Variable Input Delay CMOS Logic for Low Power Design[*]

Tezaswi Raja
Transmeta Corp.
Santa Clara, CA 95054, USA
traja@transmeta.com

Vishwani D. Agrawal
Auburn University, Dept. of ECE
Auburn, AL 36849, USA
vagrawal@eng.auburn.edu

Michael L. Bushnell
Rutgers University, Dept. of ECE
Piscataway, NJ 08854, USA
bushnell@caip.rutgers.edu

Abstract

Modern digital circuits consist of logic gates implemented in the complementary metal oxide semiconductor (CMOS) technology. The time taken for a logic gate output to change after one or more inputs have changed is called the delay of the gate. A conventional CMOS gate is designed to have the same input to output delay irrespective of which input caused the output to change. We propose a new gate design that has different delays along various input to output paths within the gate. This is accomplished by inserting selectively sized "permanently on" series transistors at the inputs of the logic gate. We demonstrate the use of the variable input delay CMOS gates for a totally glitch-free minimum dynamic power implementation of a digital circuit. Applying a previously described linear programming method to the c7552 benchmark circuit, we obtained a power saving of 58% over an unoptimized design. This power consumption was 18% lower than that for an alternative low power design using conventional CMOS gates. All circuits had the same overall delay. Since the overall delay was not allowed to increase, the glitch elimination with conventional gates required insertion of delay buffers on non-critical paths. The use of the variable input delay gates drastically reduced the required number of delay buffers.

1 Introduction

There are many ways of combining transistors to perform the logic functions such as NOT, NAND and NOR. We will describe the CMOS design style which is the most prominent in current day technologies. CMOS gates are constructed by a combination of MOSFETs to realize a logic function. But a MOSFET is not an ideal switch. When *open* it provides a large but finite resistance between its source and drain terminals. When *closed* it provides a small non-zero resistance. For a CMOS gate the output signal change follows the input change with a certain delay. First, the closing and opening of MOSFETs in the gate depends upon the slope of input signals. Then, the output signal change requires charging or discharging of the output capacitance through a low resistance path provided by the "on" MOSFETs.

Gate Delay is the time taken for a signal at the output of a gate to reach 50% of *Vdd* (logic 1 level) after the signal at the input of the gate has reached 50% of *Vdd*.

Gate delay is a function of the amount of resistance and capacitance in the current path. A MOSFET when closed offers a finite resistance R_{on} that is a function of the width and length of the device. Since gate delay is given by $R_{on} \times C_L$(where C_L is the load capacitance) it can be varied by changing the width and length of the transistor [15, 17]. For example, a NAND gate output rises due to current flow in its *p*FETs. Hence, the delay of the NAND gate for a rising transition, can be altered by changing the sizes of *p*FETs. To increase the delay, we increase the resistance of a transistor by increasing its length. Similarly the output delay for a falling transition can be varied by changing the length of the *n*FETs. The delay can also be reduced by increasing the width of the transistor. The delay is effectively changed by manipulating the width and length of the transistors in the gate. Note that it is possible only to manipulate the overall delay of a gate but not the individual delays along different paths through it. For instance, the delay of a gate when one input transitions cannot be independently controlled without altering the delay when the other input transitions. These delays are inter-related. This is a drawback in some applications.

Every signal transition consumes a finite amount of energy. For the correct functioning of the logic circuit, every signal net needs to transition at most one time in one clock cycle. But in reality, the gate outputs transition more than once and these unnecessary transitions are called *glitches*. These transitions consume energy and are quite unnecessary for the correct functioning of the circuit. Glitch power consumption can be as much as 40% (or higher) as compared to the overall power consumption and it is advantageous to eliminate the glitches in the circuit as power consumption is critical in today's chips. Glitches arise due to the differences in the arrival times of signal transitions at the inputs of the gate.

Differential Delay is the maximum difference in the signal arrival times at different inputs of a multi-input gate.

Consider the circuit shown in Figure 1. The signal arrival at the lower input of the NAND gate is always

[*] Research supported in part by the National Science Foundation Grant CCR-9988239

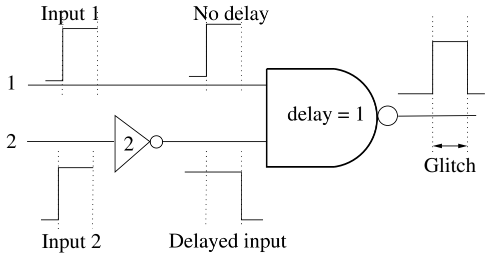


Figure 1: A circuit showing the formation of glitches. The inverter has a delay of 2 and the NAND gate has a delay 1. Due to differing arrival times at the inputs of the NAND gate, the output produces a glitch consisting of two transitions.

