High-Speed Interconnect Schemes for A Pipelined FPGA

Hyuk-Jun Lee and Professor Michael J. Flynn

Technical Report : CSL-TR-99-786

August 1999

This research has been supported by the Dept. of the Army under contract DABT63-96-C-0106

High-Speed Interconnect Schemes for A Pipelined FPGA

by

Hyuk-Jun Lee and Professor Michael J. Flynn

Technical Report : CSL-TR-99-786

August 1999

Computer Systems Laboratory

Department of Electrical Engineering

Stanford University

Gates Computer Science Building 2A, Room 230

353 Serra Mall Stanford, California 94305, USA

Phone: 650-723-9310 Fax: 650-725-6949

Email: hyukjunl@umunhum.stanford.edu

Web: http://umunhum.stanford.edu

Abstract

This paper presents two high-speed interconnect schemes for a pipelined FPGA utilizing a locally synchronized postcharging technique. By avoiding a global synchronized clock, we reduce the power consumption significantly. Through postcharging the interconnect and overlapping the postcharging delay with the logic delay, we successfully hide the postcharge time. The long channel devices reduce the area penalty due to delay elements significantly. The timing simulation is done using Hspice for a TSMC 0.35 μm and area is measured by drawing key elements in MAGIC and using the area model developed in[2]. The postcharge scheme shows a 30% delay reduction over the precharge scheme and up to 310% and 230% delay reductions over the conventional NMOS pass transistor scheme and the tri-state buffer scheme.

Contents

List of Figures

List of Tables

1 Introduction

The cycle time of a digital system implemented in Field Programmable Gate Array(FPGA) consists of delay through programmable interconnect and delay through Configurable Logic Block(CLB). The interconnect delay accounts for up to 80% of total cycle time. This delay comes from two factors: large capacitance and resistance in interconnect due to the progrmmable devices and wires, and a RC chain delay that increases quadratically as the length increases. It is reported that the capacitance found in critical path of interconnect is roughly an order larger than that in other CMOS circuits[1]. It is because the NMOS pass transistors, the size of which is roughly ten times larger than the minimum transistor size, are heavily used to achieve programmability. Besides, wires spanning hundreds micrometers introduce significant amount of capacitance. As the technology scales, wire doesn't scale well, which makes the reduction in interconnect delay remain relatively small compared to the logic delay. This results in a larger fraction of cycle time is taken for the interconnect delay.

To achieve high bandwidth interconnect, interconnect pipelining[3] and various buffering schemes[2][4][5] have been proposed. In this research, we propose improved schemes to achieve high throughput for a pipelined[1] FPGA. The techniques use monotonic signaling to reduce the delay for the global interconnect[2]. One way to implement monotonic signaling under a current NMOS-efficient technology is to precharge the interconnect. However, this technique has several potential drawbacks such as clock skew, increased power consumption, and precharge time overhead. As an alternative we propose two postcharge schemes that resolve the problems and show more than a 30% performance gain over the precharge scheme and up to 310% and 230% gains over the conventional gate-boosted NMOS pass transistor and tri-state buffer scheme.

2 Background

2.1 Interconnect delay in FPGA

A FPGA consists of three major components, figure 1(a). Configurable Logic Block(CLB), composed of several SRAM lookup tables, flip-flops, and multiplexors, performs logic functions. C-block, which stands for a connection block, connects inputs and outputs of CLB to the wires. S-block, which stands for a switching block, connects one segment of wire to another through programmable NMOS pass transistors, shown in figure 1(b).

Current commercial FPGAs use NMOS pass transistors with SRAM, figure 1(b), to realize the programmable interconnect. The interconnect of a NMOS pass transistor chain, figure 3(a), can be modeled as a resistance and capacitance network, figure 3(b). The delay of RC network can be written as $\frac{n(n+1)}{2} \times R \times C$ where n is the number of segments, and R and C are resistance and capacitance per segment.

Two sources of resistance in the interconnect are wire and NMOS pass transistors. The

[1] In pipelined FPGA the output of a logic block is always latched.

[2] A global interconnect is used to make a connection between the clusters of logic blocks[2].

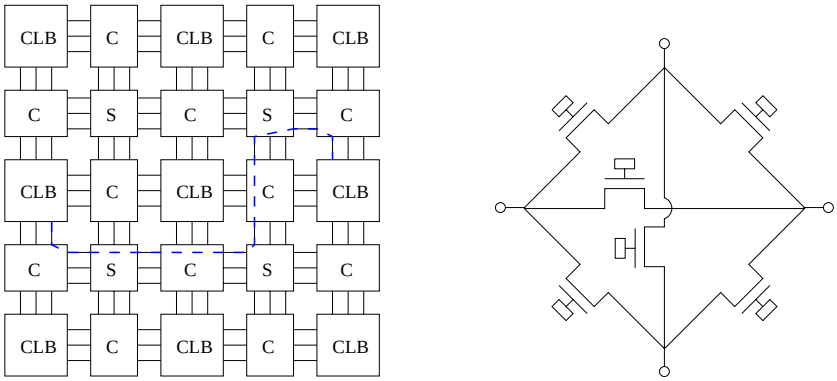

Figure 1: (a) Blocks in FPGA (b) Switch in S-block

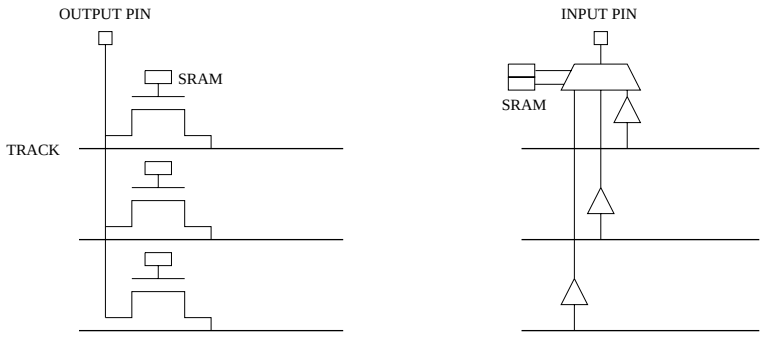

Figure 2: C-block (a)Output Pin Connection (b)Input Pin Connection

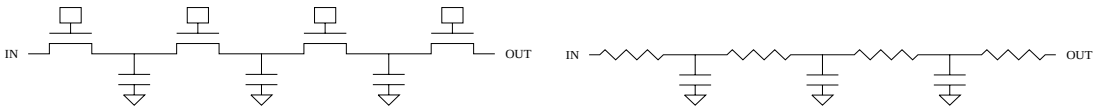

Figure 3: NMOS pass transistor Interconnect (a) Simplified Diagram (b) Simplified Hspice Model

resistance of a minimum width wire segment spanning a thousand λ(one CLB)[3] is 10 ohms in a TSMC 0.35 μm process[4]. This is a small value compared to the resistance of a NMOS pass transistor, which ranges from a few hundred ohms to several Kohms.

