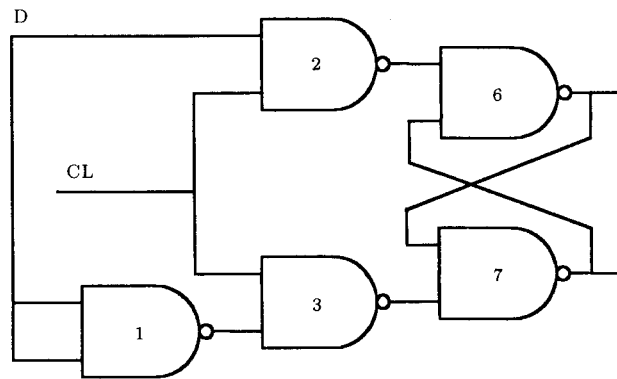


Figure 2. A clocked D latch circuit.

NAND gates.

Now, using the testable structure of Figure 1, the NAND gates could be replaced by the testable blocks. If the select lines of all of the gates are connected together and are set to the correct value, then the testable blocks will operate as NAND gates. Figure 3 shows the implementation of the clocked D latch with testable gates.

When the circuit is intended to operate in the normal mode, the select lines are set in such a manner that all of the gates operate as NANDs. In the case where all of the gates are changed to AND gates and the inputs are set to 1, then, regardless of the interconnection of the gates, the outputs will be 1. The presence of any SA0 fault causes a change in the output and, hence, the fault is detectable. By the same token, if the gates are changed to OR gates and all of the inputs are set to zero, then, any SA1 fault will be detectable [10]. Even though the idea of such a testable gate seems attractive, the basic element of Figure 1 is forbiddingly complex. The structure in Figure 3 shows that, rather than five NAND gates, more than thirty logic gates are used. Even if one omits the NOR gate of Figure 1, a 500% increase in complexity still seems to

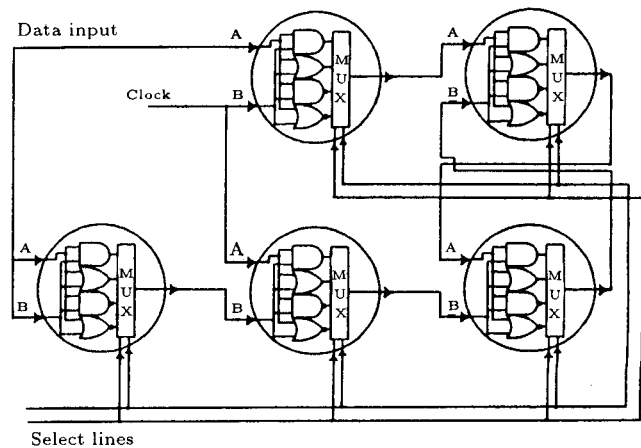


Figure 3. A clocked D latch circuit built by the testable elements.

have occurred. Here, it is intended to offer a testable gate at the transistor level, which has the attributes of Figure 1 while reducing the discussed complexity overhead. Therefore, if one could design a NAND gate that is convertible to an AND or an OR, then, any combinational circuit with any degree of complexity could be tested with the application of just two test vectors.

Let us look at one implementation scheme for the testable block. The following expression is extracted from Figure 1, where $S1$ and $S0$ are the select-lines of the multiplexes.

$$Y = A\bar{S}1.\bar{S}0 + \bar{A}.S1.\bar{S}0 + B.\bar{S}1.\bar{S}0 + \bar{B}.S1.\bar{S}0 + \bar{B}S1\bar{S}0 + \bar{A}.\bar{B}.\bar{S}1.S0 + ABS1S0. \quad (2)$$

The above Boolean function could be implemented using a number of standard gates. When building the block using these gates, it is unavoidable to have some internal faults that are distinguishable from the terminal faults of the block. Hence, extra test vectors are required to test the internal nets of the block and this is in contrast to the original purpose of the block. Mercer mentioned this bottleneck, but no structure has been suggested to circumvent the difficulty [9]. Any structure that is built using the standard gates will present the same problem. In this paper, two transistor structures that satisfy both the Karnaugh map and the testability requirements are presented [11].

