Bus Delay Reduction by Bit Discarding

A.Bystrov

a.bystrov@ncl.ac.uk

NCL-EECE-MSD-MEMO-2011-004

27th June 2011, rev. 5/07/2011

The purpose of this report is to develop the mathematical baseline for a new idea of performance improvement of concurrent systems by discarding the slowest process. The mathematical result is illustrated by the application to parallel buses where one or several slowest bits are discarded. This is expected to have an effect on both the average and worst-case delay. One of possible applications of this research is to combine the bit discarding method with ECC fault tolerance. It is expected to find conditions under which the gains of bit discarding would outweigh the cost of ECC, thus making it meaningless to not use it. The mathematical analysis of the problem is expected to have value as a path to understanding of the role of variability in the timing closure of digital designs.

1 Introduction

The purpose of this report is to develop the mathematical baseline for a new idea of performance improvement of concurrent systems by discarding the slowest process. The mathematical result is illustrated by the application to parallel buses where one or several slowest bits are discarded. This is expected to have an effect on both the average and worst-case delay. One of possible applications of this research is to combine the bit discarding method with ECC fault tolerance. It is expected to find conditions under which the gains of bit discarding would outweigh the cost of ECC, thus making it meaningless to not use it. The mathematical analysis of the problem is expected to have value as a path to understanding of the role of variability in the timing closure of digital designs.

2 Mathematical analysis of the problem

The main idea is show how much latency improvement can be achieved by disregarding the slowest bit(s) of a parallel bus.

2.1 The state of the art – all wires complete their transactions

Consider a data bus having n wires, each wire being characterised with a value of propagation delay d. The wires are statistically independent with respect to the delay, and the delay is modelled with the Gaussian probability distribution function, or PDF, shown in (1), where σ^2 and d_0 are the variance and the mean value of d correspondingly.

$$\phi(d) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(d-d_0)^2}{2\sigma^2}} \tag{1}$$

For our analysis a normalised value of the mean delay is chosen ($d_0 = 1$). The analysis is performed for arbitrary values of variance and the number of wires in the bus. A significant limitation of the analysis method is the assumption of stochastic independence of the wires within the bus. In the real life there exist many physical factors causing a drift of parameters, usually monotonic w.r.t. different wires, such as the temperature or the power supply voltage. There also exist causes of non-monotonic variations, such as cross-talk or interference affecting a group of wires in the same time. These types of variations are disregarded for now and discussed later in the text.

A convenient representation of a stochastically defined delay in a wire is its cumulative distribution function, or CDF. A CDF, being an integral of PDF on the interval $]-\infty, d]$, gives us the probability of the transaction on this wire being completed by the time d. The CDF for a Gaussian PDF is shown in (2), where x is used instead of delay in order to distinguish between the integration variable and one of the integration limits. The function erf is commonly known as an error function and is defined in (3), where z and t are abstract variables.

$$\Phi(d) = \int_{-\infty}^{d} \phi(x)dx = \frac{1}{2}\left(1 + erf\left(\frac{d-d_0}{\sigma\sqrt{2}}\right)\right) \tag{2}$$

$$erf(z) = \frac{2}{\pi}\int_0^z e^{-t^2}dt \tag{3}$$

The plots of the Gaussian PDF and CDF functions for $d_0 = 1$ and $\sigma = 0.1$ are shown in Figure 1, which illustrates the idea of the average and the worst case delay on a set of wires. For clarity of the diagram, the worst case threshold is set not very close to 1. In real-life scenario it can be of an order of

$1 - 3 \cdot 10^{-18}$, which corresponds to one timing error in approximately 10 years on average whilst repeating the experiments with a frequency of 1GHz. If one needs to sample the delayed signal with a sufficiently high probability of a correct result, it needs to be done significantly later than the mean delay d_0, at the time labelled as the worst-case delay.

