Estimating Reliability and Throughput of Source-synchronous Wave-pipelined Interconnect

Paul Teehan, Guy G.F. Lemieux, and Mark R. Greenstreet
University of British Columbia, Vancouver, BC, Canada

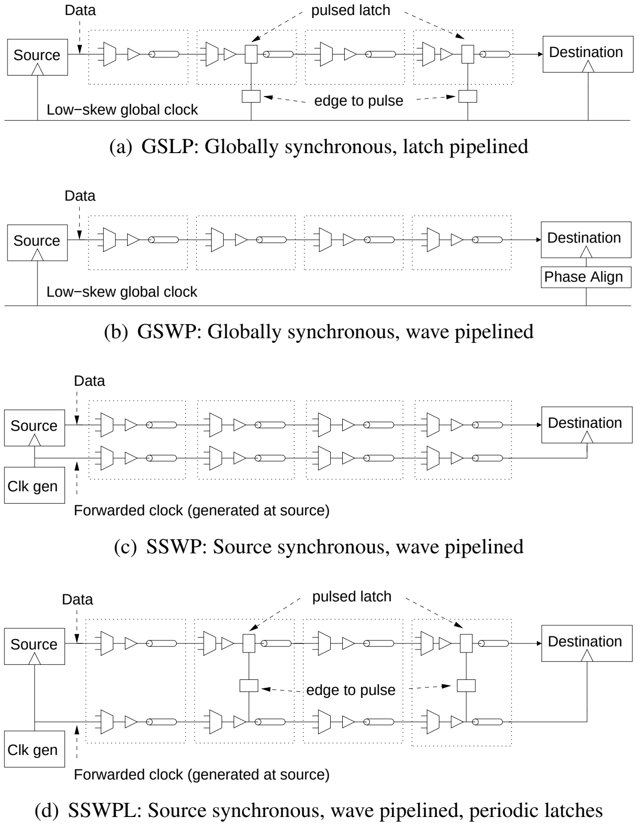
Abstract

Wave pipelining has gained attention for NoC interconnect by its promise of high bandwidth using simple circuits. Reliability issues must be addressed before wave pipelining can be used in practice; so, we develop a statistical model of dynamic timing uncertainty. We show that it is important to distinguish between static and dynamic sources of timing uncertainty, because source-synchronous wave pipelining is much more sensitive to the latter. We use HSPICE simulations to develop a model for a wave pipelined link in a 65nm CMOS process and apply a statistical approach to determine the achievable throughput at acceptable bit-error rates. Reliability estimates show that a modest amount of dynamic noise can cut achievable throughput in half for a ten-stage wave-pipelined link, and will further degrade longer links. After accounting for noise, traditional globally synchronous design is shown to offer higher throughput than the wave-pipelined design.

1 Introduction

Due to the long latencies of global wires, global interconnect is often pipelined. The traditional globally synchronous, latch pipelined (GSLP) model shown in Figure 1(a) is well-understood and can be made highly reliable. Here, a *link* consists of several *stages*. Several recent papers have proposed that wave pipelining may offer further speed, area, and power advantages over traditional latch pipelining [17, 3, 5]. It has also been proposed for global interconnect in FPGAs [10].

Unfortunately, it is difficult to achieve reliable communications with wave pipelining. Figure 1(b) shows wave pipelining with a global clock (GSWP). Without latches to keep edges separated, multiple data bits are simultaneously in flight in the link and have only their nominal time separation to distinguish them. Because the data transport latency can vary, a phase alignment circuit is needed to correct for clock/data skew at the destination. This circuit needs to be continuously adjusted for latency drift and variation resulting in a complex and/or unreliable design. In contrast, Figure 1(c) shows source-synchronous wave pipelining (SSWP), where both clock and data experience similar latency, making phase alignment largely unnecessary. To further reduce skew, it may be necessary to occasionally latch the data, shown as SSWPL in Figure 1(d).



(a) GSLP: Globally synchronous, latch pipelined

(b) GSWP: Globally synchronous, wave pipelined

(c) SSWP: Source synchronous, wave pipelined

(d) SSWPL: Source synchronous, wave pipelined, periodic latches

Figure 1. Types of pipelined interconnect: GSLP, GSWP, SSWP, and SSWPL

Even with SSWP or SSWPL, there is inevitably still some timing uncertainty due to noise and variation throughout the link. Hence, sufficient timing margin must be added between bits to ensure reliable operation. Many techniques for calculating these margins exist, but most rely upon worst-case bounds and are unrealistically pessimistic.

Random statistical models for skew were developed in [7] for synchronous pipelines. Shyur [13] proposed using statistical methods to measure timing uncertainty in wave-pipelined logic circuits, but only considered static timing. Similar yet more sophisticated approaches were taken in [5], but that paper also considered only static uncertainty. Timing constraints for source-synchronous wave pipelining were developed in [2], but that work did not include statistical modeling. There is precedent for using statistical models for random timing uncertainty in off-chip communication, for example in [8] and [12]. Zhang *et al.* [16] analyse latch and flip-flop based interconnect pipelining in a statistical model that includes both static and dynamic uncertainty. In [17], the same authors proposed a SSWP on-chip interconnect but did not extend the statistical analysis of their previous work to this wave-pipelined design.

This paper uses statistical methods to model the timing behavior of source-synchronous wave-pipelined links. This allows us to develop a methodology to estimate the probability of error as a function of circuit structure, throughput, and timing uncertainty due to noise and variation. In our analysis of SSWP and SSWPL, we show that:

- dynamic timing uncertainty causes jitter to accumulate;
- dynamic timing uncertainty plus systematic static variation cause skew to accumulate;
- skew can be attenuated with latches, making jitter the sole factor to limit throughput;
- jitter is a more dominant effect than skew, which differs from GSLP where skew is the dominant effect;
- crosstalk can be mitigated through shielding;
- high-frequency voltage supply noise remains the dominant form of dynamic timing uncertainty; and
- after accounting for dynamic timing uncertainty, traditional GSLP offers higher throughput than wave pipelining when reasonable error-rates are required.