While the resistance of a wire segment is negligible, its capacitance accounts for the substantial amount of total capacitance of the interconnect. Wire capacitance consists of two components: the coupling capacitance between adjacent wires and the capacitance to ground. The first decreases as the spacing between the wires increases, and the second decreases as the width of wire decreases. Since the minimum width metal wire gives only a small resistance, the minimum width is widely used.

Other crucial components that constitute the capacitance in interconnect are the gate capacitance of track buffers in C-block, figure 2(b), and the diffusion capacitance of NMOS pass transistors in S-block, figure 1(b). Table 1 shows the decomposition of the capacitance and their values for a interconnect segment spanning one CLB in a TSMC 0.35 μm process.

	Capacitance
Wire(length=200μm, spacing=0.8μm,width=0.8μm)	11.7 fF(to ground) 23.2 fF(to adj. wire)
Track buffers (4 inverters($Wp = 8\lambda, Wn = 4\lambda$)	17.4 fF
NMOS in S-blocks($Width = 40\lambda$, 6 transistors)	64.1 fF
Total	116.4 fF

Table 1: Decomposition of Capacitance for a single line

2.2 Previous work

In the conventional NMOS pass transistor interconnect, the delay is quadratically proportional to the length. In order to achieve a linearly increasing interconnect delay, several buffering schemes are proposed. Tsu et al.[3] proposed a pipelined interconnect and reported that they achieved 4 ns cycle time from a hierarchical interconnect pipelined into three stages. However, this scheme requires substantial FIFOs in input ports and the additional area for the memory element in the interconnect. Besides, the pipelined interconnect doesn't reduce the latency and is only applicable to a hierarchical FPGA. Betz et al.[2] showed that a S-block composed of both NMOS pass transistors and tri-state buffers is more efficient than the NMOS-only S-block in a clustered FPGA. Dobbelaere et al.[4] proposed a scheme that reduces the delay through a regenerative feedback repeater. While their precharge scheme raises clocking issues that remain to be resolved, the CMOS repeater suffers from

[3] λ = half the minimuim feature size for a VLSI process

[4] We chose a TSMC 0.35 μm process developed by Taiwan Semiconductor Manufacturing Company since it is well characterized and its parameters are generally available. The state-of-the-art FPGAs use more advanced technology such as 0.25 and 0.18 μm processes. As the feature size shrinks, the wire and hence interconnect delay scales more slowly than the gate delay. Thus our results provide a conservative estimate for the performance improvement using more advanced technology

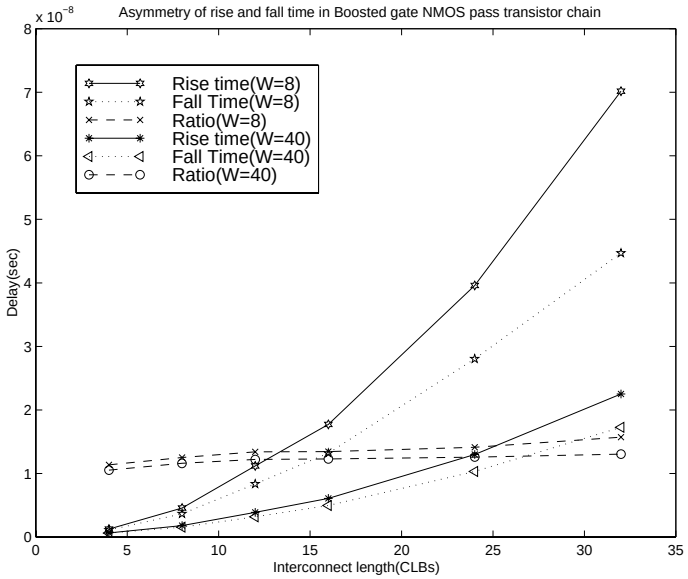

Figure 4: Asymmetry in rise and fall time of gate-boosted NMOS pass transistor interconnect

large area penalty, and is slower than the precharge scheme due to the additional capacitance of a pull-up device. In addition, optimal spacing between repeaters is not readily applicable to the two-dimensional array structure.

Monotonic signaling schemes such as precharging and postcharging provide several additional advantages over conventional buffering schemes. First, they utilize the asymmetry of the resistance of NMOS pass transistors. Figure 4 shows the rise and fall time of interconnect delay using gate-boosted NMOS pass transistors. The voltage at the gate of the NMOS pass transistor is boosted to 3.9 volts while the supply voltage is 3.3 volts. The rise time is $1.2 \sim 1.6$ times slower than the fall time for the NMOS width ranging from 8λ to 40λ. Thus the monotonic signaling, where only the fall time is important, potentially reduces the delay to the worst case fall time. Second, these schemes reduce the size of pull-up devices[5] when they are combined with buffering schemes, which leads to both area and delay reduction. Third, the precharging and postcharging schemes generally don't require gate boosting although there is a slight performance gain. Finally, we can have highly skewed inverters at the receivers and any intermediate buffers since we need to detect only one transition.

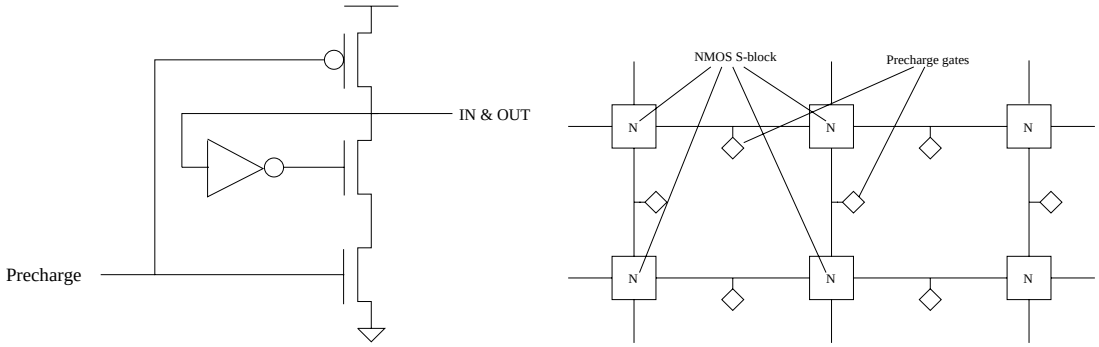

Figure 5: Precharge scheme (a) precharge gate (b) interconnect for precharge scheme