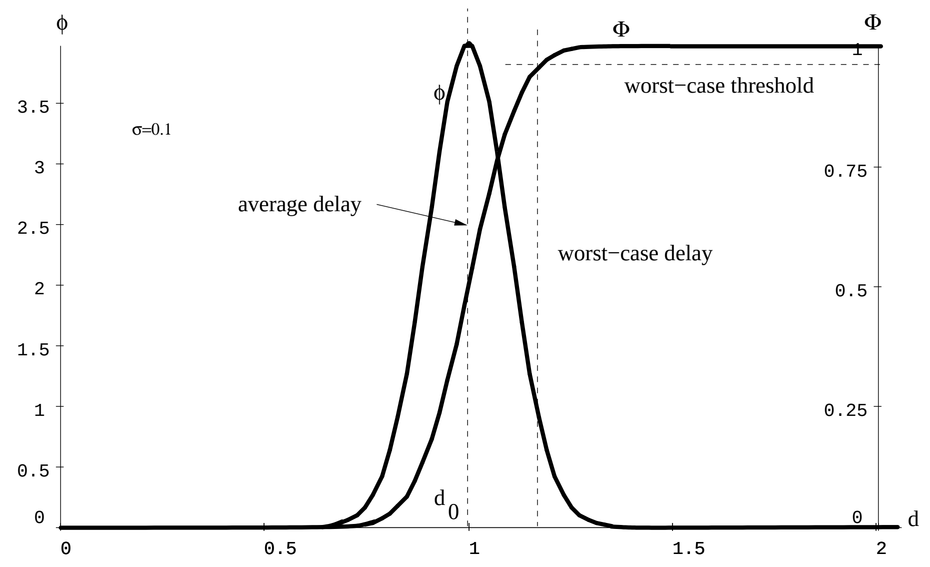


Figure 1: Average and the worst case delay on the distribution plots

Now, let us see how the CDF will change when a single wire is replaced with n stochastically independent wires in a bus. In order to deliver a data item error-free, all wires of the bus must complete the transaction before the data can be sampled. This will be a joint probability of independent events, which is a product of the probabilities of the component events. As the probability of each wire having completed the data transaction is represented as its CDF, the joint probability will be the product of the respective CDFs, and the result will also be a CDF as shown in (4), where n is the bus width and $\Phi_i(d)$ is the CDF for a given wire i, all wires having their own respective mean delay and variance.

$$\Phi_{all}(d) = \prod_{i=1}^{n} \Phi_i(d) \tag{4}$$

The plots of delay CDF for buses of different width and identical stochastic characteristics of each wire are shown in Figure 2. The effect of the bus width is very pronounced in the area close to the mean value of delay and is virtually non-existent near the worst case delay threshold. The latter can be explained by a very low probability of a signal not completing its transaction in the worst-case area. This makes the joint probability of two or several signals failing to finish by that time negligible.

The effect of increasing the average delay is most visible in the buses composed of wires with identical stochastic characteristics. In an extreme counterexample where one wire had PDF/CDF as in Figure 1 whilst all other wires have CDF shaped like the Heaviside step function, neither the average nor the worst case delay would depend on n.

2.2 Proposed approach – analysing the boundaries

The analysis of the experiment in Figure 2 shows that the worst case delay on a traditional bus is defined by the stochastic characteristics of a single bit which switches the last. This naturally leads to an idea of disregarding the slowest bit with the purpose of performance improvement. The lost bit can be recovered

