2011 24th Annual Conference on VLSI Design

Path Delay Tuning for Performance Gain in the face of Random Manufacturing Variations

Kautalya Mishra, Ahmed Faraz, Adit D. Singh
Auburn University, Auburn, Alabama
Abhijit Chatterjee
Georgia Institute of Technology, Atlanta, Georgia

***Abstract*—One of the factors now beginning to seriously limit clock rates in large synchronous designs is manufacturing variations in device parameters. Moreover, such random process variations are increasing significantly with device scaling as technology approaches the end of the silicon roadmap. In a large design containing several millions of transistors, virtually every manufactured part will have a few hundreds of transistors that are significant performance outliers. Any one such device in a critical path can greatly limit the highest clock rate that can be achieved by the chip. In this paper we propose and analyze a new design approach that allows for the post manufacture tuning and speed-up of exceptionally slow circuit paths to recover much of the performance lost due to such outlier devices. We show that such tuning of exceptionally slow paths can result in a significant increase in the average clock speed attainable by the manufactured parts. We also show this method to be defect tolerant, implying an additional benefit of increasing the semiconductor yield.**

***Index Terms*—Performance, tuning, path, delay, speed-up, clock rate**

I. Introduction

Sustaining technological development as forecast by Moore's Law is now facing formidable new challenges [1], as further scaling comes up against fundamental limitations imposed by the inherent physical properties of the underlying materials, required tolerances down to a few atomic dimensions in the manufacturing processes, and statistical anomalies from the finite number of atoms constituting device structures in highly scaled technologies. While the International Technology Roadmap for Semiconductors (ITRS) predicts continued success with scaling device dimensions for at least another decade, circuit performance gains, as measured in terms of processor clock rates, are projected to taper off. Indeed, clock rates for the fastest commodity microprocessors have stagnated, or even fallen back a little, in the past couple of technology generations. To compensate, microprocessors are being designed with multiple cores to achieve increased performance through parallelism. Future plans calls for dozens and even hundreds of cores in advanced designs. However, decades of research on multiprocessing indicates that additional processors yield diminishing returns in most general purpose applications – a result that is commonly referred to as Amdahl's Law. While performance improvement to date from dual and quad core designs has been impressive, moving to an ever larger number of cores in lieu of faster clock rates in the long term is unlikely to provide performance gains in general purpose computing applications comparable to those experienced during past decades of microprocessor development. To exploit the full potential of the scaled technology, it is imperative to develop innovative techniques that address challenges that limit processor clock rates from achieving faster clock frequencies with scaling.

One of the key mechanisms beyond the control of designers that limits clock rates in large clocked designs is manufacturing variations in device parameters. Such random process variations are increasing significantly in nanoscale devices, and are expected to be even more of a factor at the 22nm node and beyond [2-7]. In a large design containing billions of transistors, it is statistically likely that virtually every manufactured part will have dozens of transistors that are performance outliers, with parameters 4-6 standard deviations or greater from nominal values. Even a single abnormally slow device, say 3-6X or more slower than the nominal, falling on a critical path can significantly limit the clock rate that can be achieved by that specific chip. The actual performance degradation due to the outlier will of course depend on the exact contribution of the exceptionally slow device to the overall path delay. It is to be noted however, that most high performance applications are speed optimized in the design stage to have all or a large number of balanced paths close to equal to the worst case critical path delay. This is done to reduce the amount of power dissipated in the chip. In such applications the presence of an outlier transistor on any path can push up path delays by a significant amount, since there is a statistically high chance of that path being close to critical. To illustrate this with an over simplified example, a 4X increase in delay of a single outlier gate along an 8-gate critical path increases the path delay by 50% and would reduce the allowable clock rate by 33%.

In this paper we propose and analyze a new design approach that allows the post manufacture tuning and speed-up of exceptionally slow circuit paths to recover much of the performance lost due to such outlier devices from manufacturing. It is assumed in this work that each chip or core can be individually "speed binned" and operated at the fastest clock rate that it can reliably sustain to take advantage of this tuning. The speed up is performed post manufacturing to compensate for specific instances of parameter variations in individual ICs and cannot be performed at the design stage. Process variations can only be guard banded against during design, which is inefficient and imposes significant performance penalties on all parts. The core idea investigated in this paper is to add a minimum amount of extra circuitry in each CMOS gate that can be programmed, post manufacture,

This material is based upon work supported by the National Science Foundation under Grant numbers CISE 0834620 and CISE 0903449.

to speed up one selected transition, either rising or falling, at the cell output. The cost paid is some increase in delay for the complementary transition at the cell output, and additional static power dissipation in the cell. Using this gate design, once an exceptionally slow outlier path in a manufactured circuit is identified, cells along the path can be programmed to speed up the clock limiting transition. Such tuning, limited to a few exceptionally slow paths that are statistical outliers can result in a significant increase in the attainable clock speed for the design, with minimal impact on power dissipation. Power dissipation impact is expected to be minimal because only a few transistors, in a circuit with close to billion transistors, will be turned on.