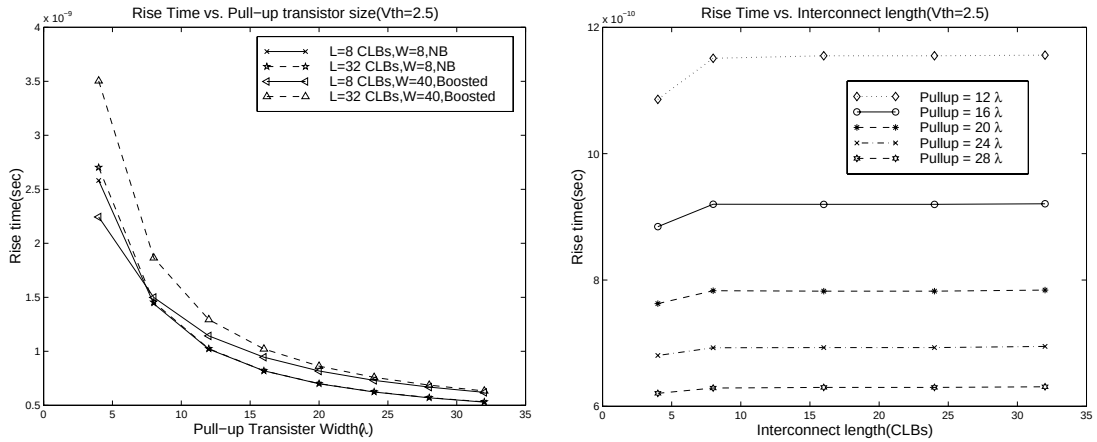

Figure 6: Precharge time (a) different PMOS size (b) different interconnect length

3 Precharge vs. Postcharge

3.1 precharge

The circuit diagram of the precharge gate proposed by Dobbelaere[4] is shown in figure 5(a). The operation can be broken into two phases: precharge and evaluation. The node, "IN&OUT", is precharged when precharge signal is low. A transition from low to high turns on the lower NMOS in the stack. Once a highly skewed inverter detects a signal transition from high to low, the upper NMOS is turned on, and "IN&OUT" node is pulled down through the feedback path. The precharge gate is connected to a track as shown in figure 5(b).

One of the most important issues in the precharging scheme is clocking. One obvious way is that we use precharge logic throughout FPGA. However, it raises whole new issues such as power consumption, clock skew, monotonic signal handling, area overhead. As an alternative, we used the precharging only for the interconnect such that the operation and

[5]The pull-up device is usually twice larger than the pull-down device to match the driving capability

architecture for rest of the FPGA remains the same. However, this scheme requires the precharging time to be small, bounded, and independent of the length of the interconnect. Our simulation shows, figure 6 (b), that the precharge time is independent of the length of interconnect and bounded. For a short interconnect length, the precharge time is determined from the driver capability and the local precharge device width. For a long interconnect length, the precharge time is determined only from the local precharge device width.

A large pull-up device reduces both precharge time and interconnect delay up to a certain point. However, a large pull-up device also increases the capacitance of the interconnect. We chose 24 λ for the pull-up device size because a larger device size only slightly improves the delay and increases the area and power consumption, figure 6(a). The precharge time is roughly 0.7 ns for this device size.

While the design and operation of the precharge scheme is relatively simple and effective, it has several drawbacks. First, it requires a global synchronized clock. A highly skewed clock signal causes a variance in the precharge time. In the worst case the driver fires before the interconnect is fully precharged. However, a well laid-out design such as H-trees introduces only $1 \sim 2$ fanout-4 gate delays [8], which is roughly $150 \sim 300$ ps in a 0.35 μm process. Since the local clock skew is even smaller, the clock skew problem is manageable. Second, a locally generated precharge signal with a 0.7 ns pulse width must drive both pull-up and pull-down device for each track. The device sizes are relatively large: the pull-up is 24 λ and the pull-down ranges from 16 λ to 40 λ. Hence, regardless of a signal transition in the interconnect a local precharge pulse generator has to drive the precharge gates for all the tracks in the channel[6]. This could be a serious drawback for an energy efficient design. Third, the precharge time overhead adds roughly 0.7 ns to the propagation time. Fourth, this scheme is susceptible to noise. The major noise source in FPGA interconnect is coupling noise from adjacent wires. The coupling noise can be described as

$$Vnoise = Vdd \times \frac{2 \times Cc}{2 \times Cc + Cgnd} \tag{1}$$

where Cc is coupling capacitance and Cgnd is capacitance to the ground.

Figure 7(a) shows the coupling noise for various wire spacing. This coupling noise limits the skew ratio of the inverter in the precharge gate, figure 5(a). The noise margin of the skewed inverter should be greater than the worst-case coupling noise. The noise margin of the precharge gate can be theoretically computed by computing the PMOS and NMOS ratio of the inverter when the correctness of operation starts to fail. When the input of the inverter transits from high to low due to noise, the PMOS of the inverter is in saturation and the NMOS of the inverter is in the linear region. We can compute the minimum voltage to keep the NMOS, driven by the inverter, off. Let the voltage be Vin_{min}.

$$\begin{aligned} Vin_{min} &= Vdd - K_c V_{tn} - |V_{tp}| \\ &\quad - \sqrt{(Vdd - K_c V_{tn} - |V_{tp}|)^2 - (|V_{tp}| - Vdd)^2 + 3K_c V_{tn}^2)} \end{aligned} \tag{2}$$