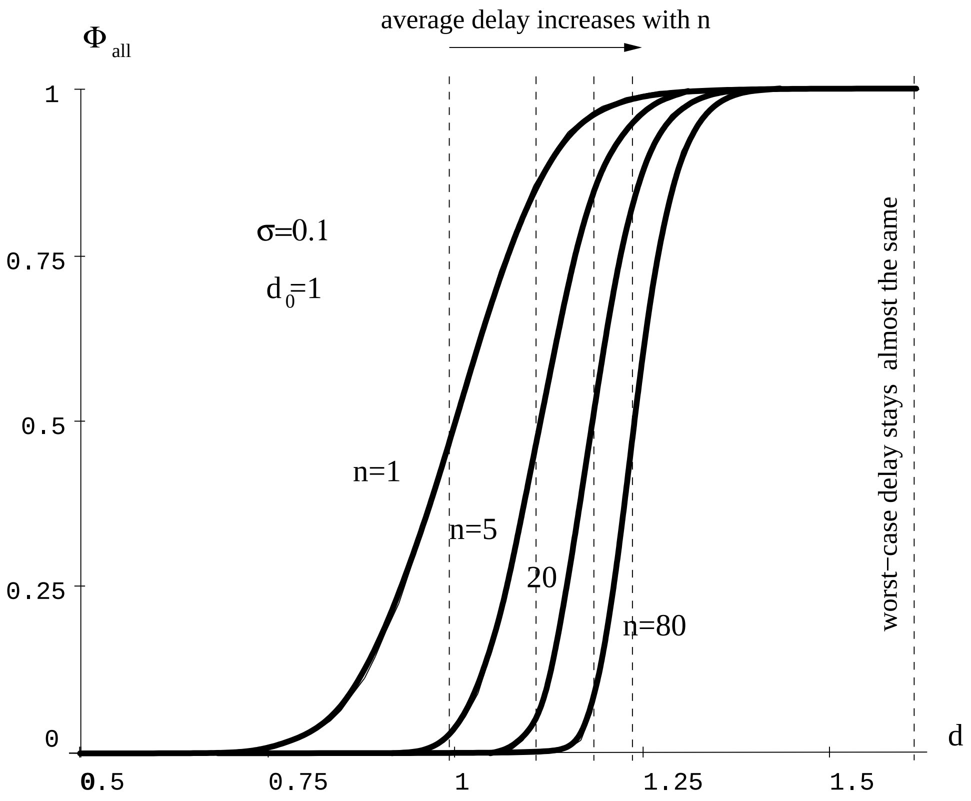


Figure 2: Delay CDF of a bus with *n* identical wires

by using error correction, which in the proposed approach will be performing two functions: correction of intermittent logic errors and also performance improvement by correcting the disregarded slow bits. The hope is that the resultant performance gain may outweigh the costs of the error correction resources above a certain level of variance, which tends to increase in deep submicron technologies and under extremely low V_{dd} operation characteristic to autonomous embedded and energy harvesting applications.

A boundary case of the shortest delay is easy to identify – this is where only the fastest bit is used and all other bits are disregarded. This case is very expensive and not very practical. It is only needed for comparison and evaluation of the other schemes. The probability of 1-of-n independent events can be calculated as a complement of the joint probability of the complements of the said events. Effectively it means taking the probability of one wire completing the transaction, calculating the probability of the same wire not completing (complement), then finding the probability of all wires not completing on time (joint probability), and, finally, calculating the probability that this joint event has not happen (yet another complement). This is done in (5), where the CDF Φ_i is used to express the probability of a selected wire i having completed its transaction by the time d. It is clear that the output of (5) is also a CDF.

$$\Phi_{at\,least\,one}(d) = 1 - \prod_{i=1}^{n}(1 - \Phi_i(d)) \qquad (5)$$

The bounds for the average and worst case delay calculated for a bus with 20 identical wires $n = 20$, $\sigma = 0.1$, $d_0 = 1$ are plotted in Figure 3, the lower bound being $\Phi_{at\,least\,one}(d)$ and the upper bound being $\Phi_{all}(d)$. By observing this plot one may arrive that a significant delay reduction can be achieved by disregarding the slowest bits of a bus. The greatest potential exists for improving the worst-case delay, which is most relevant to synchronous designs. More plots of the bounds derived for different values of n and σ are shown in the further sections.