THE PROPOSED TESTABLE CMOS GATE

To construct any digital circuits using CMOS technology, one requires access to the NAND gates. According to the previously given definitions, what one needs for a testable gate is the ability of the NAND to change, in the test mode, to an AND or an OR gate. The following configuration can satisfy these requirements [11].

A NAND gate is built using two pull up PMOS and two pull down NMOS transistors. This is shown in Figure 4 by transistors Q9 to Q12.

In order to convert the gate to an AND gate, it is required to invert the output of the Nand gate. If the output of the gate, which is the signal at the drain of Q12, is routed through an inverter, then, the signal corresponds to an AND function. The inverter at the output is built by Q13 and Q14. Hence, if the control signal $S1$ is zero, Q15 is turned on, Q16 is off and the output of the NAND is inverted. Now, if the inputs of a NAND gate are inverted, then, the output of the gate represents the OR function. In the authors' gate, the activation of $S0$ causes the inversion of both inputs. Therefore, setting $S1S0$ to 11 causes the gate to operate normally as a NAND. To test the circuit for SA0s, the control lines, $S1S0$, should be set to 01

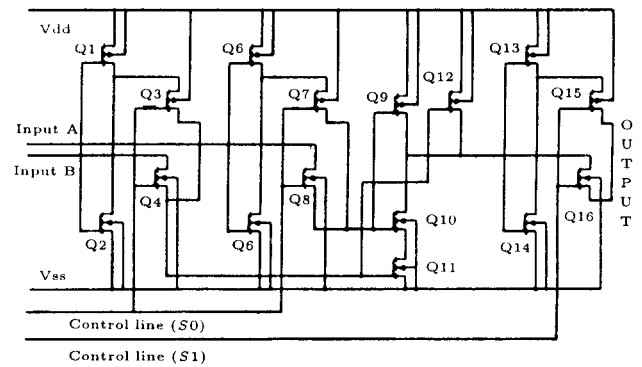


Figure 4. The proposed CMOS universal logic gate (circuit A).

transforming the gate to an AND. In order to change the circuit to an OR gate and testing the SA1 faults, the control signals, $S1S0$, are to be set to 10. The proposed gate also has the capability of performing as a NOR gate by selecting $S1S0$ as 00. This gives complete universality to the gate and permits the design of any digital circuit.

Functionality of the circuit was verified by SPICE simulation. A number of different cases for varying inputs and constant control signals were performed. The results for one of the runs are shown in Figure 5. In this case, input A was driven by a 45 kHz square wave, while input B was connected to a 100 kHz signal. The control line, $S1$, was connected to a 22 kHz source while the other control line was grounded. It can be seen that the circuit is operating as a NOR gate in the first 23 μ seconds and then operates as an OR gate.

In evaluating the proposed circuit, a number of drawbacks can be noticed. The required real estate for implementing this gate, deduced by a simple transistor count, is 300% more than that of a static CMOS NAND gate. The other important factor in evaluation of a logic gate is its propagation delay. This parameter is completely dependent on fabrication technology. Factors such as the size of the transistors (L and W), ohmic resistance of the terminals, oxide thickness and junction capacitances are some examples of the fabrication-dependent parameters that effect the propagation delay. Nevertheless, an effort was made to compare a simple NAND gate's delay with that of circuit A. Both circuits were built using the same type of transistors having the same set of parameters. Figure 6 shows the results of a NAND circuit simulation.

In getting the results of Figure 6, one of the inputs of a NAND gate was kept at logic 1 and a 25 MHz square wave was connected to the other input. A 14 ns delay was observed for this setup. Figure 7 shows the results for determining the delay of circuit A. In order to come up with these results,