[6] A channel is the space between two adjacent CLBs where the tracks are routed

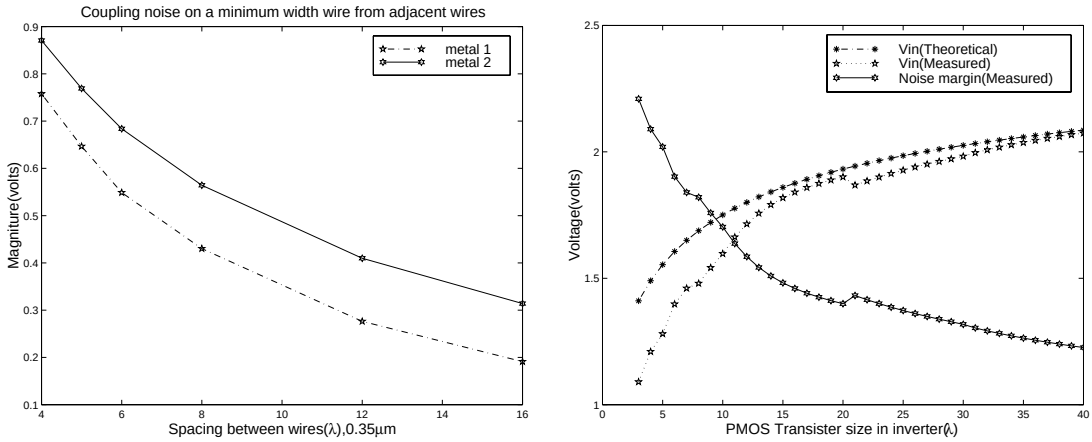

Figure 7: (a)Coupling noise between wires (b)Noise margin

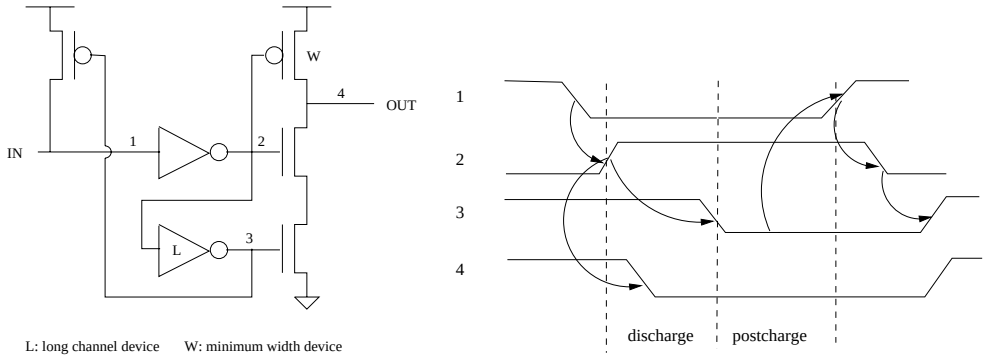

Figure 8: Postcharge I (a) Postcharge gate (b) Operation sequence

where V_{tn} and V_{tp} are the threshold voltage of NMOS and PMOS and $K_c = \frac{\mu_n W_n}{\mu_p W_p}$. The noise margin is defined as $Vdd - Vin_{min}$. Figure 7(b) shows computed and measured Vin_{min} and the noise margin for different PMOS and NMOS ratio assuming $V_{tn} = 0.6, V_{tp} = 0.73, \frac{\mu_n}{\mu_p} = 2.19$. In our design, $W_n = 4\lambda$ and four is chosen for $\frac{W_p}{W_n}$ ratio both here and later in two postcharge schemes because it corresponds to the noise margin of the gate-boosted NMOS pass transistor scheme. Further increase in the skew ratio doesn't improve the delay; Vin_{min} increases slowly and larger PMOS increases gate capacitance. Hence, the gain from the dual-rail version is marginal. This will be discussed later.

3.2 postcharge I

A postcharge scheme literally postcharges the interconnect after the signal propagation instead of precharging it. Figure 8(a) and (b) shows the postcharge gate and the operation sequence. The gate isolates an input node from the output node such that the postcharge PMOS device sees only a small capacitance and thus a minimum device size for the PMOS is sufficient. Initially, the input node and output node are postcharged to high. The high-

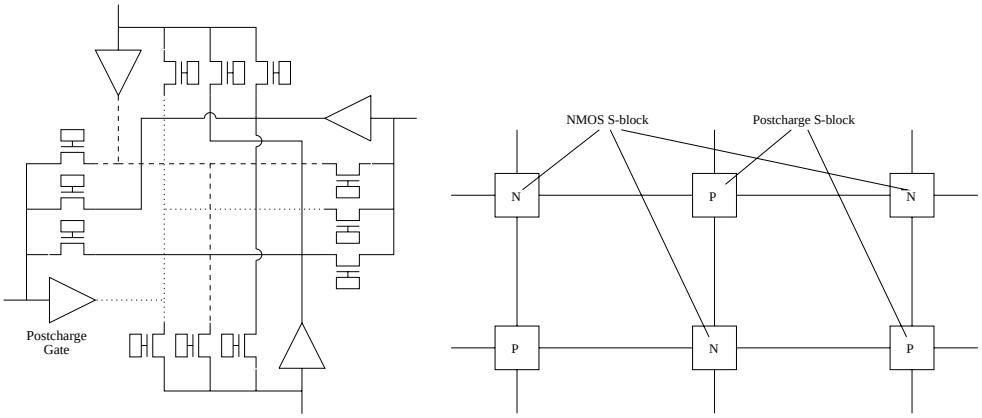

Figure 9: Postcharge I (a) S-block (b) Interconnect

to-low transition in the input node causes the output of the upper inverter to become high and discharges the output node. High at the upper inverter's output makes the bottom inverter's output transit from high to low. This stops discharging. Low at the output of the bottom inverter turns on the PMOS and starts postcharging the input node. Finally, high at the input node propagates through the inverter chain and enables the bottom NMOS in the NMOS stack and disables the upper NMOS and the postcharge PMOS.

A weak PMOS device on the top of NMOS stacks is used to pull the node to high initially and improve the noise margin. The postcharge scheme resolves the problems raised in the precharge scheme. First of all, self-timed postcharge doesn't suffer from the clock skew and power consumption for driving precharge gates. Furthermore, the postcharge scheme hides the postcharging latency by overlapping postcharge time with the delay through configuration logic block(CLB). The postcharge time can be set to 3 ns, which is roughly the delay through one CLB. The relatively relaxed timing requirement and node isolation by buffers reduce the pull-up device size substantially. The driver size at the output pin of CLB can be significantly reduced since it drives only one segment.

There are two drawbacks to this scheme. First, the postcharge scheme doubles the number of NMOS in a S-block, figure 9(a), which results in both area and delay penalty. The area overhead can be halved by alternating the postcharge S-block, figure 9(a), and the conventional NMOS S-block. This optimization, figure 9(b), reduces both area and delay. Second, in order to generate a sufficiently long delay for discharge, we are forced to use a long inverter chain. However, the use of long-channel devices reduces the number of inverters in the chain. Our experiment shows one long channel inverter is sufficient for the single lines and three long channel inverters are sufficient for the quad lines[7]. In the postcharge scheme, we can save power consumption by adding a T(toggle) flip-flop at the receiver. In this case, the driver generates a low-going pulse only when there is a transition at the output instead of low at the output. This pulse is used to toggle the output of the T-flip flop. This scheme requires only a small additional area.