The paper is organized as follows. Section 2 briefly describes a programmable, source-synchronous pipelined interconnect design which is used as an example throughout the paper. Section 3 describes traditional worst-case timing constraints, illustrates the difference between static and dynamic timing, and makes an argument for statistical models. Section 4 shows how to measure dynamic timing uncertainty and provides simulation results which do so. Section 5 uses those results to estimate reliability in terms of probability of error as a function of throughput and dynamic uncertainty. Section 6 concludes the paper.

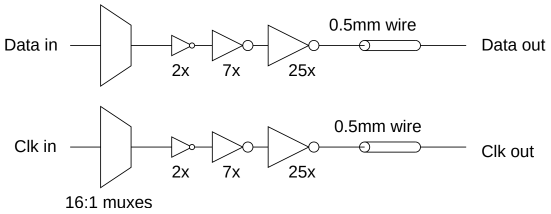


Figure 2. Wave-pipelined interconnect circuit

2 NoC Circuit Design

To evaluate the reliability and throughput of wave pipelining, we start with a particular NoC circuit design which will be used as an example throughput the paper. The four NoC interconnect pipelining strategies are shown in Figure 1 and have already been introduced. We assume the NoC must have low area and support both bursty and continuous data transmission – this limits our ability to employ circuitry like DLLs or PLLs, which can be large and take a long time to lock. Also, we presume that NoC interconnect is based upon a switched fabric, not fixed point-to-point links. Hence, we include multiplexers in the interconnect paths to allow for packet or circuit switching.

To keep power low and ensure clocking rates do not limit data transfer rates, we use pulsed latches that are sensitive to both clock edges (DDR clocking). Figures 1(c) and (d) use source-synchronous clocking. Its main advantage is that data and clock experience similar static timing uncertainty, which means static timing does not limit the link speed. This is elaborated upon in the next section. Although only one data wire is shown in these circuits, similar results would apply for bundled-data wires provided they are kept close enough to encounter the same uncertainty conditions.

Figure 2 shows a detailed circuit of one stage in the interconnect pipeline. The 16:1 multiplexer puts two minimum-size CMOS transmission gates in the signal path to implement a configurable fabric. Three tapered inverters, the largest of which is 25 times minimum size, drive a 0.5mm wire. A link is composed of any number of such stages cascaded together. The circuit was designed to optimize the area-power-delay-throughput product; the last term, throughput, is unique to wave-pipelined circuits and requires a relatively large driver [15].

3 Timing uncertainty in wave pipelining

Many factors affect timing. Some are high frequency disturbances such as crosstalk and power supply noise; some drift at lower rates such temperature or electromigration; and some, such as most process effects, are effectively fixed [11]. Lumping all of these effects into one category of

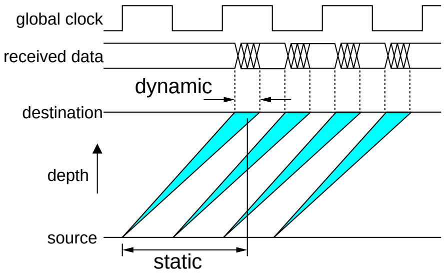


Figure 3. Timing uncertainty in globally synchronous wave pipelining (GSWP)

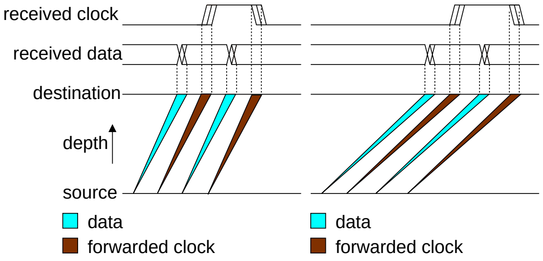
“timing uncertainty” is not always appropriate. For example, when considering time separation between edges representing successive bits in SSWP, *only those noise sources which operate in a time range comparable to the time separation between bits need to be considered.*

This paper classifies noise sources as contributing to either dynamic or static timing uncertainty. Sources contributing to dynamic timing uncertainty must influence timing on a cycle-to-cycle basis. For this paper, this means a time scale of roughly 1ns or smaller. Crosstalk and fast supply noise are the two biggest sources of dynamic uncertainty [11] and both are considered in this paper. The other noise sources described above vary slowly enough to be considered static, even if they include a time varying component such as temperature.

3.1 Globally synchronous wave pipelining

Figure 3 illustrates timing uncertainty in GSWP, wave-pipelined systems that use a global clock (similar to a figure in [17]). Timing constraints can be derived by bounding arrival times at the destination. Both forms of timing uncertainty, static and dynamic, must be considered. To sample the data reliably, setup and hold constraints must be met on each global clock edge. Due to data transport latency, there is no guarantee that the data arrives exactly aligned with the global clock edges, so phase alignment circuitry must compensate for any skew between the global clock and received data. Dynamic timing variation causes edges to arrive outside of their expected times, resulting in a window of uncertainty that narrows the data eye opening. The clock rate must be chosen slow enough to keep the eye aligned with the compensated clock edge. In addition, consecutive data edges must be spaced far enough apart to not interfere with each other through inter-symbol interference (ISI).

Keeping the receivers sampling clock phase aligned with the incoming data represents an additional challenge for circuit design and reliability. The details of how phase alignment is done are crucial to the reliability of the circuit. Also, proper modeling the link reliability is sensitive to these circuit design details. For these reasons, we do not consider GSWP any further in this paper.