The rest of the paper is organized as follows. The next Section presents our modification of the CMOS gate to introduce programmable performance tuning transistors. Section 3 presents SPICE simulation results for critical path speed-up in a small circuit. Section 4 presents statistical results for average potential speed-up in large high performance designs. The results and implications are further discussed in Section 5. Section 6 concludes the paper.

II. Overview of the Tunable CMOS Gate

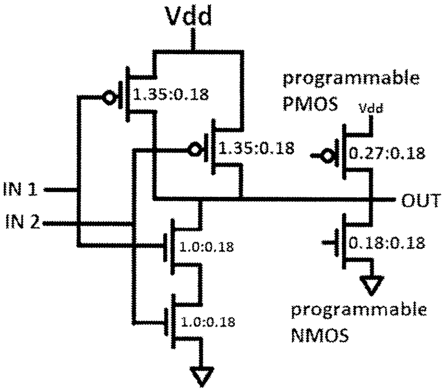

Figure1. A Tunable NAND2 Gate

The proposed tunable CMOS gate design adds two programmable transistors in each gate, a P transistor parallel to the pull-up P transistor network, and an N transistor in parallel to the pull down N transistor network. While this is shown for a two-input NAND gate in Figure 1, the same two transistors can be added to any gate, including complex CMOS gates. The transistors are sized as in a pseudo N-MOS design to ensure that the redundant P transistor has an effective channel resistance that is at least 4X that of the highest possible resistance of the pull down N network; this ensures that with the redundant P transistor ON, the pull down network is still able to provide an acceptable low level output voltage of about VDD/5. The redundant N transistor is similarly sized to be 4X the worst case P network resistance. Figure 1 shows the sizing that can be used for a tunable NAND, assuming a 2:1 carrier mobility ratio for the N and P transistors. The ratios 1:0.18 and 1.35:0.18 are the standard NAND gate transistor ratios obtained from the design tools for 180nm technology. The tuning transistor ratios are fixed by us.

Normally the redundant tuning transistors are OFF, and the gate behaves like a normal CMOS gate. The switching delay for one transition, either rising or falling at the gate output can be speeded up by permanently turning ON one of the two redundant transistors. For example, if the redundant P tuning transistor is permanently turned ON, it will assist and speed up rising transition at the output, while opposing and slowing down falling transitions. Figure2 (a) shows SPICE simulations (for randomly chosen near nominal transistor parameter values) of the NAND gate in Figure 1 with the P tuning transistor both ON and OFF to display this speed up of the rising transition, and slow down of the falling transition. The plots are obtained by sending a pulse through one input, while keeping the other input at logic 1 always. Observe that, as expected, the rising transition is modestly speeded up (28%), while the falling transition is slowed down (8%).

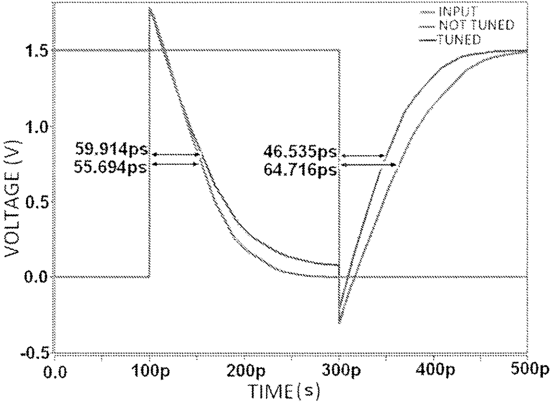

Figure 2(a): SPICE Simulation results for rising and falling transitions (random near nominal transistor parameter values)

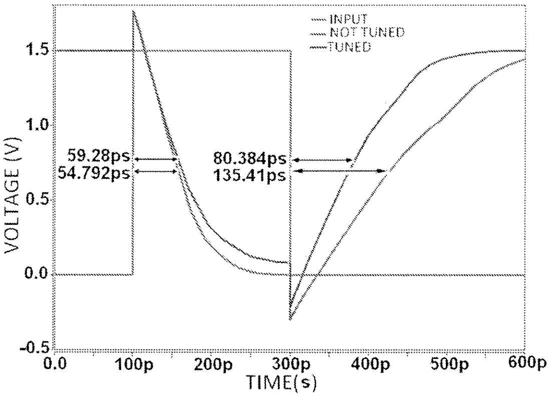

Figure 2(b): SPICE Simulation results for rising and falling transitions (slow outlier pull-up P transistor controlling rising transition)