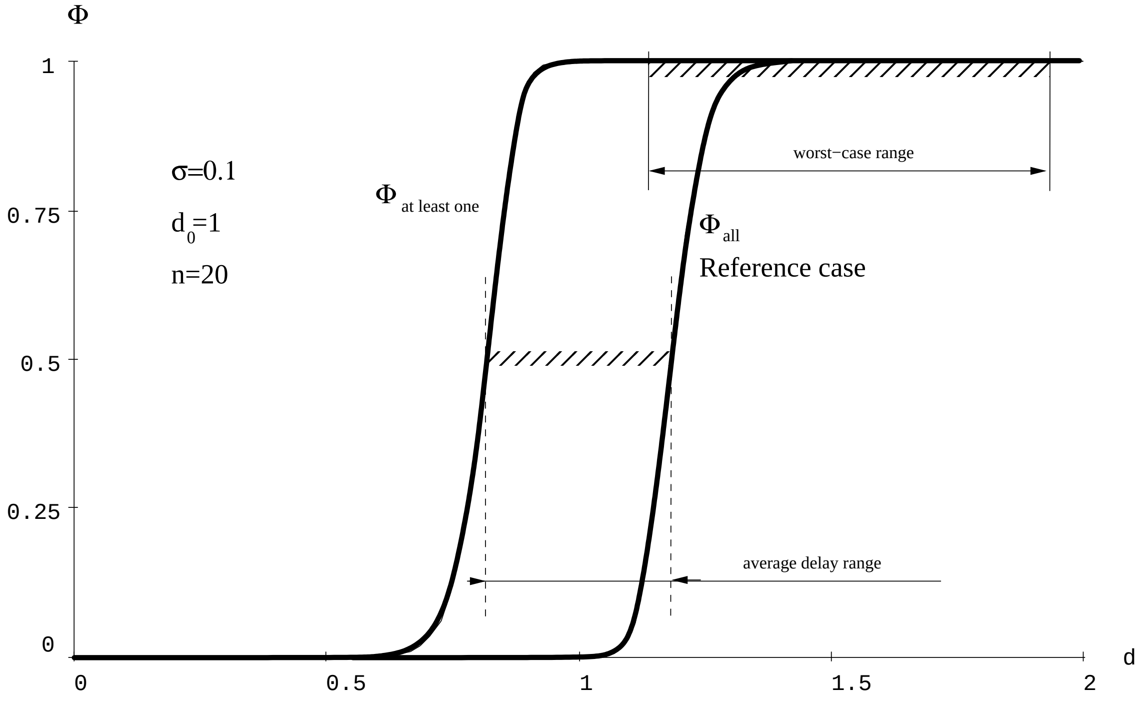


Figure 3: Delay bounds for a bus

2.3 Proposed approach – one slowest bit ignored

The plots of functions (4) and (5) in Figure 3 show a potential advantage of disregarding the slowest bits in the bus. However, the lowest delay bound was achieved by sacrificing almost all bits, which is an overly high price to pay. This conclusion naturally leads to the idea of disregarding just one slowest bit, which is analysed in this section.

The probability of a single bit *i* to have competed the transaction by the fixed time *d* is defined as a CDF $\Phi_j(d)$. Therefore, $1-\Phi_j(d)$ is the probability of its failure to complete. The probability that $(n-1)$ wires complete, whilst one fixed wire does not, is $(1-\Phi_j(d))\prod_{i=1..n|i\neq j}\Phi_i(d)$. Now, observe that there exist $(n+1)$ ways to finish the transaction on the bus: *n* combinations when one late wire is disregarded and one option when all *n* wires have completed. This leads us to the formula (6) that combines the probabilities of all $(n+1)$ mutually exclusive outcomes.

$$\Phi_{ignore\,1}(d) = \sum_{j=1}^{n}\left((1-\Phi_j(d))\cdot\prod_{i=1..n}^{i\neq j}\Phi_i(d)\right)+\Phi_{all}(d) \tag{6}$$

The function $\Phi_{all\,but\,1}(d)$ is plotted with thick lines in Figure 4(a) alongside with the lower and upper bounds (dashed lines) defined as $\Phi_{at\,least\,one}$ and Φ_{all} correspondingly. Each of these functions is presented as a family of curves derived for the different bus widths $(n=5,20,80,320)$. One can see that all $\Phi_{ignore\,1}(d)$ curves closely follow their respective upper bounds, which are moving in the direction of longer delay with increasing the width of the bus. There is an improvement in the average delay for $\sigma=0.1$ (or variance $\sigma^2=0.01$) of approximately 9% for the 5-bit bus, which is gradually reduced to 3% when the number of bits reaches 320. The percentage of improvement depends on the value of σ, being almost proportional to it.