(a) Fast static timing (b) Slow static timing

Figure 4. Timing uncertainty in source-synchronous wave pipelining (SSWP)

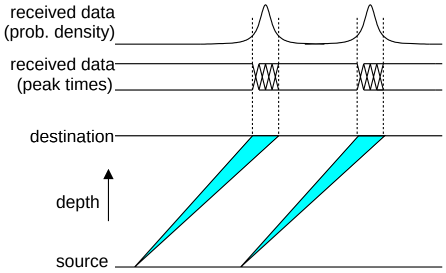


Figure 5. Statistical timing uncertainty

3.2 Source-synchronous wave pipelining

Figure 4 shows timing uncertainty in SSWP, a source-synchronous wave-pipelined link (similar to a figure in [2]). There is no global clock; instead, the receiver samples the data using a forwarded clock that is sent alongside the data. Timing constraints in this case are not derived from the global clock, but instead are derived from the relative timing between the final received clock and data signals.

The figure highlights the differences between static and dynamic timing. The static delay through a link will vary from chip to chip, but for a given link it is fixed. There will still be timing uncertainty due to dynamic effects, such as crosstalk and power supply noise, but the mean relative timing between edges (either between data and clock edges, or between consecutive edges on the same wire) does not change from cycle to cycle. Two scenarios are illustrated; the first shows a fast chip, while the second shows a slow chip. Despite the wide range in mean arrival times, the relative timing between consecutive edges and between data and clock edges is the same regardless of the static timing.

Because clock and data paths travel on very similar paths, they experience very similar delays. However, the

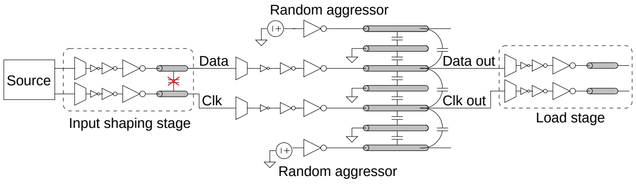
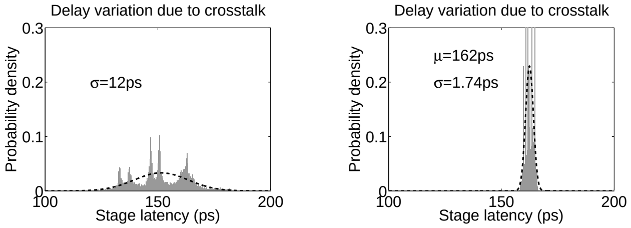


Figure 6. Crosstalk simulation circuit



(a) No shielding (b) Full shielding

Figure 7. Delay variation due to crosstalk

paths will not, in general, be identical due to OPC differences, difference in neighboring layout structures, data-pattern dependent delays, etc., resulting in one path being slightly faster or slower than the other. Furthermore, these differences may include systematic or correlated effects that apply at every stage, resulting in static skew that accumulates linearly with link length.

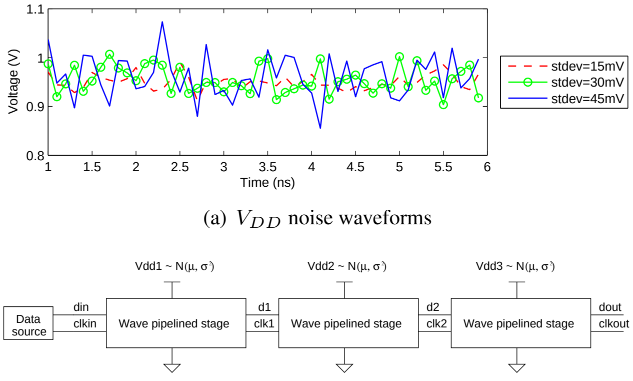
Unlike GSWP, reliability is independent of static timing variation in SSWP. However, dynamic timing variations will still affect the timing spread of each individual bit just as for SSWP, influencing both reliability and throughput.

3.3 Statistical timing model

The timing diagrams in the previous section are a convenient way to visualize timing uncertainty. At first glance, they imply a worst-case, bounded timing model. For example, with SSWP we simply need to bound the arrival times to guarantee: (1) clock and data arrival times do not overlap, and (2) consecutive data and clock edges are spaced far enough apart such that they don't interfere.

Bounded models are by their nature conservative, especially for long wave pipelined links in which timing uncertainty can accumulate. The bounded model can only say if a circuit is safe given a worst-case noise estimate. We believe a probabilistic noise model, shown in the top line of Figure 5, is more useful in this context, since it can be used to derive the probability of failure for a given amount of injected timing uncertainty. This allows us to compare the robustness of different pipelining techniques when subjected to the same input noise conditions.

To do so, we model the skew as a random variable S which is normally distributed with zero mean and unknown standard deviation σ_s. The jitter, which affects consecutive edge separation, is modeled as a random variable E with zero mean and unknown standard deviation σ_e. In Section 4, HSPICE measurements are used to estimate these values. Section 5 then uses these random variables in a statistical timing model to show how the bit error rate of a link may be estimated as a function of the data period.



(a) V_{DD} noise waveforms

(b) Test circuit used to measure delay

Figure 8. Experimental setup to measure delay impact of V_{DD} noise

4 Quantifying dynamic timing uncertainty

The goal of this section is to determine the delay impact of the two biggest sources of dynamic timing uncertainty, crosstalk and supply noise, in SSWP. The delay impact of each source is measured from HSPICE simulations in a CMOS 65nm process and approximated as a normal distribution to estimate the standard deviations of jitter and skew, σ_e and σ_s, respectively. In addition, we demonstrate that timing uncertainty accumulates with link length.

4.1 Crosstalk

Crosstalk is often modeled with changes in the coupling capacitance between two neighboring wires, which is a function of the voltage on each wire. This model is useful for producing worst-case bounds on crosstalk-induced delay variation. Section 3 discussed why probabilistic bounds are more useful; this is true when discussing crosstalk as well if there are many possible sources of crosstalk and transition times are not known ahead of time, which is the case if the link can be circuit switched.