[7] A single line is a wire segment connecting two S-blocks that are one CLB apart. A quad line connects two S-blocks that are four CLBs apart

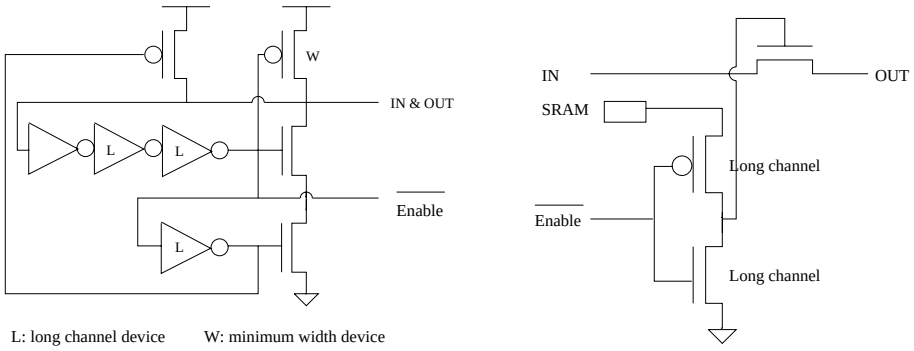

Figure 10: Postcharge II (a) Postcharge gate (b) Modified NMOS and SRAM pair in S-block

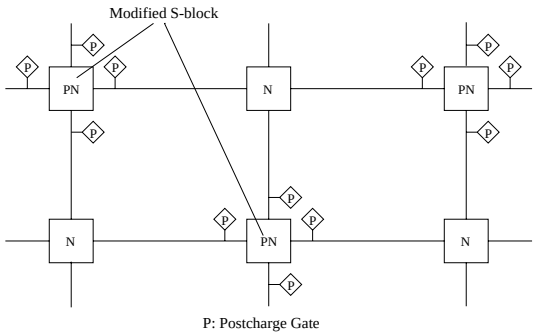

Figure 11: Interconnect for postcharge scheme II

3.3 postcharge II

One major difference from the postcharge scheme I is that the input and output of the postcharge gate in the scheme II are connected, which provides feedback as in the precharge scheme. Figure 10 (a) shows the postcharge gate. This gate must be placed close to the S-block, figure 11, because $\overline{enable}$ disables the NMOS pass transistors in S-block during the postcharge phase. The SRAM and NMOS pass transistor in the conventional NMOS S-block is replaced with a circuit shown in figure 10(b). The inverter's power supply is connected to the output of a SRAM. When the output is low(ground), the inverter's output is low(ground) disabling the NMOS pass transistor. When the output is high, $\overline{enable}$ controls the NMOS pass transistor. Each NMOS in S-block, figure 10(b), can be controlled by $\overline{enable}$ from either of the two postcharge gates to which it is connected.

The operation of the scheme II is similar to that of scheme I. The difference is that in the postcharge phase each node is isolated from its neighbors by disabling the NMOS pass transistors in S-blocks.

The scheme II reduces the number of NMOS pass transistors in a S-block from 12 to 6 per track. This reduces the number of NMOS diffusion physically connected in a S-block from 9 to 6 which leads to a slight delay reduction. A larger delay reduction comes from the use of feedback in the pull-down network. When the input of the upper inverter, shown

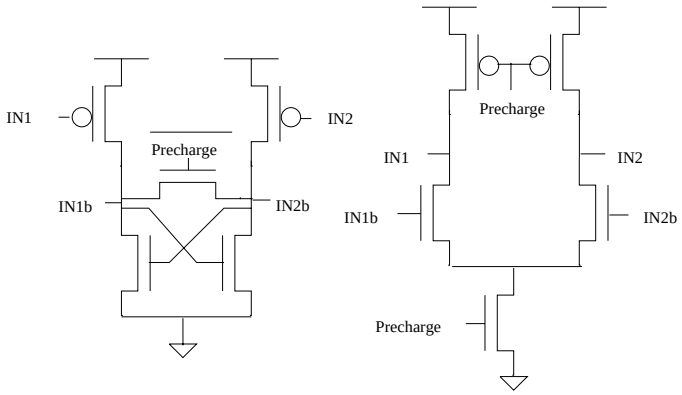

Figure 12: Dual-rail design for precharge method

in figure 10(a), starts to go low, it turns on the pull-down path and pull down the input much faster.

The area overhead in this scheme includes additional delay elements per wire and an inverter for each NMOS and SRAM pair in a S-block.

3.4 Dual-rail design

A dual-rail design is applicable to both precharge and postcharge schemes. Figure 12 shows the dual-rail implementation for the precharge scheme with cross-coupled NMOS. The dual-rail design provides two potential advantages. First of all, it reduces the sensitivity to the noise commonly coupled on two wires. This relaxes the constraint imposed on the skew ratio of an inverter. It allows the higher skew ratio that could improve the interconnect delay. Secondly, cross-coupled design improves the transition speed through its positive feedback. However, our experiment shows that the gain from the dual-rail design was marginal. We varied the PMOS size to increase Vin_{max}. As we can see from figure 7(b), Vin_{max} doesn't increase much beyond $Wp = 16\lambda$. Besides, the larger PMOS increases the capacitance in the critical path. Secondly, the feedback used in the dual-rail scheme is not necessarily faster than the feedback used in the single-rail design. Due to the area overhead and maginal performance gain this scheme was not considered further in our research.

4 Simulation

4.1 Delay model

The circuits proposed in this research are implemented in SUE[10] and the critical path with all capacitive loads is simulated using Hspice for a TSMC 0.35 μm process. The loads include S-blocks, track buffers, wires, precharge and postcharge gates, and output drivers.