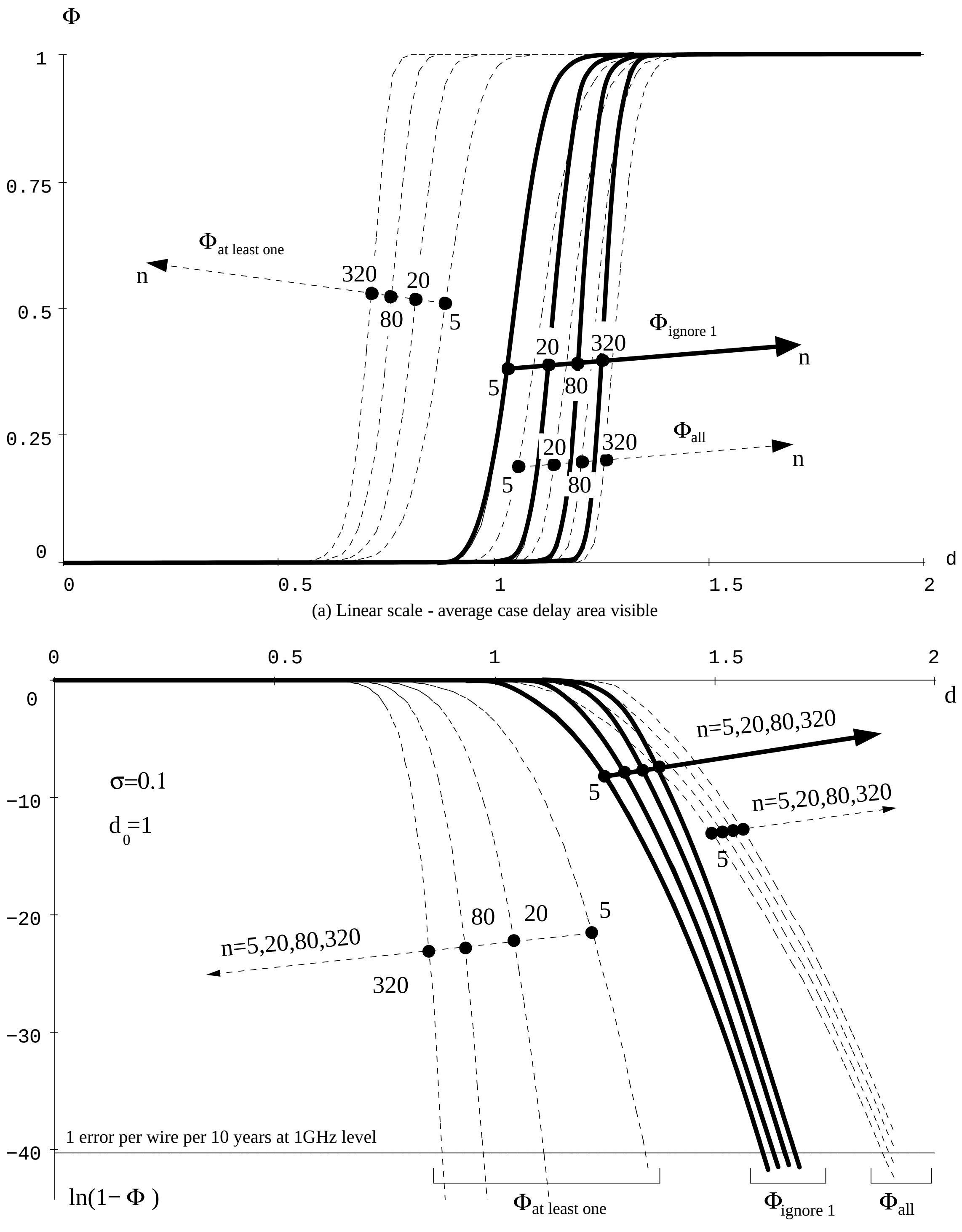


Figure 4: Delay CDF for a bus with one slowest bit disregarded

In order to be able to see the behaviour of the distribution functions in the area of the worst-case delay (where they are very close to 1), one needs to "zoom in" onto the corresponding part of the graph. For this purpose the function $\ln(1-x)$ is chosen, where x is substituted with the CDF-s. The transformed diagram is shown in Figure 4(b). The horizontal line in this plot is the threshold corresponding to the probability of an event that happens once in 10 years under the frequency of repeating the experiment 1GHz. Three families of depicted curves are the boundary cases (dashed lines) and the bus where the slowest bit is disregarded (thick solid lines). One can see that the distributions Φ_{all}, representing buses where all bits must complete, and $\Phi_{ignore\,1}$ of the proposed approach are very closely spaced within their families, which can be explained by a very low probability of more than one bits being unable to complete the transaction under a long delay. The worst case delay for the one-but-1 bus is approximately 16% better than the delay of the traditional bus. The spread of the CDFs for the boundary case $\Phi_{al\,least\,one}$ (where at least one bit completes) is much greater, because more bits – better the chance that at least one completes the transaction. One can also see that the percentage improvement increases when the threshold goes down, which means the probability asymptotically approaching '1'. One can also see that the improved average and the worst case delay values are still very far from the left boundary formed by $\Phi_{al\,least\,one}$, which leaves plenty of space for further improvement.