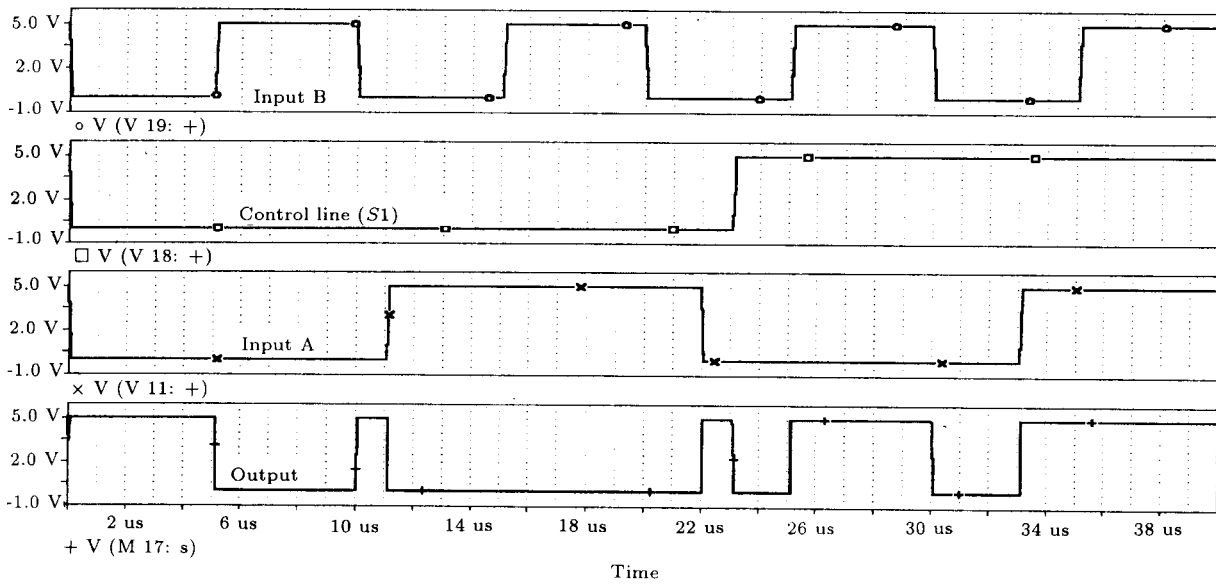


Figure 5. SPICE simulation of the proposed circuit A.

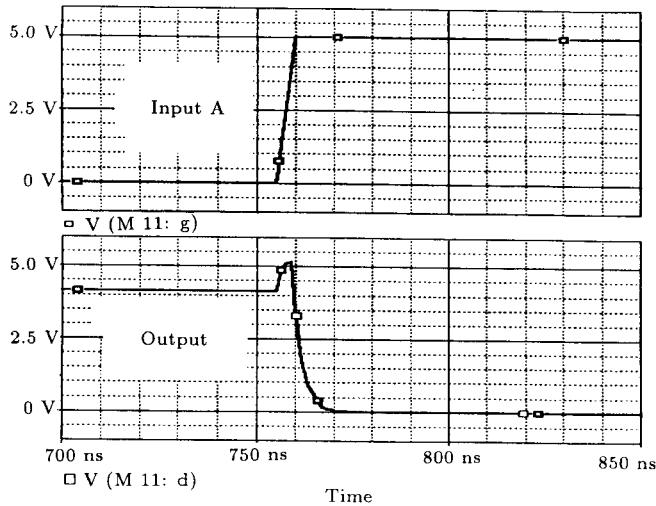


Figure 6. Propagation delay calculation for a NAND gate.

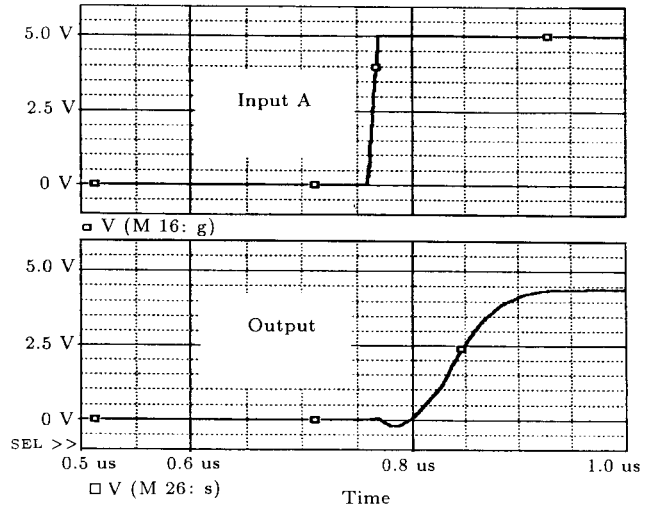


Figure 7. Computation of the propagation delay in the proposed circuit A.

one of the inputs of circuit A was kept at logic 1 and a 25 MHz square wave was connected to the other input. The select lines were setup so that the circuit operated as NAND. The computed worst-case delay was 38 ns. Therefore, the delay of the proposed circuit is 2.7 times that of a simple NAND gate.