Figure 2(b) shows a similar simulation for the case where the pull up P transistor conducting during the rising transition is a performance outlier, with a Vth value of 0.2V above the nominal value. Observe in Figure 2(b) how the greatly increased rise time due to the slow outlier pull up transistor is significantly speeded up by turning on the redundant P tuning transistor. In this case the negative impact on the complementary falling transition is less in absolute terms; the speed up in the rising transition (41%) is significantly more than the slowdown in the falling transition (8%). This allows us a means to tune exceptionally slow paths due to outlier devices using this gate structure, without excessively increasing the complementary transition. Additional speed up of the slow transition containing the outlier transistor can be achieved by speeding up other gates on the same path, even if they contain nominal transistors, up to the point where the speeded up path, or the complementary transition which is now slowed down, or a different path, displays the worst case delay for the circuit, which is acceptably close to the average case delay.

Adaptation of the methodology described above requires a post manufacture programming capability to turn on the selected tuning transistors. It is envisioned that this will be implemented using a non volatile memory technology such as floating gate transistors. (The tuning gates can themselves be programmable floating gate transistors). A number of such technologies, being developed for use in reconfigurable circuits, can be adapted to our approach. Commonly they superimpose a memory like grid on the design to support the programming of the configuration transistors.

Including tuning transistors adds an area overhead to the design. But this increase in area is compensated by the Silicon that is being saved, as a consequence of getting a higher yield that is also achieved through this design methodology. Also, note that while the area overhead of the proposed approach appears somewhat high in the context of the small NAND gates, as in Figure 1, it will be lower for larger NAND gates and multi-input complex gates.

Another point to be noted is that the drain diffusion capacitances of the two tuning transistors get included as parasitic capacitances, and contribute to some additional gate switching delay. But it is showed that the worst case delays for outlier cases are brought down considerably on tuning, and the final gain obtained overcomes the increase due to the additional parasitic capacitances.

It should be pointed out that a CMOS cell similar to the redundant cell structure in Figure 1 was presented in [11] as a defect tolerance methodology. The idea there was that if a manufacturing fault occurred in the pull-up P network in the cell, the entire P network is disconnected from the power rail (this requires extra configuration transistors, not required by the design in Figure 1) and is replaced by a properly sized single always ON P pull up transistor (our tuning transistor) to effectively convert the CMOS gate into a pseudo N MOS structure. Similarly, the redundant N pull down transistor can replace a faulty N network. However, this defect tolerant structure has never been considered nor analyzed for performance tuning in any earlier research, which is the main innovative contribution of this work. Clearly the potential exists to combine both mechanisms and achieve both defect tolerance and performance tuning in aggressive CMOS technologies.

III. Simulation Results for Speed-Up from Performance Tuning

Several factors can affect the overall clock performance gain that can be achieved by such a tuning methodology. The size of the design is a key factor. Observe that path tuning is most effective in boosting performance that is severely limited by a few extreme outlier devices. Such outliers can be expected to number from about one device in 10,000 to one in 1,000,000 or fewer depending on the quality and statistics of the manufacturing process. If designs are extremely small, containing only a few thousand transistors, most manufactured parts will be free of slow outlier transistors, so any performance gain from tuning a few circuits averaged over the entire production population will be very small. On the other hand, virtually every multi-million transistor design is likely to contain a few performance limiting outlier devices; the average performance improvement here from tuning can potentially be much greater.

A second important factor impacting the effectiveness of the proposed performance tuning approach is the path delay distribution of the design. If this distribution is highly skewed with relatively few long paths, worst case performance will be less affected by slow devices; in most cases such outlier devices will be found on the (more numerous) shorter paths where their performance impact will be mostly absorbed by the timing slack. On the other hand, fast, speed optimized designs typically have a large number of balanced critical and near critical paths. These are all critically vulnerable to slow transistors and can greatly benefit from performance tuning. High performance designs mostly fall into this latter category. Thus, tuning path delays has the greatest potential application in large, high performance designs, in the face of significant random device performance variability.

In practice, path tuning will require testing and diagnosis of the performance limiting slow path, followed by activation of speed up transistors in selected gates along the paths targeted for speed-up.