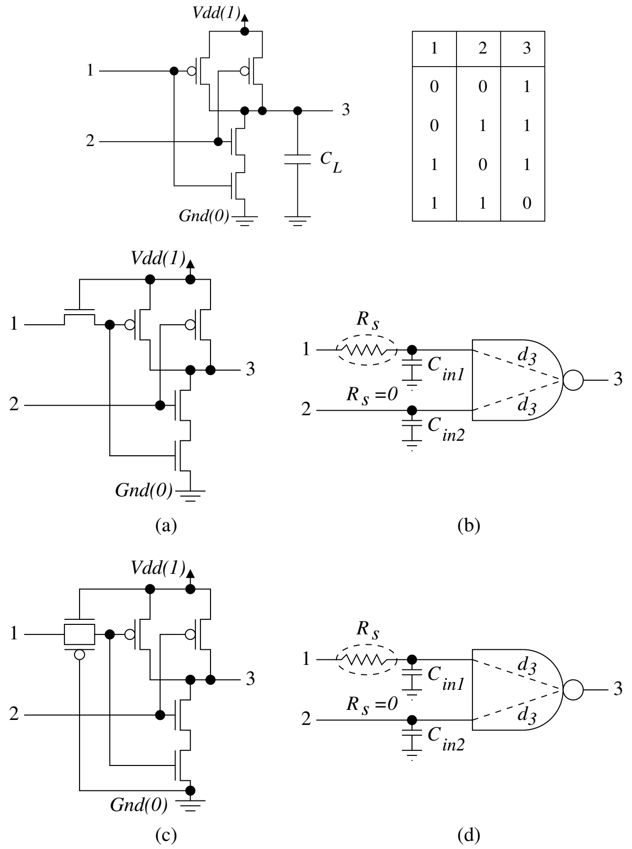
2 time units later than the signal arrival at the upper input due to the inverter in its path. Thus the differential delay at the NAND gate is 2. This differential delay makes the output of NAND gate transition twice when in reality, it should have no logic transition at all. These extra transitions are the glitches and they waste energy.

There have been many techniques proposed to eliminate the glitches. In *delay balancing*, the inputs are made to arrive at the same time by inserting extra delay buffers on selected paths [11–13, 18, 23]. In *hazard filtering*, the gate delay is made greater than the differential delay at the inputs of the gate to filter the glitch [1]. In *gate sizing*, every gate is assumed to be an equivalent inverter [3–8]. *Transistor sizing* treats every transistor's size as a variable and tries to find a glitch-free design [14, 16, 24, 25, 27, 28]. However, these techniques are either greedy approaches or have non-linear convergence problems [19].

Some techniques use *linear programming* where the gate delays are treated as variables and the optimum delays are found by solving a *linear program* (LP) [2, 19, 21]. The problem with this technique is that it inserts delay buffers in the circuit. These extra inserted elements also consume power themselves and hence reduce the achievable power savings. In all of the above techniques, the problem of buffer insertion arises due to the conventional gate design. The conventional CMOS gates have a single delay, no matter which input of the gate causes the transition. A new technique by Raja *et al.* proposed a LP technique using a new gate delay model, where the input delay of the gate can also be varied [22]. This makes the gate delay different for different input-output paths through the gate. The advantage of this gate model is that the glitches can be completely eliminated in the circuit without the insertion of any delay buffers, thus achieving more power savings. A novel implementation of this technique is the focus of this paper.

2 Proposed Gate Design

As described above, it is advantageous to design a gate with differing delays along different input-output paths of the gate. We define such a gate as a *variable input-delay gate.* In this section, we propose a transistor level implementation of the gate and its characteristics [20]. Consider a two-input NAND gate shown in Figure 2. Suppose, the delay required along the path 1-3 is 2 units and 2-3 is 1 unit.



1	2	3
0	0	1
0	1	1
1	0	1
1	1	0

(a) (b)

(c) (d)

Figure 2: Schematic of the proposed variable input delay gate: A conventional 2-input CMOS NAND gate characterized by a single output delay (top), and two ways of varying input delays by always-on nMOS pass transistor (center) and by always-on CMOS transmission gate (bottom).

$$\begin{aligned} d_{1\rightarrow 3} &= R_{on} \times C_{in1} + d_3 \\ d_{2\rightarrow 3} &= R_{on} \times C_{in2} + d_3 \end{aligned}$$

where C_{in} is the input gate capacitance seen at the inputs of the gate and R_{on} is the series resistance of the ON transistor in the previous stage. A conventional CMOS gate (top figure) is characterized by a single delay normally assigned to the output. To control the delay along the different paths we examine four different implementations.

- **Input capacitance manipulation** is the technique by which C_{in1} is increased without altering C_{in2}. This is achieved by increasing the sizes of the transistors connected to input 1 such that $C_{in1} > C_{in2}$. Now the delays along different paths are:

$$d_{1\rightarrow 3} = R_{on} \times C_{in1} + d_3$$

$$
\begin{aligned}
d_{2\rightarrow 3} &= R_{on} \times C_{in2} \;+\; d_3 \\
d_{1\rightarrow 3} &> d_{2\rightarrow 3}
\end{aligned}
$$

The problem with this implementation is that ON resistances of transistors in the series path are interrelated and hence the output delay is also altered. The formulation becomes non-linear.