Transistors Q15 and Q16 are responsible for taking the signal through the final stage of the circuit. This final stage is a NOT gate formed by Q13 and Q14. The signals at the input or output of this NOT are "strong" 1's and 0's. However, the signal at the source of Q15 and Q16 has to stay one threshold voltage (v_t) above V_{ss} and the same amount below V_{dd} . This is a requirement to keep these transistors operating. Therefore, the final output signal of the

circuit will not be a strong 1 or 0. To alleviate this problem, special pass transistors with a small threshold voltage can be used for Q15 and Q16. Another solution is to use transmission gates, which are a pair of NMOS/PMOS transistors. This solution is not recommended, since it increases the number of the transistors to 18. A third solution can be considered by connecting $(V_{dd}+v_t)$ or $(V_{ss}-v_t)$ voltages to the S1 control signal. This, too, has the disadvantage of requiring different voltage sources for one circuit, which, nowadays, is not practiced.

TEST OF SEQUENTIAL CIRCUITS

The basic idea presented in this paper is applicable to combinational circuits. An algorithm to initialize

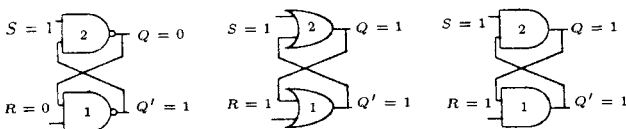


Figure 8. SA0 test of an RS flip flop with NAND gates.

and test the sequential elements of any logic circuit is now presented. Four test patterns are required to detect all possible single “stuck-at” faults. Two of these patterns initialize the states of all memory structures and the other two patterns detect the faults as previously discussed.

An example is shown in Figure 8. Let it be assumed that an RS latch with NAND gates is to be tested. Regardless of the state of the latch, the inputs are set to 11 while converting the gates to OR. The output of the good circuit will definitely be one. Now the gates are switched to AND mode. Obviously, the circuit will be tested for any SA0 fault.

Therefore, to test the latch for SA1 fault, one has to apply 00 at the inputs and change the gates to AND mode. Then, the outputs will be 0. The next pattern should be 00 at the inputs and switching the gates to OR. This procedure is shown in Table 1. It is shown that any sequential circuit that is built by the proposed universal gates, regardless of its complexity, is tested with a maximum of 4 test vectors.

The proposed algorithm, in most cases, is advantageous to ITG [8] or any other ATPG (Automatic Test Pattern Generator) algorithm. This is due to the fixed and small number of test patterns required. It may seldom happen that ITG generates less than four test patterns to test a sequential circuit. All of the sequential algorithms for a moderate size circuit require a sequence of a large number of patterns. Even for initializing a sequential circuit, a relatively long sequence (distinguishing, homing or synchronizing) is required.

In order to abridge the number of transistors in “circuit A” the circuit of Figure 9 is propounded. The circuit’s design is based on CMOS domino logic. A single-phase clock that is connected to the gate of Q1 and Q5 controls the operation of the circuit. When the clock is low, Q5 is off and Q1 is on. Under these circumstances, regardless of the values of the circuit’s inputs, the drain of Q1 is held high. Via the control

Table 1. Required steps to test any sequential circuit.

Test for SA0			Test for SA1		
Step	Inputs	S1, S0	Step	Inputs	S1, S0
1	111.....1	10	1	000.....0	01
2	111.....1	01	2	000.....0	10

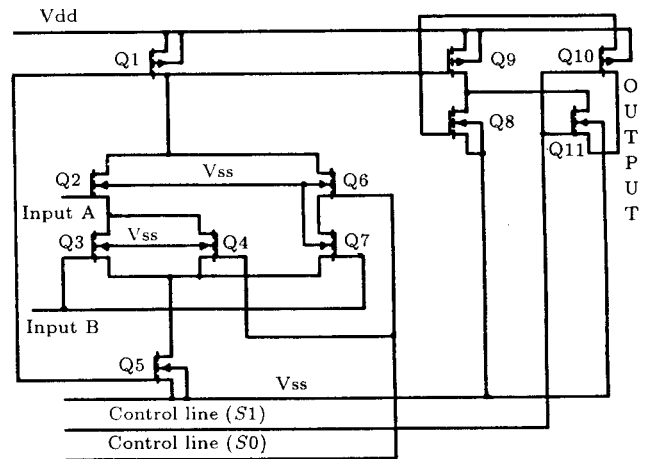


Figure 9. The proposed CMOS domino logic gate (circuit B).