4.1.1 Simulation setup

Probability distributions for crosstalk-induced delay were estimated by applying random data onto neighboring aggressor wires as shown in Figure 6. The latency through one wave-pipelined stage from the clock input to clock output was measured about 10,000 times using different random aggressor data. All signal wires including aggressors are twice minimum width. In the unshielded case, all wires are spaced apart by twice the minimum spacing, while in the shielded case, minimum-width shields are inserted between each pair of signal wires at the minimum spacing. All wires are assumed to be in one of the middle metal layers. Coupling capacitances are determined from an HSPICE field solver using process data; second-order coupling capacitances (i.e. from a signal wire through a shield to the next signal wire) are included, and account for about 3% of total capacitance. The data wire carries a 16-bit pattern while the clock wire has an edge corresponding to each bit.

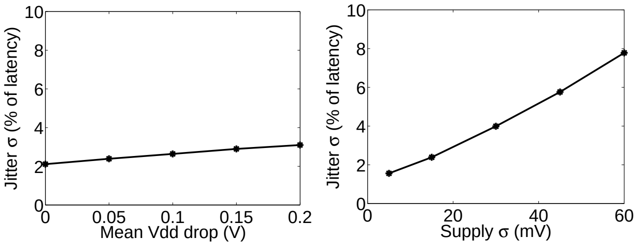
4.1.2 Results

The resulting delay histograms are shown in Figure 7. The curves are not normally distributed because of deterministic coupling between wires. Also, a slight mismatch between rising and falling edges leads to double-peaked behavior. When the behavior is fit to normal curves, the standard deviation of the delay is $\sigma_e = 12$ps for the unshielded case, while in the shielded case $\sigma_e = 1.74$ps. The reduction due to shielding is sufficient to suggest that shielding should always be employed for wave-pipelined interconnect.

4.2 Supply noise

There are many ways to model supply noise. Some models include slow sinusoids in the 100–500MHz range [4] to model resonance in the LC circuit formed by the power grid and the lead inductance. One study of ASICs measured power supply noise and found a mixture of deterministic noise caused by the clock signal and its harmonics, random cyclostationary noise caused by switching logic, and random high frequency white noise [1]. A recent study suggests that decoupling capacitors can remove this high frequency noise, so the supply should be considered a constant DC voltage [6]; we believe this to be unrealistic and include high frequency noise in our simulations.

In the context of high-speed wave-pipelined links, slowly varying or constant changes in supply noise will not impact the dynamic (i.e., cycle-to-cycle) timing; instead they will affect the static timing. To assess the static and dynamic effects independently, supply noise will be modeled as the sum of a nominally fixed DC component and a fast transient component. The transient noise is assumed to be a memoryless random process which is normally distributed and changes value every 100ps; this rate was chosen because it has a strong impact on cycle-to-cycle delay at the bit rates in this paper. The mean or DC level, μ, is nominally 1.0V; noting that low supply voltages limit performance more than high supply voltages, our analysis focuses on DC voltage levels below the nominal. The standard deviation σ is left as a parameter. Figure 8(a) shows example power supply waveforms at $\mu = 0.95$V DC and $\sigma = 15$mV, 30mV, and 45mV; supply voltages like these are applied in circuit simulations. Note that $\sigma = 45$mV leads to 3σ variations of $\pm 0.135V$, or $\pm 13.5\%$, which is more pessimistic than typical bounds of $\pm 10\%$.



(a) DC noise response ($\sigma = 15$mV, μ =variable)

(b) Transient noise response $\mu = 0.95V$, σ =variable)

Figure 9. Delay variation due to supply noise

4.2.1 Simulation setup

Simulations were conducted to measure the impact of DC and transient supply noise on delay. A multi-stage link is constructed and the stage latency is measured over several thousand trials. Figure 8(b) shows the measurement circuit. Each stage has an independently varying supply voltage, but all supply voltages have the same distribution parameters, with mean or DC value μ and standard deviation σ.

To measure the impact of transient noise, the DC value is fixed at 0.95V, and transient noise ranging from $\sigma = 0$mV to $\sigma = 60$mV is applied. To measure the impact of DC noise, a small fixed amount of transient noise ($\sigma = 15$mV) is applied, and DC voltage is varied from 1.00V to 0.80V.

4.2.2 Results

At each supply voltage tested, the delay measurements are plotted as histograms and fit to normal curves (not shown due to space constraints; the normal distributions fit quite well, which is unsurprising because the input supply noise was also normally distributed.) Figure 9 shows a summary of the histogram data; the standard deviation, σ_e, of the jitter, is plotted against the amount of added noise. The curves are plotted as a percentage of the stage latency, which is 165ps at $V_{DD} = 0.95V$. In this case, a $\sigma_e = 5\%$ jitter corresponds to $\sigma_e = 8.3$ps per stage.

The trend lines clearly show that jitter increases steadily with applied transient noise but is relatively insensitive to changes in DC value. Slow changes in the DC voltage level will thus have relatively little impact on cycle-to-cycle jitter.

4.3 Jitter and skew propagation

If a normally distributed timing uncertainty with standard deviation σ_e is applied at each stage, then the uncertainty at stage k should be $\sigma_e\sqrt{k}$; we are pessimistically assuming the timing uncertainty is independent at each stage. The previous simulations measured the uncertainty at one stage; simulations in this section measure it for eight-stage links and extrapolate out to fifty-stage links in order to confirm this assumption. We limited our simulations to eight stages because of the large simulation times for performing thousands of trials.

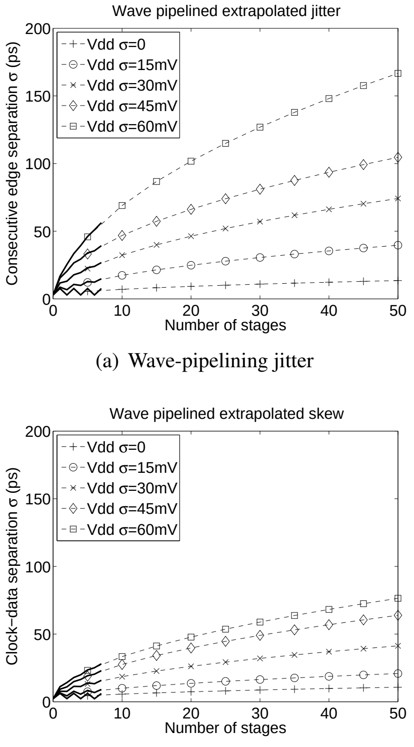
Instead of measuring the latency through a stage as was done when preparing Figure 9, we will now directly measure the separation of consecutive edges and the skew between the strobe and the data. Because these measurements are the difference between two varying edges, the measured standard deviation will be $\sqrt{(2)}$ times larger than what was computed in Figure 9.