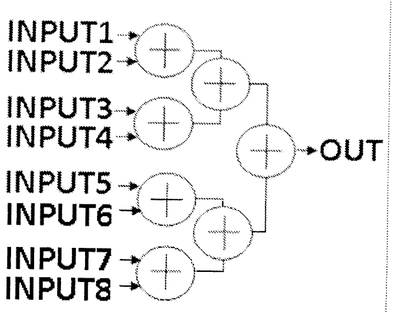

Figure 3a:
7-EXOR Tree Circuit

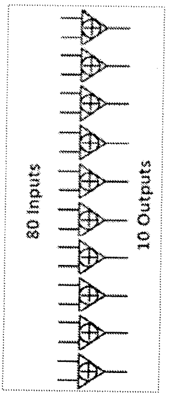

Figure 3b:
Circuit Constructed from
10 7-EXOR Trees

In reality, path delay testing and diagnosis is itself quite challenging [14], but we assume here that such a methodology is available. What can perhaps make this task somewhat more manageable (as compared say to diagnosis to support physical failure analysis) is that the diagnosis need not be exact; multiple candidate transistors or paths can be speeded up in sequence, and the parts functionally speed tested until satisfactory performance is observed. Recent advances in the delay testing of multi-core microprocessors [13] will be of relevance here.

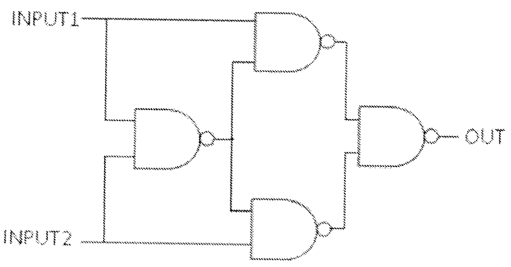

Figure 4: 4-NAND EXOR Implementation

To evaluate the potential performance gain from our tuning methodology, we first investigate the small 8-input EXOR tree shown in Figure 3a. Each EXOR is implemented using 4 NAND gates as shown in figure 4. Recall that an EXOR tree has the property of always propagating the switching transition to the output when a single input is switched, independent of the logic values on the other inputs. While this EXOR circuit contains switching paths of varying lengths, we focus only on the switching paths that result from single input changes with all other inputs held low. There are 8 such paths in the circuit, one from each input to the output. For example, one of the paths we consider is the path from input 1 to the output. We simulate and analyze this path for both rising and falling transitions by forcing a low-to-high, and a high-to-low transition on the input while holding all other inputs low. The seven other similar paths from the other seven inputs to the output are also studied. Observe that there are 6 NAND gates along each of these paths, giving them balanced delays, and mimicking a high performance design. We assume that circuit performance is determined by the slowest transition along these 8 balanced "critical" paths. While the overall circuit has 168 transistors, (84 N transistors and 84 P transistors, including the tuning transistors), only 70 (42 N and 28 P) appear on these 8 balanced target paths. It is assumed that device parameters only for this critical set of 70 transistors on these critical paths affects overall worst case circuit timing; the other paths are assumed short enough (even though in our case they have more gates) that parameter variations are extremely unlikely to make them critical.

Observe that while in order to limit timing simulation complexity we have used a simple example, and limited ourselves to consider only a few circuit paths, the scenario that we have constructed is reasonably realistic. In this small circuit (which we will later use to construct larger designs), performance is determined by the slowest amongst the sixteen transitions that propagate along the eight selected "critical" paths of equal length. Furthermore, approximately 50% of the transistors in the circuit are on these paths and can critically affect overall circuit performance.

The rising and falling transitions along the 8 target paths were simulated to obtain switching delays from input to output using SPICE for a large number of statistical instances (20,000) of the circuit. In each instance of the circuit, to emulate the effect of random process variations, the transistor threshold voltages Vthn and Vthp, for each individual N and P transistor, were randomly drawn from two sets of Gaussian Distributions; one with a standard deviation equal to 0.075V, and the other with a standard deviation of 0.15V. The mean chosen for both the distributions was equal to the nominal value of Vth, $\text{Vth}_{\text{nominal}}$. This ensures that every transistor in the circuit, including the redundant tuning transistors, have randomly drawn threshold values. Furthermore, each instance (copy) of the circuit had different randomly drawn transistor parameters to mimic the many different circuits in a production run. The mathematically small (although perhaps realistic) standard deviation used ensured that relatively few transistors were exceptionally slow (e.g. only about 30 out of every 1,000,000 were statistically expected to be three times or more, slower than the nominal speed). This implied that in most instances of the small 7-XOR gate circuit that we simulated, the observed delays on the 8 critical paths (16 transitions) were fairly well matched, even with the injected process variations. Tuning would not be of much benefit in such circuits. But an occasional circuit instance would contain an exceptionally slow outlier transistor on a critical path. Here path tuning can significantly impact worst case delay. The simulation results for one such instance simulated with 1.5V supply voltage and 0.15V sigma are presented in Table 1.