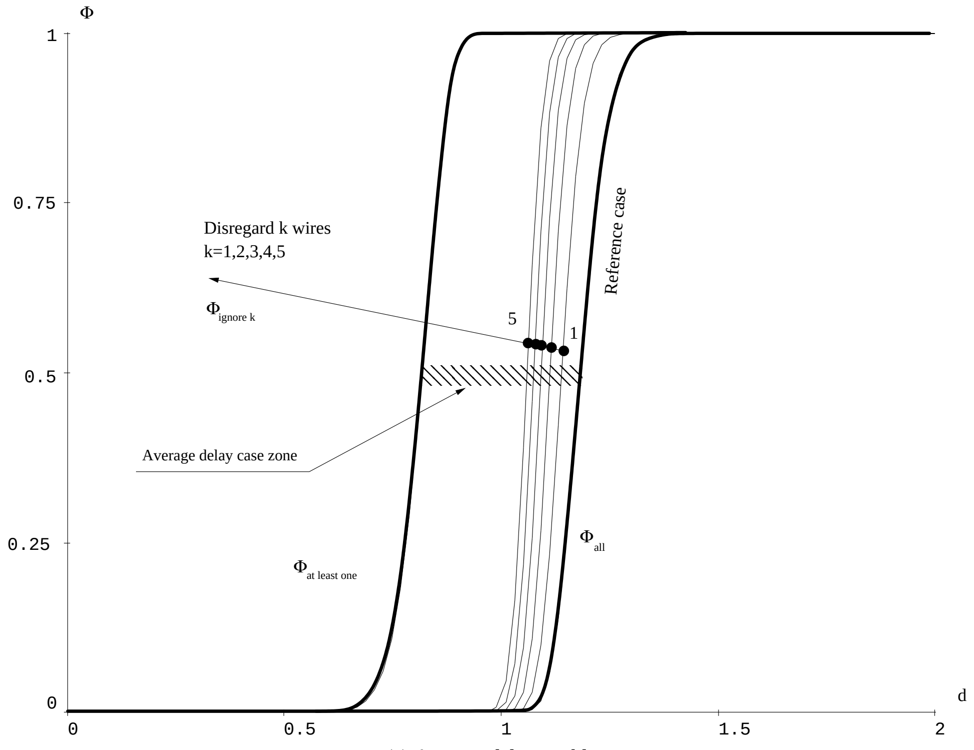
2.4 Proposed approach – k slowest bits ignored

A natural extension to the idea of disregarding one slowest bit of a bus is to disregard two, three, or in a general case k slowest bits. One can follow the flow of reasoning used in deriving equation (6), adding to it the probabilities of all 2-of-n, 3-of-n, etc. classes of combinations. The number of combinations in each class is a binomial coefficient $\binom{n}{k} = n! \cdot \frac{n!}{k! \cdot (n-k)!}$, where n is the bus width and k is the number of disregarded bits. The formula enumerating all possible combinations within the classes is awkward to write in a compact way, but easy to implement as an algorithm. Alternatively, one can assume the bus wires being stochastically identical, which will lead to a significantly simpler formula (7), where $\Phi(d)$ is the CDF for a single wire.