- **Resistance with a single nMOS pass transistor** can be added in series to the path in which extra delay is desired. This scheme is shown in Figures 2(a) and (b). This nMOS transistor is always ON and hence adds a series resistance R_s to the path 1 → 3. Now the delays are:

$$
\begin{aligned}
d_{2\rightarrow 3} &= R_{on} \times C_{in2} \;+\; d_3 \\
d_{1\rightarrow 3} &= (R_{on} + R_s) \times C_{in1} \;+\; d_3 \\
d_{1\rightarrow 3} &> d_{2\rightarrow 3}
\end{aligned}
$$

  The resistance R_s can be controlled by increasing the size of the nMOS transistor. The delay along the path 1 → 3 can be controlled independent of the delay along path 2 → 3. Hence the gate has different delays along different input-output paths through it. The disadvantage of this design is that the nMOS pass transistor degrades the signal when it passes a logic 1. This causes the transistors in the next stage to have a higher leakage current. This aspect is further discussed in the next subsection.

- **Resistance with a CMOS pass transistor** can be added to introduce the extra resistance in the path as shown in Figures 2(c) and (d). The principle is the same as adding a single nFET, but the CMOS pass transistor contains both nMOS and pMOS transistors that are always ON. This does not degrade the signal but has the disadvantage that it adds an additional transistor.

- **Resistance with a feedthrough resistive cell** is a technique of adding the resistance using a *polysilicon* serpentine resistor overlaid with *silicide blocking.* This is the standard way of creating a resistance in analog layout design but can be used for this purpose. The advantage of using these cells is the continuous controllability of resistance rather than the discrete control provided by transistors [26].

2.1 Design Issues

There are several design issues regarding the variable input delay gate design. The delay along a path can be changed by changing the series resistance. R_s is a function of the length of the transistor/transmission gate and hence the delay along the line can be altered by changing the length of the extra transistor/transmission gate.

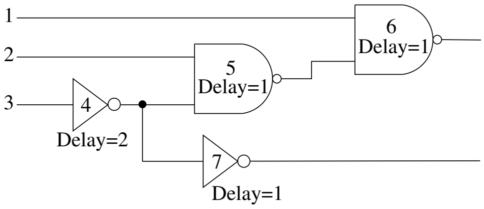


Figure 3: An example circuit.

- This transistor cannot be infinitely large as this would increase the voltage drop across the transistor and cause signal integrity issues at the output of the gate. Hence there is a realistic limit to the length of the transistor added and this determines the maximum differential delay that can be added. Raja *et al.* describe this as the *gate differential delay upper bound* u_b in their low power design [22]. This parameter u_b is related to the technology the gate is implemented in and hence is called the *feasibility condition.* Our calculations have predicted and measured a u_b of 10 units for the 0.25μ fabrication process [20].

- If the voltage drop across the transistor is too large, it does not drive the gate transistors in the fanout into cut-off. This increases the leakage from the supply to the ground through the gate transistors as they are not completely off. This problem can be alleviated by using a CMOS transmission gate instead of a single transistor. The effect of increased leakage is shown in the results section.

- The placement of the series transistor with respect to the routing capacitance also needs to be examined. If the routing capacitance is small it does not matter where the transistor is placed in the path. But if the routing capacitance is large, then the delay at the input of the gate changes as the transistor is moved along the path as it sees a different capacitance at every stage. We have inserted the transistor at the end of the routing path in our designs [20].

3 Results

In this section we present an application of the new gate design in implementing custom circuits for minimum dynamic power.

An Unoptimized Example Circuit. Consider the simple example circuit of Figure 3. Assume that the delays of all gates are the minimum allowed by the technology. We observe that the differential delay, at gates 5 and 6, exceeds the inertial delay and we expect these gates to glitch. The circuit was simulated for rising signals at all three inputs. The simulation was done using the *Spectre* analog simulator from *Cadence.* As expected, gates 5 and 6 transition 2 and 3 times, respectively. These are the glitches we wish to eliminate in the following designs.

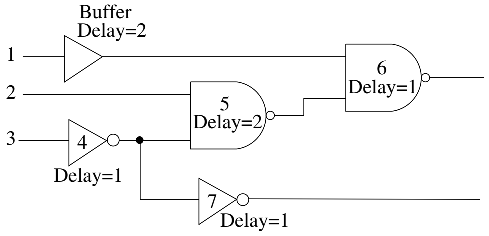


Figure 4: Optimized example circuit with a delay buffer (two inverters).

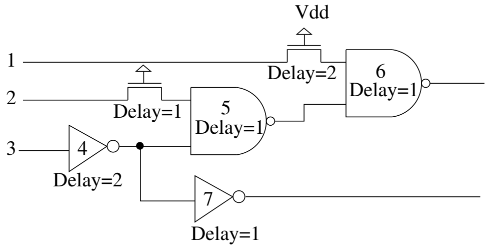


Figure 5: Optimized example circuit with the proposed gate.