4.3.1 Simulation setup

The simulation setup is similar to the setup in Figure 8, except a longer link with 9 stages is simulated. The goal of this simulation is to measure the skew and jitter at the output of each of the first 8 stages to determine the relationship with respect to link length. The ninth stage provides a load for the output of stage 8. We ran one hundred trials with each trial containing sixteen measurements. Ideally more trials would be run, but this would require longer simulation times than we could complete.

4.3.2 Results

The jitter and skew measurements produce histograms with a normal distribution at a certain mean and standard deviation (not shown). Mean skew and jitter is constant, but the standard deviation varies both with the amount of noise applied and with the length of the link. Figure 10 shows the standard deviation of jitter and skew. Simulation data is marked on the graph with a thick line,. At very small levels of noise, the curves are jagged because disparities between rising and falling edges are the dominant source of skew and jitter. The simulation data is also fit to curves of the form $y = A\sqrt{x} + B$, where x is the number of stages. The curves show a good fit to the simulation data and we extrapolate them out to 50 stages as shown with dashed lines. Table 1 gives the standard deviations (A in the equation; the B terms are small). For comparison, the table includes the jitter measurements from Figure 9, which have been scaled by $\sqrt{2}$ because they measured uncertainty for one edge, not the difference between two edges.



(a) Wave-pipelining jitter

(b) Wave-pipelining skew

Figure 10. Jitter and skew propagation

4.4 Summary

This section decried techniques for quantifying dynamic timing uncertainty due to crosstalk and fast supply voltage noise in terms of their standard deviations, assuming they are normally distributed. If the timing uncertainty applied to each stage has a standard deviation of σ, the uncertainty at stage i is shown to have a standard deviation of $\sigma\sqrt{i}$. From Table 1, we also notice that $\sigma_e \approx 1.8\sigma_s$. Hence, jitter is larger in magnitude than skew.

Table 1. Standard deviations of timing uncertainty

	Units in ps		
Supply σ (mV)	Predicted σ_e (Fig 9$\times\sqrt{2}$)	Extrap. σ_e (Fig 10a$\times\sqrt{2}$)	Extrap. σ_s (Fig 10b$\times\sqrt{2}$)
15	5.5	5.7	2.7
30	9.2	10.7	5.8
45	13.3	14.8	9.3
60	18.0	21.5	11.0

5 Estimating reliability

Knowledge of the standard deviation of the dynamic uncertainty allows us to estimate link reliability. Table 2 lists the parameters and variables used in this analysis. There are two timing constraints that must be met as illustrated in Figure 11. First, the separation between two consecutive edges (nominally equal to the period, T) must be greater than some minimum edge separation; otherwise, clock or data events could be lost. The minimum edge separation, t_{sep} was measured from simulation using the approach in [15] and was found to be about 160ps in the worst case. Second, the time separation between the data edge and the clock edge, which is nominally equal to half the period, must be larger than the setup time at all latches in the link. Using a simple logical effort [14] model, the latch setup time was estimated to be about 20ps.

We define $P_{E,I}$ to be the probability of an error due to intersymbol interference resulting from a violation of the minimum edge separation, and $P_{E,L}$ to be the probability of an error due to incorrect sampling at any latch. A successful transmission requires that neither error condition occur. We can thus define P_E, the overall probability of error, to be probability of an error due to ISI or due to sampling failure, so that $P_E = P_{E,I} + P_{E,L} - P_{E,I} \cap P_{E,L}$. The two events are likely dependent and positively correlated, such that the existence of one type of error increases the likelihood of the other. Because the dependency is difficult to estimate, we will pessimistically assume the two conditions are independent and make use of the approximation $P_{E,I} \cap P_{E,L} \approx P_{E,I} \cdot P_{E,L}$.

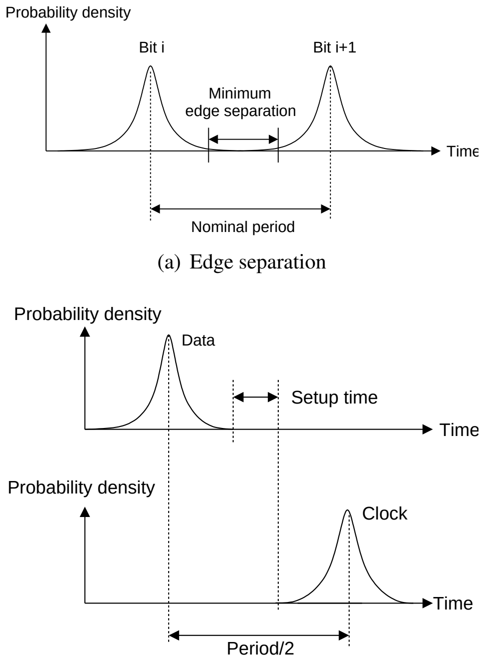
Because we are assuming normal distributions, the tails are unbounded. We cannot guarantee that the constraints will never be violated, but we can determine the probability of error and design the link so that it is sufficiently low. A link operating continually at 3GHz experiences about 10^{16} events per year. There may be thousands of such links on a chip, and we may wish to have a mean time between failure of hundreds of years. Thus, we require the overall probability of error to be in the range of $P_E = 10^{-20}$ to 10^{-25}.

5.1 Calculating probability of error

In this section, we calculate the probability of error for GSLP, SSWP and SSWPL using statistical timing methods.