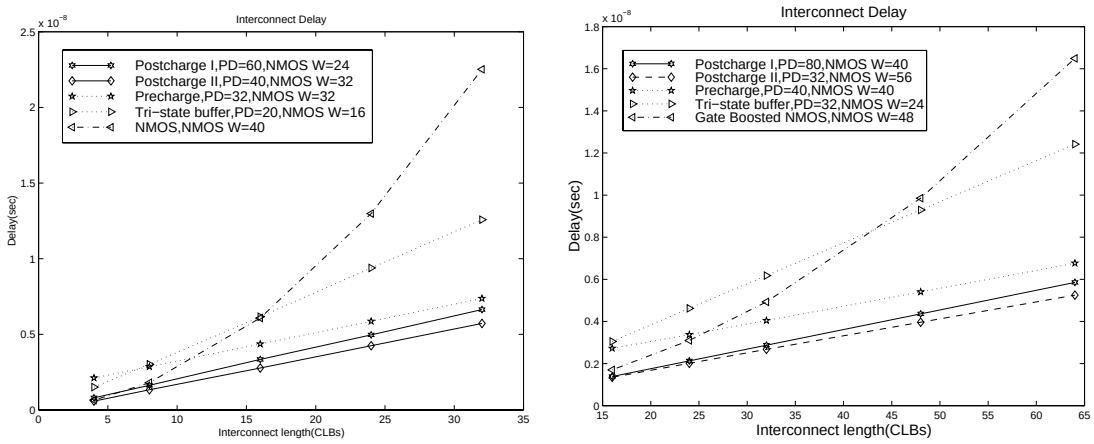

Figure 13: Delay at minimum Area x Dealy (a) Single (b) Quad

4.2 Area model

We use the area model developed by Betz[2]. He developed an area model in which the area for the transistors of various sizes is normalized to the area required for the minimum size transistor. The area includes the minimum space between two transistors. Using the model, total area for a particular design is merely counting the number of transistors in the design and multiplying them with their sizes.

In this research, we adopt his model except for the circuits that use long-channel devices. This is because his model assumes a minimum channel length device. We measured the area for the circuits that require long-channel devices by drawing them using MAGIC in two metal layers. Table 2 shows the area for the buffers used in precharge and postcharge schemes for a single line.

	Area
Precharge(pull-down width=32λ)	$1{,}748\lambda^2$
Postcharge I(pull-down width=32λ)	$1{,}764\lambda^2$ without S-block
Postcharge II(pull-down width=32λ)	$2{,}622\lambda^2$ without S-block

Table 2: Area for the buffers used in precharge and postcharge schemes

5 Results

Figure 13(a) shows the delay for the single line. The pull-down device size(PD) and NMOS width in S-block are chosen for each scheme such that its area-delay product is minimum.

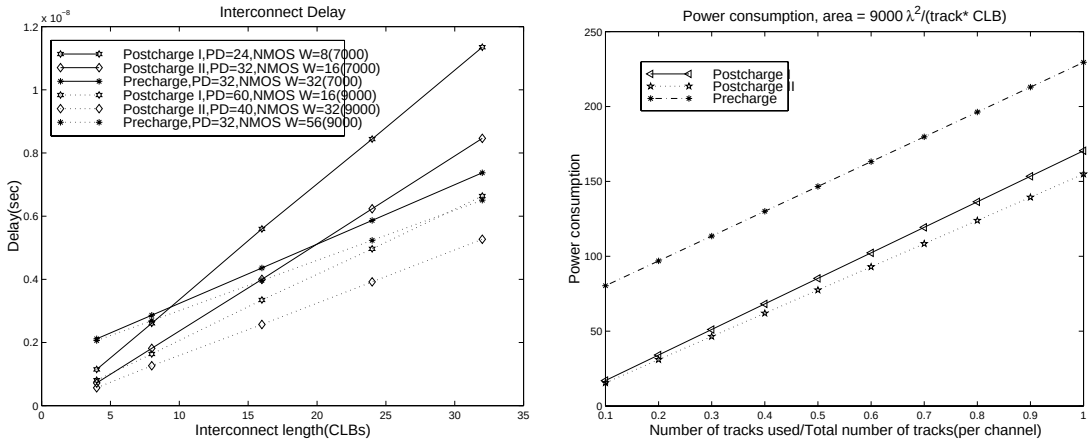

Figure 14: Single line (a) Delay with fixed area (b) Average power consumption at 9000 $\frac{\lambda^2}{track \times CLB}$

The area-delay product curves are shown in Appendix[8]. The gap between precharge and postcharge schemes is caused by the precharge time overhead. The delay for the gate-boosted NMOS pass transistor scheme is reduced by 2.19 (at length = 16 CLBs) to 3.93 (at length = 32 CLBs) using the second postcharge scheme. The delay reduction over a tri-state buffer ranges from 2.19 to 2.56.

Figure 14(a) compares the delay of three schemes at the same area penalty. For the area penalty of 7000 $\frac{\lambda^2}{track \times CLB}$, the precharge scheme outperforms two postcharge schemes. However, with a large area budget, the postcharge schemes show a large gain while the precharge gain is limited. Figure 14(b) compares the average power consumption of three schemes at the same area penalty. The increased power consumption of the precharge scheme is caused by driving large pull-up and pull-down devices for all the tracks in the channel, which takes a significant portion of total power consumption if a small fraction of tracks are used.

The delay comparison for the quad line is shown in figure 13(b). Roughly 1.84(at length = 32 CLBs) to 3.14(at length = 64 CLBs) times reduction in delay is achieved by the second postcharge scheme over the NMOS scheme. The delay reduction over a tri-state buffer ranges from 2.26 to 2.36.

6 Effects on system performance

In FPGAs, area is often traded for performance. Therefore, to measure the effects of the proposed schemes on system performance, the area penalty must be considered. To measure the total FPGA area, we define the baseline FPGA model as in table 3. The cycle time is the sum of delay for the global interconnect and delay through the logic blocks and

[8]In the case where multiple sets of PD and NMOS width are showing comparable area-product values, the one set that shows the best delay was chosen

local interconnect in the cluster. The typical delay for non-cascaded logic blocks and local interconnect is roughly $3 \sim 4ns$ in a 0.35 μm process.

	Components	Total Area
Logic Block	(2) 4-LUT, (8)10:1 Mux to LUT input, (8) output buffer of 10:1 Mux, (2) DFF, (2)2:1 Mux, (2) Clk buffers, S/R Logic	127,000 λ^2
C Block	10 single lines(per C-Blk), 10 quad lines(per C-Blk) 2 Track buffers(per track, C-Blk), 20:1 Mux(per track,C-Blk), (2)Output buffers	116,160 λ^2
S Block	6 NMOS(40λ) (per track, S-Blk)	140,400 λ^2
Total		363,560 λ^2

Table 3: Baseline FPGA

The second postcharge scheme achieves up to 393%(single spanning 32 CLBs) and 314%(quad spanning 64 CLBs) reductions in interconnect delay. This translates into a 240% reduction in cycle time assuming that the cycle time is determined by quad line spanning 64 CLBs. The area penalty is 24% assuming a C-block contains 10 single lines and 10 quad lines.