Buffer Optimized Circuit. The buffer optimization using conventional gates requires the use of one buffer for the circuit to operate at the same speed [2, 21]. The optimized circuit with the buffer is shown in Figure 4. It is implemented using two CMOS inverters and has an overall delay of 2 units. The buffer optimized circuit was simulated for the same vector-pair as the unoptimized circuit. As expected, the optimization eliminated all glitches.

Low-Power Design with Proposed Gate. When variable input-delay gates are used the optimized circuit is shown in Figure 5. We have used the single *n*MOS transistor implementation here but any of the proposed designs could have been used. The circuit-level simulation of the two vectors, 000 and 111, for the three circuits is shown in Figure 6. The glitches at the outputs of gates 5 and 6 are eliminated in the optimized designs. However, the buffer optimization requires that the transition of input 1 should pass through the two inverters of the buffer. This increases the total transitions in that circuit as shown in Table 1.

3.1 Energy Consumption

During the simulation for the three circuits described above, we measured the supply current for the given input vectors and computed the energy. The results are shown in Table 1. The simulations were done with *Spectre* analog simulator from *Cadence* [10]. As recorded in the table, the unoptimized circuit consumes 800fJ, the buffer optimized circuit consumes 550fJ and the new gate optimized circuit consumes 300fJ. Thus the energy savings of the new design are 62.5% with respect to the unoptimized circuit. The new gate design achieves 36.8% more savings than the buffer optimized design with respect to the unoptimized circuit. The total power obtained from the simulator includes the short circuit and leakage components as well. However, for the 0.25μ CMOS technology used the dynamic power dominates as discussed in the next subsection. Table 1 also shows a good correlation between the reduction in the number of transitions and power saving.

3.2 Leakage Current

The introduction of an *n*MOS pass transistor degrades the signal at the gates of the transistors. This increases the leakage current of the circuit and may even drive the transistors out of cut-off. The current flowing in the steady state is called the *quiescent current* (I_{DDQ}) and is due to the leakage through off transistors. The quiescent current is a function of the input vectors at the PIs of the circuit. To analyze the relative effect, we simulated circuits with two input vectors and let the circuit settle for a long time after each vector. Three circuits were simulated for leakage. These were the unoptimized circuit of Figure 3, the optimized circuit of Figure 5, and another optimized circuit obtained by replacing the *n*MOS pass transistors in Figure 5 with CMOS transmission gates. The leakage currents for the vector 000 showed no change for the three circuits as in this state the *n*MOS transistors are passing logic 0, which is not degraded (Table 1). For vector 111, however, there was an increase of 0.45% in leakage due to the *n*MOS pass transistors. The circuit with the CMOS transmission gates had an increase of only 0.2%. This increase is not due to the degradation of the signal but is due to the leakage path added from Vdd to Gnd through the sidewall capacitance. This is a very minor increase for the 0.25μm fabrication technology but further analysis needs to be done for more recent technologies.

3.3 Benchmark Circuits

We optimized several ISCAS'85 benchmark circuits for dynamic power. The results in Table 2 compare the designs done with the new variable-input delay gates to original versions of circuits, and to circuits optimized using conventional gates [19, 21]. For each method, two optimized designs were created, one, where no increase in the overall delay (*maxdelay*) was permitted and, second, where the overall delay was allowed to increase to twice that of the original design. The original designs were optimized not for power but for speed in the given 0.25μ CMOS technology.

For each circuit, first an original version (not optimized for glitch removal) was created as a reference. This version used the fastest gates available in our 0.25μ CMOS technology. These gates have larger transistors and typically consume more power. This design functions somewhat similar to a unit-delay logic circuit, which is known to consume more power [23]. Power consumption was estimated by an event-driven simulator, which assumed that each gate has the same delay and that the power consumed per signal tran-

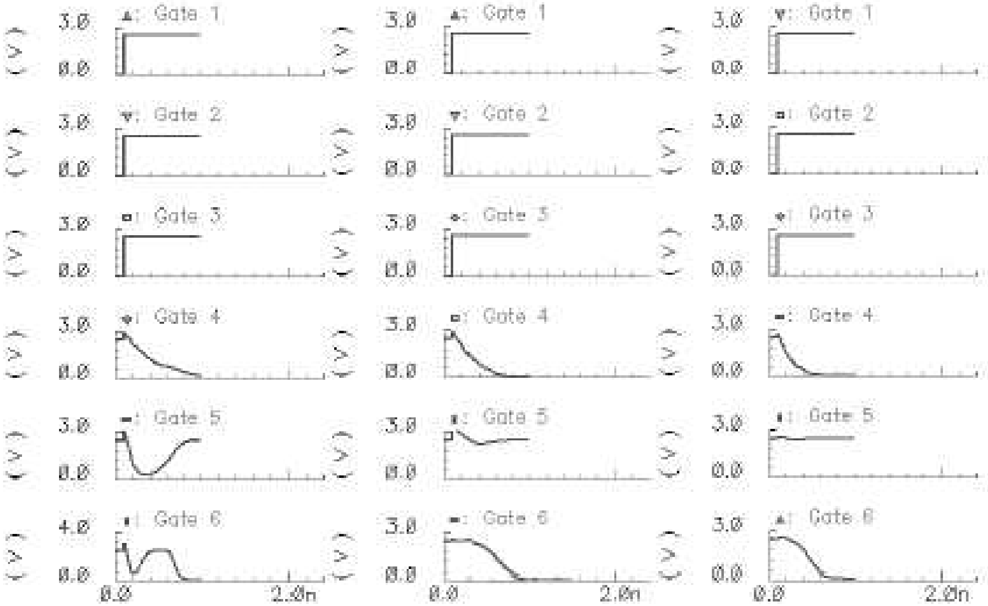


Figure 6: Circuit simulation of vectors 000 → 111 for (left to right) circuits of Figures 3, 4 and 5.