5.1.1 Probability of error for GSLP

Assume there are n stages between latches. If each stage has a latency of t_{stage}, then the latency between latches is $n \cdot t_{stage}$. Given a setup time of t_{su}, clock skew of t_{skew}, the period T must be chosen such that $T < (n \cdot t_{stage} + t_{su} + t_{skew})$. For convenience, we will define $t_{static} = n \cdot t_{stage} + t_{su} + t_{skew}$.



(a) Edge separation

(b) Skew

Figure 11. Timing constraints

Let $P_{E,L1}$ be the probability of error due to a sampling failure at any one latch. If dynamic timing uncertainty S is added, where S is a random variable representing added skew, then the link must satisfy $T < (t_{static} + S)$. Thus, $P_{E,L1} = P(T < (t_{static} + S))$, or, equivalently, $P(S > (T - t_{static}))$. Because we assume S is a random variable with zero mean and known standard deviation σ_S, the calculation is straightforward. We can evaluate the probability of error at stage k by replacing σ_S with $\sigma_S\sqrt{k}$.

Because there are latches every n stages and the link is q stages long, an error occurs if any latch fails, such that $P_{E,L} = 1 - (1 - P_{E,L1})^{\lceil q/n \rceil}$. In a GLSP link, $P_{E,I} = 0$ because only one edge is present at a time between two latches, so $P_E = P_{E,L}$.

5.1.2 Probability of error for SSWP and SSWPL

For source-synchronous wave pipelined links, we have a similar error condition as above, but must also include the possibility of an error due to intersymbol interference.

In SSWP, the only latch is at the receiver, and the clock is generated at the source and travels along with the data. We require only that the nominal sampling window, set to half the period, $T/2$, be greater than the total dynamic uncertainty S, plus any static skew between the data and clock lines, plus the latch setup time. Nominally we assume there is no deterministic static skew, but there may be a random component due to correlated within die vari-

Table 2. List of parameters and variables

Symbol	Name	Value	Source
t_{sep}	Min. consecutive edge separation	160ps	65nm HSPICE simulation
t_{stage}	Stage latency	160ps	65nm HSPICE simulation
t_{setup}	Latch setup time	20ps	Estimate/logical effort
t_{latch}	Latch latency	50ps	Estimate/logical effort
t_{skew}	Global clock skew	10ps	Estimate [9]
T	Data period	160ps to 1ns+	Calculation
n	Number of stages between latches	1 to 50	50 stages = 25mm cross-chip link
q	Number of stages in link	1 to 50	50 stages = 25mm cross-chip link
$P_{E,L}$	Probability of error due to failed sampling	10^{-25} to 1	Calculation
$P_{E,I}$	Probability of error due to intersymbol interference	10^{-25} to 1	Calculation
P_E	Probability of error, overall	10^{-25} to 1	$P_{E,I} + P_{E,L} - P_{E,I} \cdot P_{E,L}$
σ	Standard deviation of supply noise	0 to 60 mV	Model assumption
σ_e	Standard deviation of dynamic jitter	0 to 160ps	HSPICE simulation (Fig. 10a)
σ_S	Standard deviation of dynamic skew	$\approx \sigma_e/1.8$	HSPICE simulation (Fig. 10b)
σ_{S2}	Standard deviation of static skew	2% link latency	Model assumption
S	Dynamic skew (normal random variable)	$N(0, \sigma_S)$	Model assumption
E	Dynamic jitter (normal random variable)	$N(0, \sigma_e)$	Model assumption
E_2	Static skew due to variation (normal random variable)	$N(0, \sigma_{S2})$	Model assumption

ation. We can use another random variable, S_2, to represent this skew, which we assume has zero mean and is normally distributed. In that case, we have $P_{E,L} = P(T/2 > S + S_2 + t_{setup})$.

The dynamic uncertainty S at stage k has zero mean and a standard deviation of $\sigma_S\sqrt{k}$. If we assume at worst a 2% change in latency per stage, then the static skew S_2 at stage k has a standard deviation of $\sigma_{S2} = 0.02 \cdot t_{stage} \cdot k$. Because the two variables are independent, the overall skew at stage k (i.e. the sum of $S + S_2$) has zero mean and a standard deviation of $\sqrt{\sigma_S^2 k + (0.02 \cdot t_{stage} \cdot k)^2)}$. The probability of error, $P_{E,L}$, can then be calculated using the formula above.

We must also consider $P_{E,I}$, the probability of an error due to intersymbol interference. Nominally, the edge separation is T, the data period; we require simply that $T > t_{sep}$. If we add dynamic timing uncertainty E, then the probability of error is $P_{E,I} = P(E < (T - t_{sep}))$. At stage k, E has zero mean and standard deviation $\sigma_e\sqrt{k}$. The overall probability of error, P_E, is $P_{E,I} + P_{E,L} - P_{E,I} \cdot P_{E,L}$ as discussed above. For a small number of stages, say less than 30, we note that $P_{E,I}$ dominates the overall probability of error. However, for longer links, the linear skew term causes $P_{E,L}$ to dominate.

In SSWPL, latches are periodically inserted to remove skew. To keep our model simple, we reset the skew term to 0 after a latch is encountered, so that $P_{E,L}$ is similar to the GSLP case; however, jitter accumulates throughout the whole link, as in the SSWP case. In such cases, the jitter-induced failure captured in $P_{E,I}$ dominates.