TABLE I
SAMPLE UN-TUNED AND TUNED PATH DELAYS IN THE 7-EXOR CIRCUIT

		Un-tuned Delays without tuning circuitry	Un-tuned Delays Tuning circuitry included	Delays with Tuning
Input1	Rising	0.6510ns	0.6625ns	0.6626ns
	Falling	0.7045ns	0.7470ns	0.7476ns
Input2	Rising	0.6446ns	0.6804ns	0.6842ns
	Falling	0.6535ns	0.6916ns	0.6919ns
Input3	Rising	0.6500ns	0.6876ns	0.6881ns
	Falling	0.8260ns	0.8765ns	0.8772ns
Input4	Rising	0.6569ns	0.6936ns	0.6944ns
	Falling	0.8666ns	0.9192ns	0.9224ns
Input5	Rising	1.4150ns	1.4910ns	0.7818ns
	Falling	0.6145ns	0.6511ns	0.7204ns
Input6	Rising	1.5310ns	1.6160ns	0.8360ns
	Falling	0.5837ns	0.6164ns	0.6844ns
Input7	Rising	0.8277ns	0.8726ns	0.7621ns
	Falling	0.5912ns	0.6259ns	0.6501ns
Input8	Rising	0.7920ns	0.8382ns	0.7280ns
	Falling	0.5711ns	0.6026ns	0.6282ns
Worst Case Delay		1.5310ns	1.6160ns	0.9224ns

Table 1 shows the output switching delays for rising and falling transitions at each of the 8 inputs for a circuit instance that contains an outlier transistor on a critical path. Observe the exceptionally large delay for the rising transition at input6. Input 5 also displays an abnormally slow rising transition. To speed-up this slow transition we turn on the parallel P tuning transistor in the slow NAND gate with the largest Vth. Further, to achieve additional speed-up, two more tuning transistors on the path to the output are turned ON. Because inputs 5 and 6 share common paths to the output, this also has the effect of speeding up the rising transition on input 5. The timing results for the tuned circuit are also shown in Table 1. Observe that the tuned path is no longer the slowest path; the worst case delay is now reduced to 0.9224ns for the falling transition on input 4. The improvement in worst case path delay in this circuit is approximately 40%.

For tuning paths in general, in this study we used a simple tuning strategy which always turned on the tuning transistor in parallel with the slowest transistor, and then additional tuning transistors in order along the path to the output until the path being speeded up was no longer the slowest path in the circuit. We further limited this tuning to turning on at most 3 tuning transistors. In most cases only single transistor tuning was required. Optimal tuning strategies remain to be developed, and will be the subject of future research.

IV. Estimation of Potential Performance Gain in Large Designs

Clearly if only a very few circuits contain outlier transistors, as was the case for our small circuit in the previous Section, any performance benefit averaged over all manufactured circuits will be insignificant; in fact a loss, because of the increase in the delays that come with the addition of the tuning transistors. Our real interest is in studying very large circuits with a high likelihood of containing at least one outlier on a critical path. Unfortunately, it is not very practical to SPICE simulate thousands of instances of a very large circuit to evaluate average performance gain from tuning in multi-million transistor circuits. To get around this problem, in this Section we present a synthetic experiment based on simulation results for the small 7-EXOR tree of the previous Section. Our goal is to approximate the potential performance gain that might be achieved by a batch of much larger circuits employing our path tuning mechanism.

First we simulate 20,000 instances of our small 7-EXOR gate circuit, each with randomly drawn transistor parameters as described in the previous Section. For each instance, we record the critical (worst case) delay for the target paths. For circuit instances where the performance can be improved through tuning, we re-simulate to obtain the improved (tuned) worst case circuit delay. Thus for each of the 20,000 instances we have three worst case path delay values; one before without any tuning circuitry included, one before with tuning circuitry included, and one after tuning. For a majority of the instances the circuits do not require speed up as the worst case delay values are small or close to the average case delay. However, in the few cases where extreme device outliers are present in the circuit, the performance improvement can be significant, as illustrated by the example in Table 1.