Table 1: Simulation of the three designs of the example circuit for input 000 → 111.

Circuit	Logic activity		Energy consumed		Leakage I_{DDQ}	
	Gate transitions	Reduction	Total	Reduction	Vector 000	Vector 111
Figure 3	8	0.0%	800fJ	0.0%	38.1pA	60.6pA
Figure 4	5	37.5%	550fJ	31.3%	–	–
Figure 5	3	62.5%	300fJ	62.5%	38.1pA	60.9pA
Circuit of Figure 5 with CMOS Transmission gates					38.1pA	60.7pA

sition is proportional to the number of fanouts. The simulator uses a glitch-filtering procedure [9]. Thus, whenever a new event is scheduled such that a previously scheduled event on the same signal is still pending, then both events are cancelled. Estimates of peak and average power were obtained for a set of vectors. These vectors were generated for a complete or almost complete stuck-at fault coverage. It is assumed that such vectors provide appreciable logic activity and hence a reasonable measure of power. In Table 2, the power of original circuits is normalized to unity and transistor counts for all circuits are given. Power estimation for all other designs (discussed below) was similar but used the delays obtained from the LP.

Next, we redesigned the circuits with variable-input delay gates. An LP determined the input and output delays for all gates under an input diffrential delay constraint of $u_b = 10$ (see Subsection 2.1). Each circuit was designed for two overall delays, $maxdelay =$ 1 and 2, respectively, normalized with respect to the corresponding reference design. Columns 4, 5 and 6 of Table 2 show the number of transistors added (see next paragraph) and the power consumption normalized with respect to the corresponding original design. To meet the *maxdelay* constraint, some circuits used delay buffers. But in most cases no buffers were required. In the linear program optimization, an upper bound (u_b) is used on the input differential delay that can be achieved. This upper bound is a technology parameter and is determined through actual design and simulation of gates. When the circuit topology requires very large differential delays, delay buffers must be used to satisfy the glitch removal conditions. The linear program, however, keeps the number of such buffers to a minimum. The circuit c6288 is a typical case where a large number of buffers were essential. Since each delay buffer has two inverters, which provide additional node capacitances to be charged and discharged during operation, extra power is consumed. The $maxdelay = 2$ design of c6288 did not require buffers and all glitch removal conditions were satisfied by the gate input delays.

The added transistors are mostly for the nMOS transmission gates inserted at gate inputs. As explained above some circuits needed a few delay buffers. Each buffer was implemented with four transistors (two inverters). Those transistors are included in the counts given in column 4 of the table. If the designs were to be done with CMOS transmission gates, the added transistor counts will double only for transmission gates and will remain unchanged for buffers.

The last three columns of Table 2 provide a comparison with an alternative method in which conventional CMOS gates are used. These gates were designed for

Table 2: Power consumption of custom designs of ISCAS'85 circuits estimated by logic simulation. The original designs are the highest speed designs in the 0.25μ CMOS technology used.

Circuit	*maxdelay*	Orig. design	Variable input-delay gate			Conv. CMOS gate [19, 21]		
		Norm. power=1	Added	Norm. power		Added	Norm. power	
		No. of Trans.	Trans.	Av.	Peak	Trans.	Av.	Peak
c432	1.0	784	291	0.69	0.66	380	0.72	0.67
	2.0	784	98	0.65	0.55	264	0.62	0.60
c499	1.0	1,364	105	0.86	0.84	192	0.91	0.87
	2.0	1,364	86	0.71	0.65	0	0.70	0.66
c880	1.0	1,802	174	0.58	0.45	248	0.68	0.54
	2.0	1,802	154	0.56	0.45	136	0.68	0.52
c1355	1.0	2,196	550	0.48	0.42	896	0.58	0.48
	2.0	2,196	410	0.44	0.39	768	0.57	0.48
c1908	1.0	3,878	206	0.56	0.46	876	0.69	0.59
	2.0	3,878	192	0.55	0.45	280	0.59	0.44
c2670	1.0	5,684	436	0.70	0.56	628	0.79	0.65
	2.0	5,684	380	0.69	0.57	140	0.71	0.58
c3540	1.0	7,822	677	0.57	0.46	956	0.64	0.44
	2.0	7,822	642	0.54	0.43	560	0.58	0.46
c5315	1.0	11,308	1,310	0.57	0.48	1,120	0.63	0.52
	2.0	11,308	1,361	0.55	0.46	684	0.60	0.45
c6288	1.0	10,112	2,854	0.91	0.87	1,176	0.40	0.36
	2.0	10,112	1,815	0.21	0.16	480	0.36	0.34
c7552	1.0	15,512	1,439	0.28	0.24	1,464	0.38	0.34
	2.0	15,512	1,406	0.27	0.24	444	0.36	0.32