5.2 Results

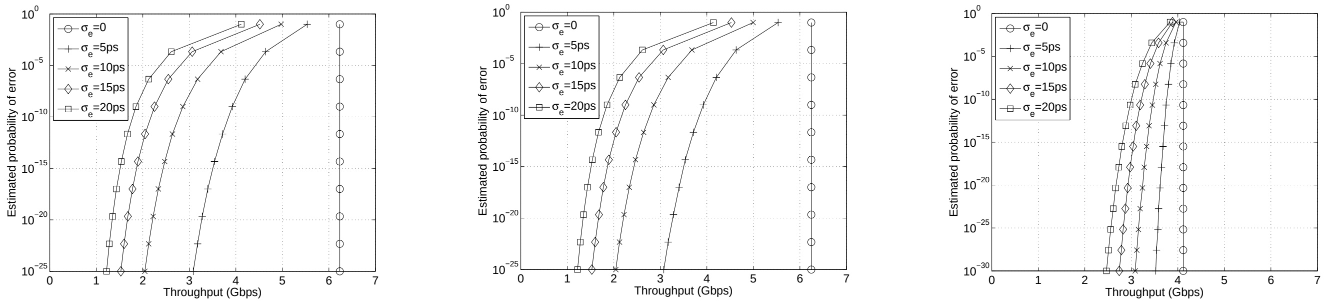
This section presents the reliability and throughput of wave pipelining and latch pipelining as a function of q, the total link length, n, the number of stages between latches, and σ_e, the dynamic jitter. Dynamic skew, σ_s, is set to $\sigma_e/1.8$ to follow the results in Table 1.

Figure 13 shows the error probability as a function of throughput at a fixed link of length $q = 10$ stages. With no noise ($\sigma = 0$), wave pipelining achieves 6.2Gbps and latch pipelining reaches 4.8Gbps. As noise is added, the achieved throughput for a given error rate very quickly drops in SSWP. However, GSLP is much more resilient. Also, there is little improvement from the additional latches in SSWPL showing that the wave pipelined designs are limited primarily by loss of edge separation rather than by skew.

Figure 14 shows the error probability as a function of the link length q at a fixed noise level of $\sigma_e = 10ps$. In this case, SSWPL outperforms SSWP when the number of stages is large, because the linear skew term is reset in SSWPL, but grows to dominate in SSWP. The GSLP graph demonstrates the expected result that throughput drops quickly as latches are placed farther apart.

Figure 12(a) presents the same throughput data at a fixed probability of error of $P(E) = 10^{-25}$ with no dynamic noise. In this case, wave pipelining has much higher throughput than latch pipelining. This result agrees with most previous work which claims that wave pipelining offers superior bandwidth. However, we note that linear skew term (a static effect) eventually causes SSWP performance to drop when the link gets too long.

With noise, however, the situation changes dramati-

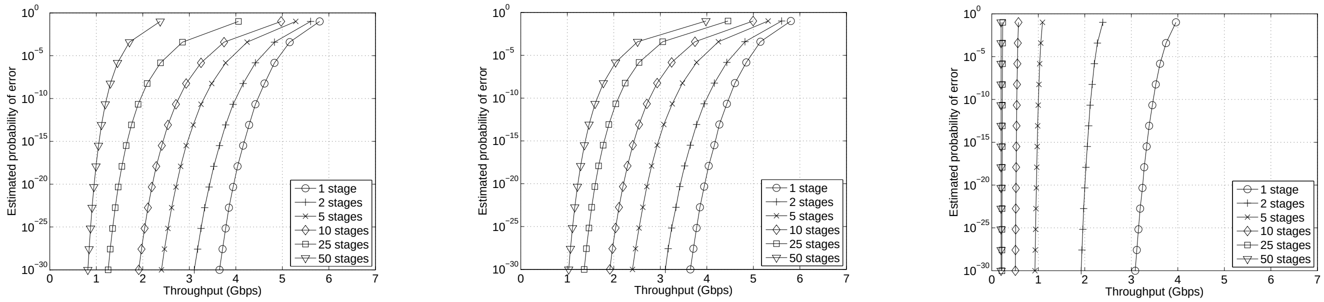


(a) SSWP

(b) SSWPL, latch every 5 stages

(c) GSLP, latch every stage

Figure 13. Probability of error estimates, fixed link of 10 stages (σ_e varied, $\sigma_s = \sigma_e/1.8$).



(a) SSWP

(b) SSWPL, latch every 5 stages

(c) GSLP, only one latch at the end of link

Figure 14. Probability of error estimates, fixed noise ($\sigma_e = 10$ps, $\sigma_s = 5.5$ps per stage)

cally. Figure 12(b) presents the same throughput data with $\sigma = 10$ps of noise. Throughput degrades for both wave pipelining and latch pipelining. However, latch pipelining throughput remains independent of the link length, while wave pipelining throughput degrades rapidly. This result indicates latch pipelining throughput is superior to wave pipelining in the presence of noise.

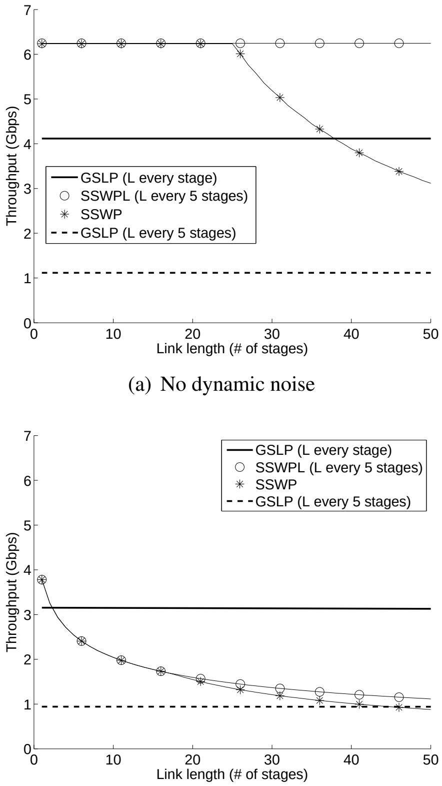
It is an open question as to how much dynamic timing uncertainty is realistic. These results are probably quite pessimistic due to the very large amounts of supply noise added. Nevertheless, these results show that designing without taking dynamic noise into account may lead to expectations for high throughput and robustness that are unachievable on a real chip.

6 Conclusions

Wave pipelining has attracted attention of NoC researchers due to its promise of high bandwidth using simple circuits and relatively low power. However, wave pipelining is much more susceptible to timing uncertainty than traditional interconnect pipelining techniques based on latches or flip-flops. This is because timing uncertainty accumulates across the entire link for a wave-pipelined design; whereas for a synchronous link, the uncertainty is reset or partially reset at each flip-flop or latch.