line, S1, the function of the logic gate can be altered between a NAND and a NOR, taking the gate’s output as the drain of Q10. While the clock is high, depending on the values at A and B, the output can be either 1 or 0. The select line, S0, controls the inverter at the output of the gate. Using S1 and S0, all four logical functions can be achieved.

The circuit’s operation was tested by SPICE simulation. The simulation was performed for a situation where S1 was at logic one and S0 was connected to a 250 kHz square wave. To test the operation of the gate as AND and NAND, input A was kept high and a square-wave signal with 500 kHz frequency was applied to B. A clock frequency of 1 MHz was connected to the Phi input of the circuit. The simulation results are shown in Figure 10.

Using the same transistors as circuit A, the propagation delay of circuit B was tested. Figure 11 shows the result of these measurements. The output of circuit B demonstrated a worst-case delay of 43 ns with respect to the clock input. This is 3 times that of a simple NAND gate. This longer delay is because of three NMOS transistors that come in series. Also, the real estate overhead for this circuit is 175% more silicon compared to a static CMOS NAND.

CONCLUSION

In this paper, two specific structures are offered for a universally testable gate. Designing digital circuits using either of these gates can be categorized under the DFT category. Recent work on the testable cells has tried to increase testability but is far short of complete testing with just two vectors. Also, an algorithm is proposed to initialize and test any sequential circuit for any single “stuck-at” fault with the application of just four test patterns.

In the proposed circuit A, potential transitional glitches during the test sequences and high silicon real

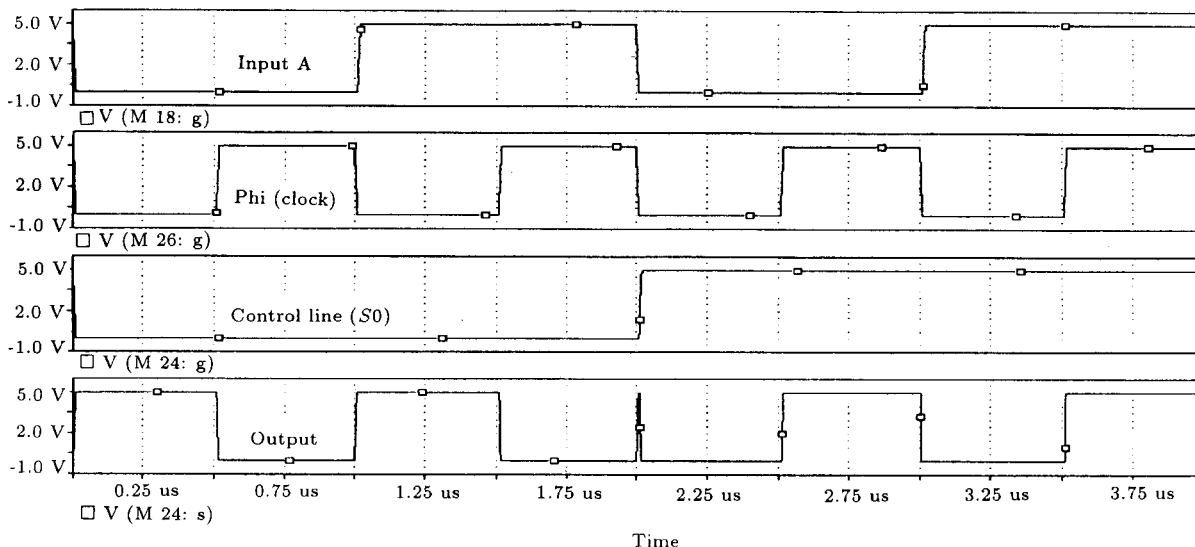


Figure 10. SPICE simulation of the proposed circuit B.

estate are the two drawbacks of the design. The proposed circuit B, reduces the real estate but the potential of transitional spikes is still present. Glitches only occur at the beginning of the test mode and are very tolerable. The main shortcoming in circuit B is its relatively protracted delay. It should be noticed that the digital circuits are not solely employed in the computational circuitry. Many control circuits use digital gates and are required to be highly reliable. The measured delays of circuit B are not intolerable for these control circuits.