almost equal input delays and are characterized by a single delay. The design here is also obtained by a linear program [19, 21], however, delay buffers are used in most cases. With the exception of a few circuits, most circuits consumed more power when compared to the variable input delay gate design. The numbers of added transistors in column 7 are due to the delay buffers, each requiring 4 transistors. Thus, c7552, required 366 buffers implemented with 1,464 transistors for the $maxdelay = 1$ design.

3.4 Chip Design and Total Power

We did the physical design of the ISCAS'85 benchmark circuit c7552. First, an "unoptimized" design was created. This circuit contained 3,827 gates and was implemented with 15,512 transistors. We used gates with smallest size transistors as compared to the fastest gates used in the "original" design of the previous subsection. The unoptimized circuit, therefore, is slower but consumes less power. Its physical layout was done by the *Cadence layout editor.* We redesigned the circuit using the proposed variable input delay gates and that design contained 1,435 nMOS transmission gates and one delay buffer, requiring 1,439 extra transistors. This design is the $maxdelay = 1$ version of c7552, shown in Table 2 (columns 4 to 6). A third design using the conventional CMOS gates (last three columns in Table 2) was also implemented. It required 366 delay buffers or 1,464 extra transistors added to the unoptimized version. All three designs were implemented in 0.25μ CMOS technology and worked at the same speed [20]. Two layouts shown in Figure 7 are for the unoptimized design and the variable-input delay gate design. The areas of these chips are $710\mu \times 710\mu$ and $760\mu \times 760\mu$, respectively.

Power consumption was evaluated in two ways. First, the logic siumlation method of the previous subsection was used with few differences. For the unoptimized circuit, gate delays were assumed to be proportional to fanouts instead of being the same, and the signal activity was weighted by the node capacitance extracted from the chip layout. The circuits were simulated for a set of 156 fault coverage test vectors. As shown in Table 3 the variable-input delay gate design saves 58% average and 66% peak power. In comparison with the conventional CMOS gate design using 366 delay buffers, the variable-input delay gate design consumed about 17% less average power.

These power savings, though appreciable, are lower than those estimated in Table 2. The reason for the discrepancy is that our "unoptimized" design uses the smallest gates and consumes less power as compared to the "original" design, which used the fastest gates. Indeed, the "original" design is faster than the "unoptimized" design.

A second evaluation of power was done with a circuit-level simulator. The results of instantaneous and average power measurements are shown in Figures 8 and 9. These results were obtained by the *Spectre* simulator from *Cadence* [10]. The measurement here includes all components of power, namely, dynamic, short-circuit and leakage. For simulation, node capacitances were extracted from the layouts. The cir-

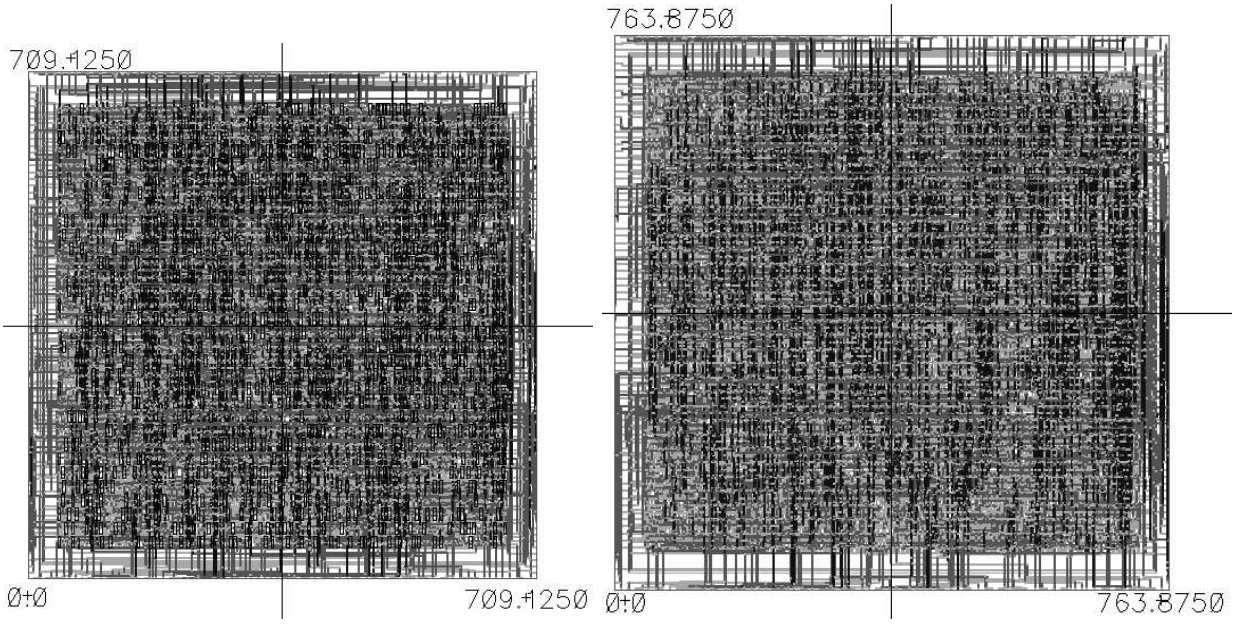


Figure 7: 0.25μ custom CMOS layouts of unoptimized (left) and optimized c7552 circuits.