Statistical models of timing uncertainty are required to obtain a realistic assessment of the impact of timing uncertainty on wave-pipelined interconnect because using simple simple, worst-case assumptions at each stage produces a highly pessimistic model. We used HSPICE simulations to develop such a model for a switched interconnect in a 65nm CMOS process and then used this model to analyze the throughput achievable with wave-pipelined and latch-pipelined interconnect. When dynamic timing uncertainty was ignored, the wave-pipelined interconnect offers about 50% greater bandwidth than traditional synchronous designs. When jitter from noise is taken into account, wave-pipelining rapidly loses its advantage and the traditional synchronous design then offers higher bandwidth. The limiter for the wave-pipelined design was loss of the strobe pulses due to jitter; thus, inserting latches clocked by the forwarded strobe to the wave-pipelined design did not improve its performance.

Finally, our analysis shows the importance of quantifying interconnect performance at realistic bit-error-rates (BERs). For BERs of 10^{-3}, about the limit of what is practical to observe with circuit simulators such as HSPICE, wave pipelining offers much higher throughput than what can be achieved at more useful BERs of 10^{-20} or 10^{-25}.



(a) No dynamic noise

(b) Noise σ=10ps

Figure 12. Throughput versus link length at a fixed probability of error, $P(E) = 10^{-25}$

For synchronous pipelining, the loss of performance with decreasing BER is much less severe than for wave pipelining. It may be possible to mitigate these limitations of wave-pipelined interconnect by using error-correcting codes, but this would increase the complexity, area, power and latency of the link. Alternatively, our work motivates looking for design techniques that mitigate the accumulation of dynamic timing uncertainty for wave-pipelined interconnect. We see this as an important area for further research.

7 Acknowledgements

We would like to thank Suwen Yang, Andy Ye, Terrence Mak, Alastair Smith, the Actel architecture team including Jonathan Greene and Sinan Kaptanoglu, and the anonymous reviewers for valuable feedback. This work was funded by NSERC.

References

[1] E. Alon, V. Stojanovic, and M. Horowitz. Circuits and techniques for high-resolution measurement of on-chip power supply noise. *IEEE J. Solid-State Circuits*, 40(4):820–828, 2005.

[2] R. Dobkin, A. Morgenshtein, et al. Parallel vs. serial on-chip communication. In *Proc. Int'l. Workshop System Level Interconnect Prediction (SLIP'08)*, pages 43–50, Newcastle, United Kingdom, 2008.

[3] R. Dobkin, Y. Perelman, et al. High rate wave-pipelined asynchronous on-chip bit-serial data link. In *Proc.* 13^{th} *IEEE Int'l. Symp. Async. Circuits Systems (ASYNC'07)*, pages 3–14, 2007.

[4] J. Jang, S. Xu, and W. Burleson. Jitter in deep sub-micron interconnect. In *Proc. IEEE Comp. Soc. Symp. VLSI*, pages 84–89, 2005.

[5] A. Joshi, G. Lopez, and J. Davis. Design and optimization of on-chip interconnects using wave-pipelined multiplexed routing. *IEEE Trans. VLSI Systems*, 15(9):990–1002, 2007.

[6] S. Kirolos, Y. Massoud, and Y. Ismail. Power-supply-variation-aware timing analysis of synchronous systems. In *Proc. IEEE Int'l. Symp. Circuits Systems (ISCAS'08)*, pages 2418–2421, 2008.

[7] C.-S. Li and D. Messerschmitt. Statistical analysis of timing rules for high-speed synchronous interconnects. In *Proc. IEEE Int'l. Symp. Circuits and Systems (ISCAS'92)*, pages 37–40 (vol. 1), 1992.

[8] F. Li, D. Chen, et al. Architecture evaluation for power-efficient FPGAs. In *Proc. ACM/SIGDA* 11^{th} *Int'l. Symp. FPGAs (FPGA'03)*, pages 175–184, Monterey, California, USA, 2003.

[9] P. Mahoney, E. Fetzer, et al. Clock distribution on a dual-core, multi-threaded itanium/sup /spl reg//-family processor. In *Dig. IEEE Int'l. Solid State Circuits Conf. (ISSCC'05)*, page 292599 Vol. 1, 2005.

[10] T. Mak, C. D'Alessandro, et al. Global interconnections in FPGAs: modeling and performance analysis. In *Proc. Int'l Workshop System Level Interconnect Prediction (SLIP'08)*, pages 51–58, Newcastle, United Kingdom, 2008.

[11] S. Nassif, K. Bernstein, et al. High performance CMOS variability in the 65nm regime and beyond. In *IEEE Int'l. Electron Devices Meeting, (IEDM 2007)*, pages 569–571, 2007.

[12] N. Ou, T. Farahmand, et al. Jitter models for the design and test of Gbps-speed serial interconnects. *IEEE Design & Test of Computers*, 21(4):302–313, 2004.

[13] J.-C. Shyur, H.-P. Chen, and T.-M. Parng. On testing wave pipelined circuits. In *Proc.* 31^{st} *Conf. Design Automation (DAC'94)*, pages 370–374, 1994.

[14] I. Sutherland, R. Sproull, and D. Harris. *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.

[15] P. L. Teehan. Reliable high-throughput FPGA interconnect using source-synchronous surfing and wave pipelining. Master's thesis, University of British Columbia, 2008.

[16] L. Zhang, Y. Hu, and C.-P. Chen. Statistical timing analysis in sequential circuit for on-chip global interconnect pipelining. In *Proc.* 41^{st} *Conf. Design Automation (DAC'04)*, pages 904–907, 2004.

[17] L. Zhang, Y. Hu, and C.-P. Chen. Wave-pipelined on-chip global interconnect. In *Proc. Asia South Pacific Design Automation Conf. (ASPDAC'05)*, pages 127–132 (vol. 1), 2005.