Now suppose we are interested in a larger circuit made up of 10 of our 7-EXOR tree circuits in a parallel configuration. This is an 80 input, 10 output circuit as shown in Figure 3b. To evaluate the statistical benefits of tuning on such a larger circuit, we could directly create and simulate (in SPICE) many instances such circuits as we did before, but this can be computationally very expensive. Alternatively, we can construct instances of this larger circuit by randomly picking 10 instances from the 20,000 available instances of the smaller circuit, for which simulation results are already available. The worst case delay for the synthesized circuit, both pre and post tuning, is easily obtained, without further simulation, to just be the longest of the 10 worst case delays of the individual 7-EXOR circuits. For example, consider the circuit in Figure 3b to be one circuit instance of a large circuit constructed from 10 instances of the smaller circuits. The worst case circuit switching delay for this instance is the largest of the worst case delays for the 10 individual EXOR trees. (It would be this delay that would determine the allowable clock rate if the circuit in Figure 3b was the combinational block of a sequential circuit.) In this way it is possible to construct a statistically significant number of instances of the larger circuit using instances of the smaller circuit, and obtain the worst case delay for each instance both before and after tuning without any additional timing simulation. The average performance gain from tuning over all instances can be easily computed.

This technique also makes it possible to construct and obtain results for even larger circuits containing 100, 1000, and even 10,000 7-EXOR circuits. The last contains over a million transistors, so virtually every instance can be expected to contain several dozen exceptionally slow transistors. Without tuning, average worst case path delay can be expected to be quite large on account of these outliers.

Figures 5a and 5b show a plot of average worst case path delay for two sigma values of 0.075V and 0.15V, both with a supply voltage of 1.5V. The plots depict the observed delay trends for circuits without tuning circuitry present, and before and after tuning, with the tuning circuitry present, versus circuit size measured in terms of the 7-EXOR tree building blocks. As expected, the results show that the average worst case path delay grows quickly with circuit size, and that there is no positive tradeoff obtained with tuning for small circuits, but good benefits are obtained in very large circuits. Circuit delays with tuning circuitry included, but which have not been tuned yet (indicated by the blue curve), are higher than circuit delays without any tuning circuitry included (indicated by the red curve). This is attributed to the presence of the excess parasitic that accompany the additional tuning transistors that are added to the circuit. But tuning a few outlier nodes, for large circuits helps bring the critical delays down (shown by the green curve), well below the delays of circuits without any tuning circuitry included (shown by the red curve). Hence, by tuning only a very small percentage of circuit paths, (only a few dozen in the circuit containing over 1,000,000 transistors) the increase in this average worst case delay can be largely contained.