Table 3: Power consumption of c7552 chips estimated by logic simulation.

Circuit	*maxdelay*	Unopt. design	Variable input-delay gate			Conv. CMOS gate [19, 21]		
		Norm. power=1	Added	Norm. power		Added	Norm. power	
		No. of Trans.	Trans.	Av.	Peak	Trans.	Av.	Peak
c7552	1.0	15,512	1,439	0.42	0.34	1,464	0.49	0.35

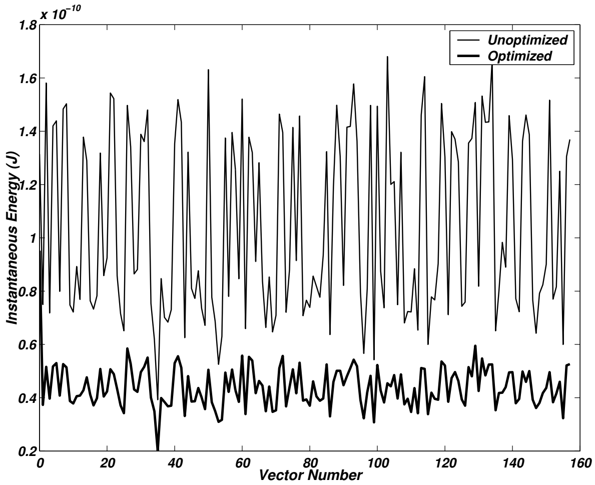


Figure 8: Instantaneous energy consumption in benchmark circuit c7552 for 156 vectors. A peak power saving of 68% over the unoptimized circuit is realized.

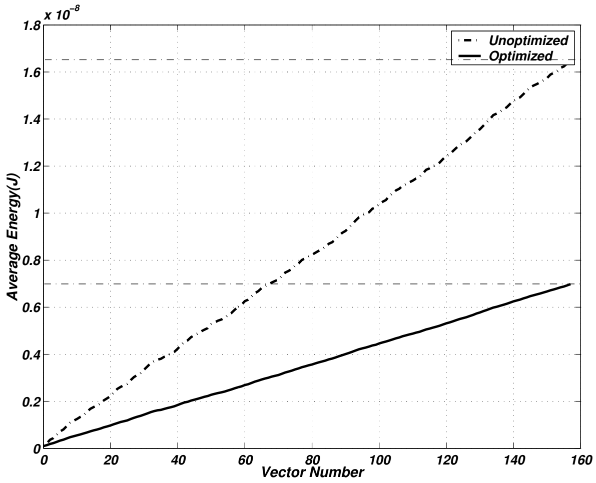


Figure 9: Average energy consumption results for benchmark circuit c7552 for 156 vectors. Results show an average energy savings of 58%

cuits were simulated for the same set of 156 vectors. These plots show a peak power saving of 68% and average power saving of 58%, which are very close to those obtained by logic simulation (Table 3). The circuit-level simulation thus confirms that the short-circuit and leakage power components were indeed negligible, as is expected for the 0.25μ CMOS technology, for both optimized and unoptimized circuits.

4 Conclusion

We have proposed a new variable input delay gate design, which has different delays along different input-output paths through the gate [20]. This new design has applications to low power design of digital CMOS circuits. Using the new gate design significant energy savings have been achieved. The delays of the gates tend to change due to process variations such as temperature, fabrication impurities, etc. These variations make the delay vary over a range rather than being a single static number. This can be accounted for in our technique during the linear program (LP) stage, where the constraints can be slightly modified to incorporate the maximum gate delay value in the latest time of arrival constraints and the minimum gate delay value in the earliest time arrival constraints. The future work should include the design of larger circuits using this technique and the application of the technique to newer fabrication technologies, especially in the environment of higher leakage. The use of multi-threshold transistors for reduced leakage may be incorporated in the LP for a simultaneous glitch elimination. The problem of glitch-free standard cell based design of appication-specific integrated circuits (ASIC) is also relevant [26].

References

[1] V. D. Agrawal, "Low Power Design by Hazard Filtering," in *Proc. of 10th International Conference on VLSI Design*, Jan. 1997, pp. 193–197.

[2] V. D. Agrawal, M. L. Bushnell, G. Parthasarathy, and R. Ramadoss, "Digital Circuit Design for Minimum Transient Energy and a Linear Programming Method," in *Proc. of 12th International Conference on VLSI Design*, Jan. 1999, pp. 434–439.