7 Conclusion

In this paper we present two postcharge schemes to achieve the delay linearly proportional to the length of the interconnect. The postcharge schemes use monotonic signaling and provide important advantages over the precharge scheme in power consumption, clock skew, and precharging time overhead. The postcharge scheme II shows more than 30% delay reduction over the precharge scheme. The delay reductions over the conventional NMOS pass transistor and tri-state buffer scheme are even more significant; 310%(NMOS) and 230%(tri-state). The effects on the system performance and cost are measured as up to 240% reduction in cycle time at 24% area penalty.

8 Acknowledgement

We thank V. Betz at University of Toronto for providing an early copy of his PhD thesis. Also we thank the members of the computer arithmetic and architecture group at Stanford University for their valuable comments.

References

[1] E. Kusse, "Analysis and Circuit Design for Low Power Programming Logic Modules' ', *Master Thesis, University of California, Berkeley*, 1998.

[2] V. Betz, "Architecture and CAD for Speed and Area Optimization of FPGAs", *PhD Thesis, University of Toronto*, 1998.

[3] W. Tsu, et al., "HSRA:High-Speed, Hierarchical Synchronous Reconfigurable Array", *International Symposium on Field Programmable Gate Arrays*, pp. 125-134 Feb. 1999.

[4] I. Dobbelaere, et al., "Regenerative Feedback Repeaters for Programmable Interconnections", *IEEE Journal of Solid-State Circuits*, Vol. 30 No. 11 pp. 1249-1253, Nov. 1995.

[5] A. Takahara, et al., "More wires and fewer LUTs: A design methodology for FPGAs", *International Symposium on Field Programmable Gate Arrays* , pp. 12-19, Feb. 1998.

[6] J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, "Architecture of Field-Programmable Gate Arrays",*Proceedings of the IEEE*, Vol. 81, No.7, pp. 1013-1029, July. 1993.

[7] S. Brown, "FPGA Architectural Research: A Survey",*IEEE Design & Test of C omputers*, pp. 9-15, winter 1996.

[8] D. Harris, "Skew-Tolerant Domino Circuits",*IEEE Journal of Solid-State Ci rcuits*, pp. 1702-1711, Nov. 1997.

[9] Xilinx Corporation, "XC4000 Field Programmable Gate Arrays:Programmable L ogic Databook", 1996.

[10] Micro Magic Inc., 'MMI-SUE Tutorial', http://www.micromagic.com/sue_tutorial.html

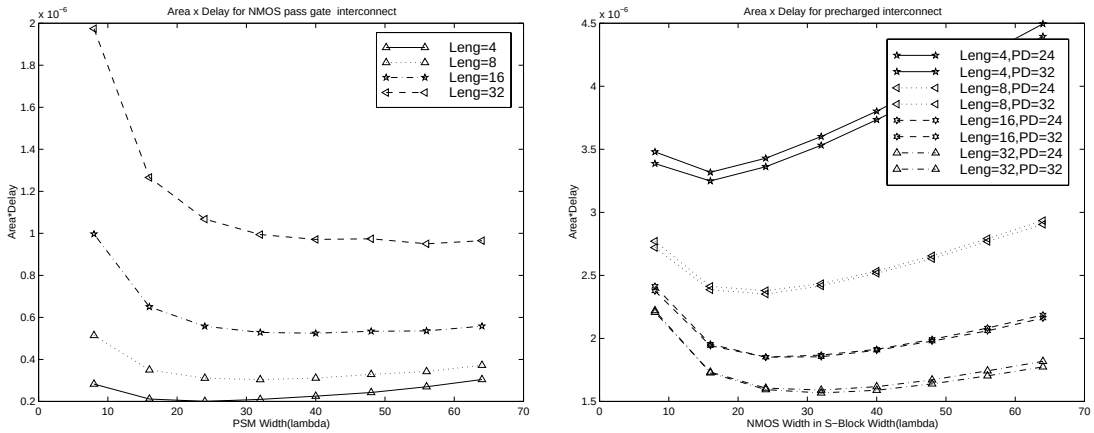

Figure 15: Single (a) NMOS (b) Precharge

Appendix

A single line

Figure 15 (a) and (b) show the area x delay curve for the gate-boosted NMOS pass transistor and the precharge scheme. Each area x delay line is normalized to the length of the interconnect. The NMOS scheme has the minimum area x delay when the NMOS width is 40 λ. The larger width doesn't reduce the delay much because it also increases the diffusion capacitance in the critical path in addition to the increased area penalty. While the NMOS width in S-block is the only variable to optimize in (a), the precharge and postcharge schemes have two variables to optimize: a NMOS width in the S-block and a pull-down device size. For precharge scheme, pull-down sizes ranging from 8 to 32 λ have been simulated. The result, figure 15(b), shows its minimum when S-Block NMOS width is 32 λ and pull-down is 32 λ. Figure 16(c) and (d) shows the area x delay for two postcharge schemes. The first postcharge scheme (c) requires a complex S-block. The rapidly increasing the line beyond NMOS size of 24 λ reflects the dominant area penalty for the S-block. Pull-down width from 24 to 80 λ were simulated. The wide range of device sizes represents a minimum area-delay product value. Those are NMOS width=24 with pull-down width=60, NMOS width=16 with pull-down width=40 and NMOS width=16 with pull-down width=32. For high performance, we can choose the first and for the area efficiency we can choose the third. The minimum is further stretched out in case for the second postcharge scheme (d) because it has the simpler S-block. Although pull-down width of 32λ shows smaller area x delay values for Length = 4 and 8, PD=40 λ performs better for the longer interconnect length. Hence, we chose the curve with pull-down width = 40λ. Figure 17 shows the result for the tri-state buffer. We used a tri-state buffer model proposed by Betz[2].