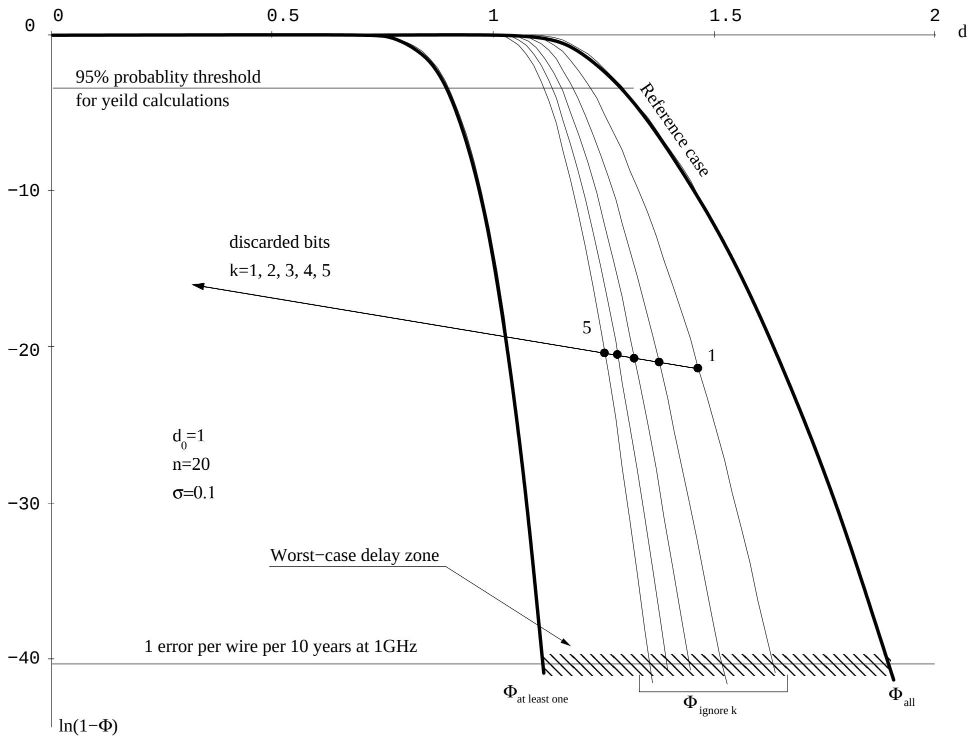
$$\Phi_{ignore\,k}(d) = \sum_{j=1}^{k} \binom{n}{j} \cdot \Phi^{n-j}(d) \cdot (1 - \Phi(d))^j + \Phi^n(d) \tag{7}$$

The function $\Phi_{ignore\,k}(d)$ is plotted is plotted in Figure 5 for $n = 20$, $\sigma = 0.1$ (variance $\sigma^2 = 0.01$) and the number of disregarded slowest bits $k = 1..5$. The plots also contains the bounds defined as $\Phi_{at\,least\,one}$ and Φ_{all} shown for reference. The plots show improvement in both the average and the worst-case delay. The worst case delay is shortened by 14%...27%, the most significant improvement obtained when disregarding one (14% improvement) or two bits (further 6% improvement). The average delay is shortened by 3.5%...10.5%, and the most significant improvement also obtained by disregarding one bit (3.5% improvement) or two bits (further 1.8% improvement). The effect of bit disregarding on the average delay is significantly smaller than the effect on the worst-case delay.

The efficiency of discarding the late bits is analysed in Table 1, where the percentage of ignored bits is compared to the relative improvement of delay. The calculations are performed for two dimensions of the bus, $n = 20$ and $n = 80$. The worst case delay probability threshold is the same as earlier, i.e. 1 error per 10 years of operation at 1Ghz. For a wider bus, of course, a single bit represents a smaller percentage of the wiring resource. The improvement in both average and worst case delay is similar for both values of n. This leads to a higher efficiency of the bit discarding method when applied to wider buses. The table also shows that there exist situations when trading the wiring resource for delay makes no sense. For example, the average case delay on the narrow bus ($n = 20$) improves only by 3.5% when 5% of the interconnection resource, which is proportional to throughput are sacrificed. This means that the bit discarding technique is not applicable (for the given values of n and σ) to, for example, self-timed circuit designs operating on the average delay case. For a greater value of n the situation is reversed, but the gain remains negligibly small. On the other hand, synchronous designs, which are operating on the worst case delay, can benefit