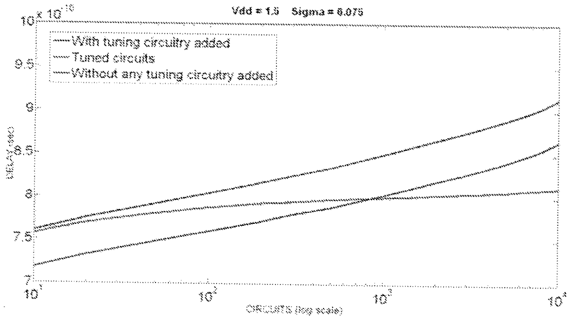

Figure 5a: Worst Case path delay versus Circuit size Vdd=1.5V sigma=0.075

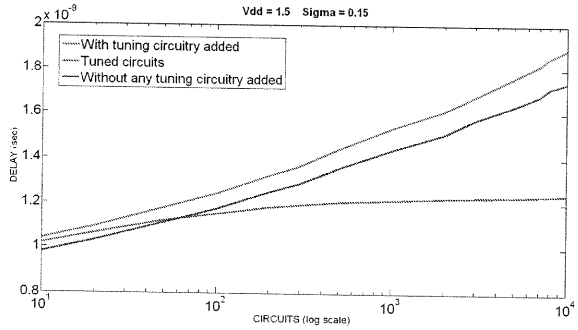

Figure 5b: Worst Case path delay versus Circuit size Vdd=1.5V sigma=0.15

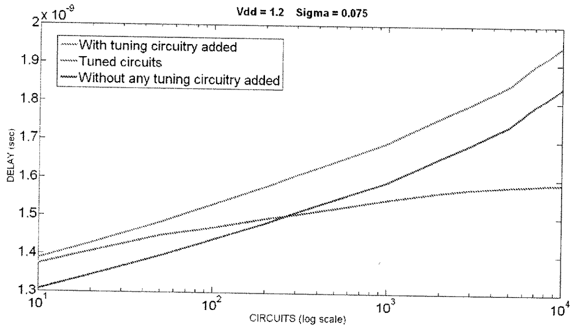

Figure 5c: Worst Case path delay versus Circuit size Vdd=1.2V sigma=0.075

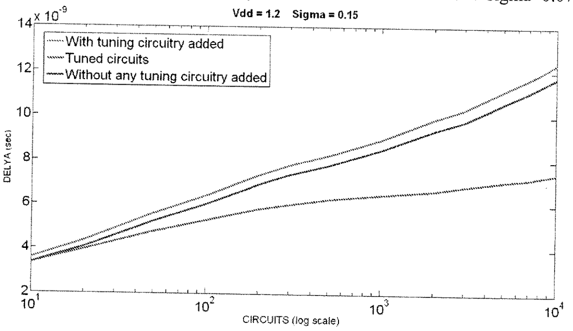

Figure 5d: Worst Case path delay versus Circuit size Vdd=1.2V sigma=0.15

Similar trends were obtained (Figures 5c and 5d) by reducing the supply voltage, Vdd to 1.2V, and simulating it again for the two sigma values of 0.075V and 0.15V. The gains here are in fact better, because of the significant increase in the worst case path delays obtained as a result of reducing the supply voltage Vdd.

In numerical terms, the average worst case delay for the largest circuits in Figure 5d has been reduced from close to 12ns to less than 8ns. This translates to a clock rate speed-up of more than 33%. It is important to keep in mind that this is for a circuit where the critical paths were taken to be 6 (NAND) logic levels. A single exceptionally slow gate will have a smaller proportional impact on a design with longer critical paths. Nevertheless, the impact of tuning on circuits with even 8-12 logic level critical paths can easily exceed 20%, and can result in significant performance gain.

The very large delay values observed in cases simulated with a supply voltage of 1.2V can be explained through observed trends of varying VOL (VOL = Vdd – Vthreshold) versus delay shown in figure 6. As observed, by bringing the supply voltage closer to the threshold voltage, the propagation delays become exponentially large. This trend can be put to use in diagnosing the slow nodes, but further research in that area is yet to be done.

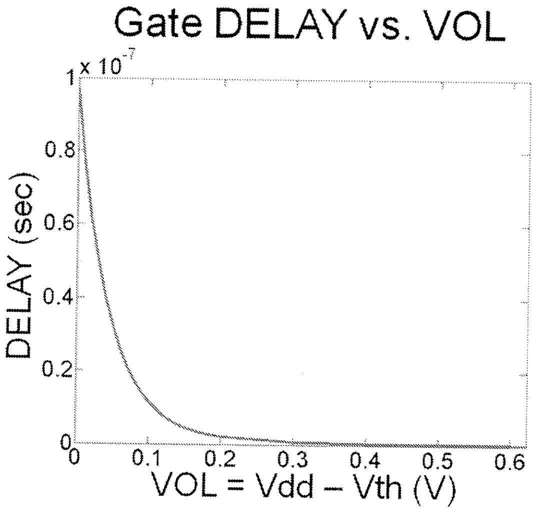

Figure 6: Delay variations with change in VOL for an inverter in a 180nm technology

V. Discussion

Observe again in Figure 5 that the benefits of path delay tuning are negligible for small circuits but become very substantial once circuit size reaches a level where most circuits can be expected to be significantly performance limited by a few random outlier devices. With increasing manufacturing variability with technology scaling, many believe that we are entering such an environment, and that random process variability will get much worse as we approach the end of the silicon roadmap. As we have demonstrated in this first paper on the subject, path delay tuning can be expected to yield handsome performance dividends.

Many important issues remain to be investigated before such an approach can be viable. Critical here are timing test and diagnosis techniques to identify the exceptional path delays, and optimum methodologies for selecting the gates to be speeded up in individual instances of the manufactured chip. On the design side, it is highly desirable to minimize circuit overhead. This will involve identifying gates in the design that cause the greatest degradation in overall performance when they contain a slow transistor, and only providing the proposed tuning capability to such performance critical gates. To make all this work, there is of course the need for a low cost (non volatile) programmable technology to control the tuning transistors.

Finally, as was pointed out in Section 2, a CMOS cell similar to the redundant cell structure in Figure 1 has been presented earlier as a defect tolerance methodology. The idea there was that if a fault occurred in the pull-up P network in the cell, the entire P network is disconnected from the power rail (this requires extra configuration transistors) and is replaced by a properly sized single always ON P pull up transistor (our tuning transistor) to effectively convert the CMOS gate into a pseudo N MOS structure. Similarly, the redundant N pull down transistor can replace a faulty N network. However, this structure has never been considered nor analyzed for performance tuning earlier, which is the innovative contribution of this work. Clearly the potential exists to combine both mechanisms and achieve defect tolerance and performance tuning in aggressive end-of-the-roadmap CMOS technologies.