[3] M. Berkelaar, "Statistical Delay Calculation," in *Workshop notes of the International Workshop on Logic Synthesis*, (Lake Tahoe, CA), May 1997, pp. 2.1.1–2.1.4.

[4] M. Berkelaar, "Statistical Delay Calculation: A Linear Time Method," in *Proc. of IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Dec. 1997, pp. 15–24.

[5] M. Berkelaar, P. Buurman, and J. Jess, "Computing Entire Area/Power Consumption versus Delay Tradeoff Curve for Gate Sizing Using a Piecewise Linear Simulator," *IEEE Transactions on Circuits and Systems*, vol. 15, no. 11, pp. 1424–1434, Nov. 1996.

[6] M. Berkelaar and E. Jacobs, "Using Gate Sizing to Reduce Glitch Power," in *Proc. of ProRISC Workshop on Circuits, Systems and Signal Processing*, (Mierlo, The Netherlands), Nov. 1996, pp. 183–188.

[7] M. Berkelaar and E. T. A. F. Jacobs, "Gate Sizing Using a Statistical Delay Model," in *Proc. of Design Automation and Test in Europe Conference*, (Paris, France), Mar. 2000, pp. 283–290.

[8] M. Berkelaar and J. A. G. Jess, "Transistor Sizing in MOS Digital Circuits with Linear Programming," in *Proc. of European Design Automation Conference*, (Mierlo, The Netherlands), Mar. 1990, pp. 217–221.

[9] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits.* Boston, MA: Kluwer Academic Publishers, 2000.

[10] "Affirma Analog Environment Reference," 2003. http://sourcelink.cadence.com, Cadence Design Systems.

[11] A. Chandrakasan and R. Brodersen, editors, *Low-Power CMOS Design.* New York: IEEE Press, 1998.

[12] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design.* Boston: Kluwer Academic Publishers, 1995.

[13] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.

[14] S. Datta, S. Nag, and K. Roy, "ASAP: A Transistor Sizing Tool for Area, Delay and Power Optimization of CMOS Circuits," in *Proc. of IEEE International Symp. Circuits and Systems*, May 1994, pp. 61–64.

[15] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *J. of Applied Physics*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[16] J. P. Fishburn and A. E. Dunlop, "TILOS: A Posynomial Approach to Transistor Sizing," in *Proc. of International Conference on Computer-Aided Design*, Nov. 1985, pp. 326–328.

[17] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective.* Upper Saddle River, NJ: Prentice Hall, 2003.

[18] J. M. Rabaey and M. Pedram, *Low Power Design Methodologies.* Boston: Kluwer Academic Publishers, 1995.

[19] T. Raja, "A Reduced Constraint Set Linear Program for Low-Power Design of Digital Circuits," Master's thesis, Dept. of ECE, Rutgers University, Piscataway, NJ 08854, Mar. 2002.

[20] T. Raja, *Minimum Dynamic Power CMOS Design with Variable Input Delay Logic.* PhD thesis, Dept. of ECE, Rutgers University, Piscataway, NJ 08854, May 2004.

[21] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Minimum Dynamic Power CMOS Circuit Design by a Reduced Constraint Set Linear Program," in *Proc. of 16th International Conference on VLSI Design*, Jan. 2003, pp. 527–532.

[22] T. Raja, V. D. Agrawal, and M. L. Bushnell, "CMOS Circuit Design for Minimum Dynamic Power and Highest Speed," in *Proc. of 17th International Conference on VLSI Design*, Jan. 2004, pp. 1035–1040.

[23] K. Roy and S. C. Prasad, *Low-Power CMOS VLSI Circuit Design.* New York: Wiley Interscience Publication, 2000.

[24] C. V. Schimpfle, A. Wroblewski, and J. A. Nassek, "Transistor Sizing for Switching Activity Reduction in Digital Circuits," in *Proc. of European Conference on Circuit Theory and Design*, volume 1, Aug. 1999, pp. 114–117.

[25] V. Sundararajan, S. Sapatnekar, and K. Parhi, "Fast and Exact Transistor Sizing Based on Iterative Relaxation," *IEEE Transactions on Computer Aided Design of Circuits and Systems*, vol. 21, no. 5, pp. 568–581, May 2002.

[26] S. Uppalapati, "Low Power Design of Standard Cell Digital VLSI Circuits," Master's thesis, Dept. of ECE, Rutgers University, Piscataway, NJ 08854, Oct. 2004.

[27] A. Wroblewski, C. V. Schimpfle, and J. A. Nassek, "Automated Transistor Sizing Algorithm for Minimizing Spurious Switching Activities in CMOS Circuits," in *Proc. of IEEE International Symp. Circuits and Systems*, May 2000, pp. 291–294.

[28] A. Wroblewski, O. Schumacher, C. V. Schimpfle, and J. A. Nassek, "Minimizing Gate Capacitances with Transistor Sizing," in *Proc. of IEEE International Symp. Circuits and Systems*, May 2001, pp. 186–189.