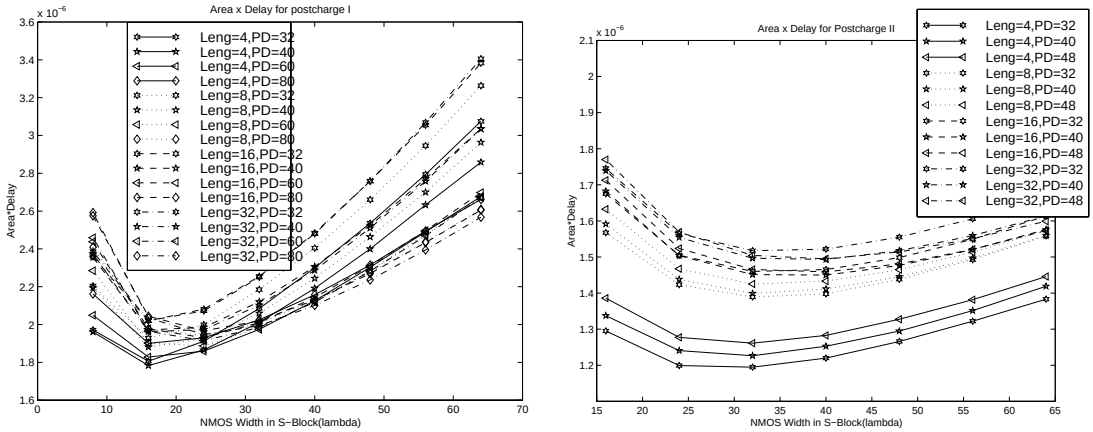

Figure 16: Single (c) Postcharge I (d) Postcharge II

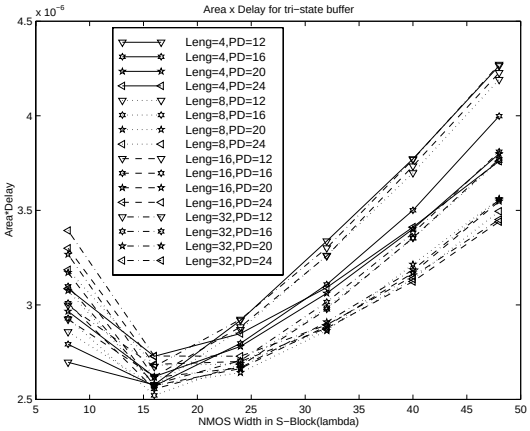

Figure 17: Single (e) Tri-state buffer

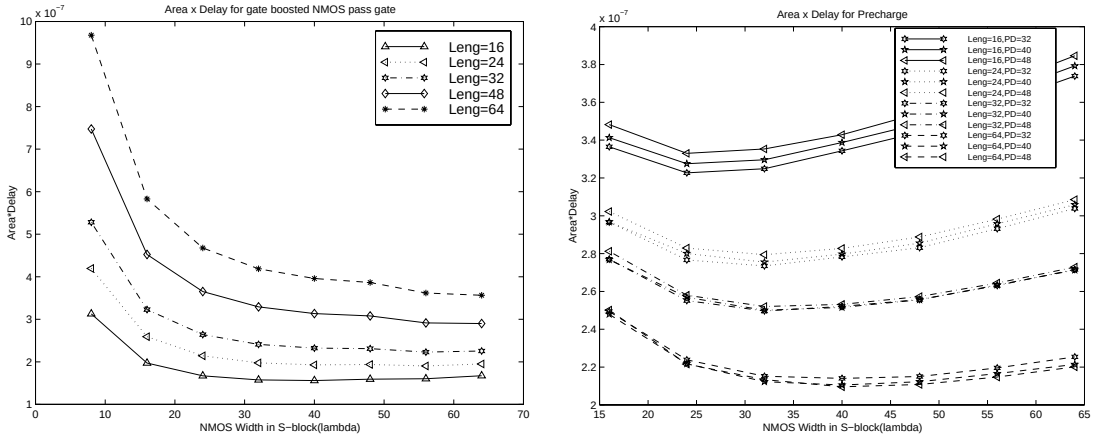

Figure 18: Quad (a) NMOS (b) Precharge

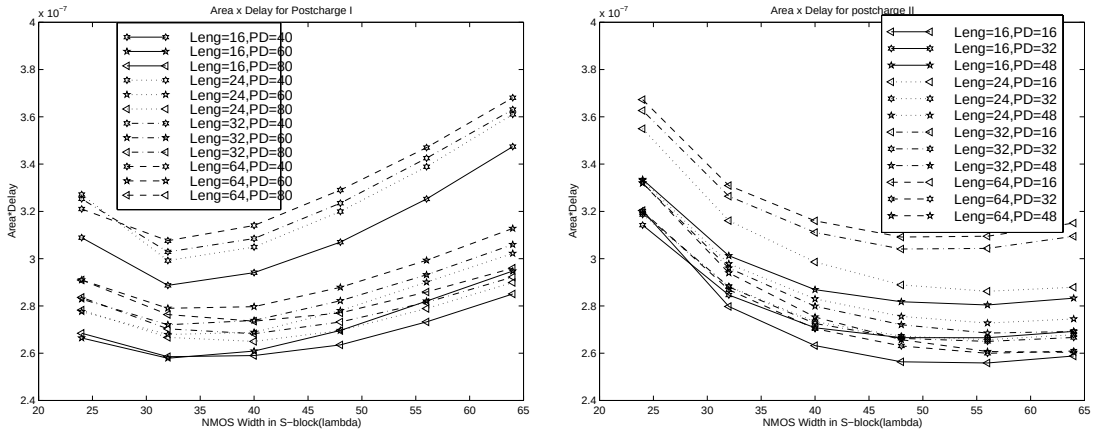

Figure 19: Quad (c) Postcharge I (d) Postcharge II

B quad line

Since a quad line is spanning four CLBs before it is terminated with programmable devices, the capacitance per segment is larger than that of a single line. Hence, the NMOS width in S-block that give the minimum area x delay product is larger than the single line for four schemes. The results are shown in figure 18, 19, and 20.

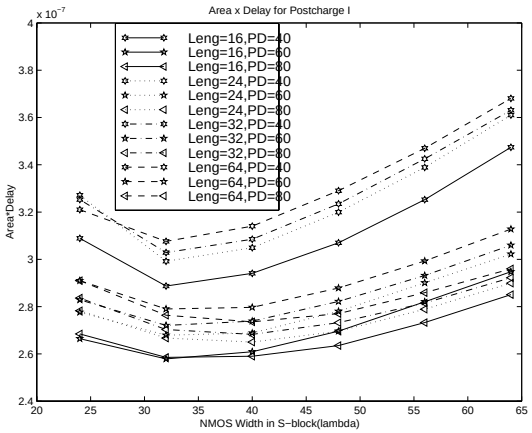

Figure 20: Quad (e) Tri-state buffer