VI. Conclusion

One of the factors now beginning to seriously limit clock rates in large synchronous designs is manufacturing variations in device parameters. Moreover, such random process variations are increasing significantly with device scaling as technology approaches the end of the silicon roadmap. In a large design containing millions of transistors, virtually every manufactured part will have dozens of transistors that are significant performance outliers. Any one such device in a critical path can greatly limit the highest clock rate that can be achieved by the chip. In this paper we have proposed and analyzed a new CMOS gate design, along with a performance tuning methodology that allows for the post manufacture speed-up of exceptionally slow circuit paths to recover much of the performance lost due to such outlier devices. Our simulations show that such tuning of exceptionally slow paths can result in as much as a 50% increase in the average clock speed attainable by the manufactured parts.

References

[1] http://public.itrs.net/Files/2009ITRS/Home2009.htm, The International Technology Roadmap for Semiconductors, 2009.

[2] P.A. Stolk, F.P. Widdershoven, and D.B.M Klaassen, "Modeling Statistical Dopant Fluctuations in MOS Transistors," *IEEE* Trans. Electron. Devices, vol. 45, no. 9, Sep. 1998, pp. 1960-1971.

[3] Kuhn, Kelin J., "Reducing Variation in Advanced Logic Technologies: Approaches to Process and Design for Manufacturability of Nanoscale CMOS." IEEE International Electron Devices Meeting, *IEDM Technical Digest*, December 2007, pp. 471– 474.

[4] C. H. Diaz, H.-J. Tao, Y.-C. Ku, A. Yen, and K. Young, "An experimentally validated analytical model for gate line edge roughness (LER) effects on technology scaling." IEEE Electron Device Letters, *Volume 22, Issue 6*, June 2001, pp. 287–289. [19] H.-W. Kim, J.-Y. Lee, J. Shin, S.-G. Woo, H.-K.

[5] Cho, and J.-T. Moon, "Experimental investigation of the effect of LWR on sub-100-nm device performance." IEEE Transactions on Electron Devices, Volume 51, Issue 12, December 2004, pp. 1984–1988.

[6] Fukutome, H., Momiyama, Y., Kubo, T., Tagawa, Y., Aoyama, T., and Arimoto, H., "Direct Evaluation of Gate Line Edge Roughness Impact on Extension Profiles." IEEE Transactions on Electron Devices, Volume 53, Issue 11, November 2006, pp. 2755–2763.

[7] Asenov, A., Brown, A.R., Davies, J.H., Kaya, S., and Slavcheva, G., "Intrinsic parameter fluctuations in decananometer MOSFETs introduced by gate line edge roughness." *IEEE Transactions on* Electron Devices, Volume 50, Issue 5, May 2003, pp. 1254–1260.

[8] S. Nassif, K. Bowman, "Design for Variability in DSM Technologies," ISQED 2000

[9] S. Borkar, "Design challenges of technology scaling," IEEE Micro, Vol. 19, Issue 4, pp. 23-29, August 1999.

[10] Ashouei, M., Nisar, M., Chatterjee, A., Singh, A.D., and Diril, A., "Probabilistic Self-Adaptation of Nanoscale CMOS Circuits: Yield Maximization under Increased Intra-Die Variations," International Conference on VLSI Design, Jan. 2007, Bangalore, India, pp. 711-716.

[11] Ashouei, M., Singh, A.D., and Chatterjee, A., "A Defect-Tolerant Architecture for End-of-Roadmap CMOS," European Test Symposium, May 2007, Freiburg, Germany.

[12] A. D. Singh, " Scan Based Testing of Dual/Multi Core Processors for Small Delay Defects", in Proc. International Test Conference, 2008.

[13] A. D. Singh, "A self-timed structural test methodology for timing anomalies due to defects and process variations", in Proc. International Test Conference, 2005.

[14] Saha, K. S., "Modeling Process Variability in Scaled CMOS Technology," IEEE Design and Test of Computers, Vol. 27, Number 2, pp. 8-16, March/ April 2010.

[15] Cheng, B., & Dideban, D., & Moezi, N., & Millar, C., Roy, G., Wang, X., Roy, S., Asenov, A., "Statistical-Variability Compact-Modeling Strategies for BSIM4 and PSP," IEEE Design and Test of Computers, Vol. 27, Number 2, pp. 26-35, March/ April 2010.

[16] Victoria Wang, Kanak Agarwal, Sani R. Nassif, Kevin J. Nowka, Dejan Markovic , "A Design Model for Random Process Variability" Proceeding 2008 ISQED pp. 734-737