In reference to Equation 1, using the proposed circuits, the test cost becomes minimal. This is because there will be no design for testability, no cost for designing test vectors and, finally the application of only two vectors requires minimal time and cost. Obviously, these reductions in the test cost come with an increase in the die cost. The issue of real estate, as in any design, is an issue of trade-offs. In complex vital

control circuits, where the importance of testability increases, the required reliability may surpass the real estate and speed factors.

REFERENCES

1. Arguelles, J. and Bracho, S. "A design for test proposal for improving dynamic current testing reliability on regenerative sense amplifiers", *Digest of Papers of the IEEE International Workshop on IDDQ Testing*, pp 29-32 (1997).
2. Heidel, D., Sang Dhong, Hofstee, P., Immediato, M., Nowka, K., Silberman, J. and Stawiasz, K. "High speed serializing/de-serializing design-for-test method for evaluating a 1 GHz microprocessor", *Proceedings of the 16th IEEE VLSI Test Symposium*, pp 234-238 (1998).
3. Kinoshita, K. "Design for testability of sequential circuits", *Digest of Technical Papers of the Symposium on VLSI Circuits*, pp 5-8 (1993).
4. Chakradhar, S.T., Agrawal, V.D. and Bushnell, M.L. "Energy minimization and design for testability (combinational circuits)", *Journal of Electronic Testing: Theory and Applications*, 5(1), pp 57-66 (1994).
5. Sparmann, U., Muller, H. and Reddy, S.M. "Universal delay test sets for logic networks", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(2), pp 156-166 (June 1999).
6. Cheng, K.T., Devadas, S. and Keutzer, K. "Delay-fault test generation and synthesis for testability under a standard scan design methodology", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(8), pp 1217-31 (1993).
7. Al-Asaad, H. and Hayes, J.P. "Design verification via simulation and automatic test pattern generation", *Digest of Technical Papers of 1995 IEEE/ACM In-*

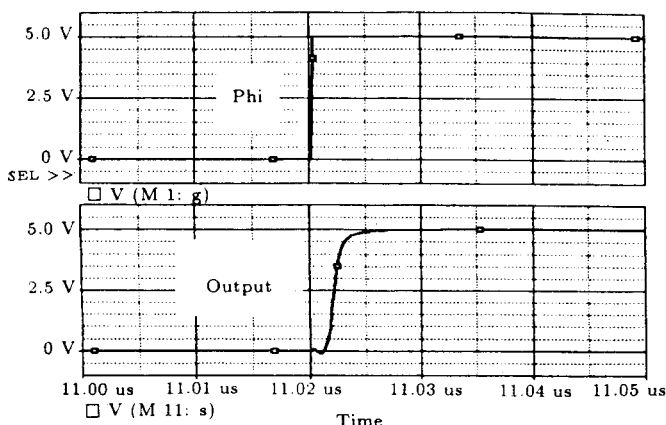


Figure 11. Propagation delay calculation for the proposed circuit B.

- ternational Conference on Computer-Aided Design*, pp 174-80 (1995).
8. Samavi, S. "An algorithm for fault modeling in sequential logic circuits", *Proceedings of the Third Iranian Conference on Electrical Engineering*, Tehran, Iran (1995).
 9. Mercer, M.R. "Logic elements for universally testable circuits", *Proceeding of the IEEE International Test Conference*, pp 443-448 (1986).
 10. McCluskey, E.J. and Clegg, F.W. "Fault equivalence in combinational logic networks", *IEEE Transactions on Computers*, pp 1286-1293 (1971).
 11. Samavi, S. and Ferdowsi, A. "Test of sequential circuits using universal CMOS logic gates", *Proceedings of the Sixth Iranian Conference on Electrical Engineering*, **3**, pp 29-34 (1996).