(a) Average delay visible



(b) Zoom on the worst-case delay threshold

Figure 5: Effect of disregarding k bits

Table 1: Relative delay improvement per percentage of disregarded bits

k	$n=20$			$n=80$		
	k/n, %	$\triangle d_{average\,case}$	$\triangle d_{worst\,case}$	k/n, %	$\triangle d_{average\,case}$	$\triangle d_{worst\,case}$
1	5%	3.5%	14%	1.25%	2.7%	13%
2	10%	5.3%	20%	2.5%	4.4%	19%
3	15%	7.0%	24%	3.75%	5.6%	23%
4	20%	8.8%	26%	5.0%	6.5%	25%
5	25%	10.5%	27%	6.25%	7.2%	27%

(For each wire: $d_0 = 1$, $\sigma = 0.1$ (variance $\sigma^2 = 0.01$))

significantly from trading wires for performance. The table shows, for example, that for $n = 80$ the speed can be increased by 13% by trading off only 1.25% of the interconnection resource. A more significant gain in the worst case performance is fundamentally possible by discarding more than one bit, e.g. 27% of performance bought with 6.25% of the wiring resource, but this might be difficult to implement due a need in expensive multiple error correction techniques.

A similar approach can be used for yield calculations. Figure 5(b) shows how much the bus delay can be improved by statically disregarding the slowest bit. This is done under an assumption that the wire delays are fixed at the time of fabrication and remain static. The static approach is widely used in memory design, where one or several defective columns in the matrix can be disconnected and replaced with redundant columns at the time of production testing.

3 Searching for use cases

3.1 Combined use with ECC

As mentioned in the Introduction, a possible application of the bit discarding method is its combination with ECC in designs having increase variance of wire delay, e.g. deep submicron technologies, future nanotechnology, energy harvesting designs, low voltage circuits, etc. A way to approach this pilot study is to take several standard bus sizes with and without standard ECC with Hamming codes, and calculate the variance at which the performance gain (due to allowing intermittent timing errors) would outweigh the cost of the redundant wires. We assume that the bus throughput is proportional to the number of wires in it.

The width of the first chosen benchmark bus is $n = 64$. A popular SECDED (Single Error Correction Double Error Detection) Hamming code for this bus is $(72,64)$, which has 8 redundant bits.

The worst-case delay is calculated for both buses with and without redundant bits and for different values of σ. For the redundant bus one bit can be discarded ($k = 1$) and for the irredundant bus $k = 0$. The threshold of the probability is the same as used earlier: one timing error per wire per 10 years at 1GHz. The mean delay on the individual wires is again $d_0 = 1$. The plots of $\sigma(d_{worst\,case})$ are shown in Figure 6, where the value of throughput is calculated as the number of irredundant data bits divided by the worst-case delay (normalised, as in all previous plots). This is an “abstract” throughput, as it does not take into account any other delay except for the random independent delay variation between the bits – it is only suitable for a preliminary study. The curve “64 no ECC” is for the case of a 64-bit bus without ECC; the curve “(72, 64) ECC bit discarding” is for a bus protected with the (72, 64) Hamming code and timed to ignore one slowest bit; the curve “72 no ECC” is a 72-bit bus without ECC, where all 72 bits contribute to the data throughput. The ECC bus performs better than the others under high variance conditions, which means that the redundant wiring resource has stopped being a penalty. In fact, not using the added bits as parity bits becomes a penalty. The use of the error correction system to disregard the slowest bit does not affect its ability to correct the intermittent errors caused by any other factors, e.g. cosmic radiation, because the joint probability of an error caused by bit discarding and an error due to cosmic radiation is negligible.

3.2 Self-timed design

Asynchronous self-timed circuits use a special signal to acknowledge completion of each transaction. Therefore, they do not fail if one signal on a bus takes abnormally long time to propagate. Furthermore, the throughput of self-timed circuits corresponds to the average case delay. Such circuits can benefit from discarding the late bit. Figure 4(a) gives an overall idea of by how the average delay depend on the number of wires in a bus, and Figure 5 shows dependency of the average delay on the number of disregarded bits. A further improvement in the average delay can be obtained in the asynchronous designs using so-called monotonic or self-indicating codes, which are often used in order to distinguish between the transitional and final states for each data word (completion detection). Examples of such codes include 1-hot, padded 1-hot (thermometer code), m-of-n, dual-rail, etc. The number of bits switching in these codes is usually either half or less than a half of the bus width. As the choice of which bits to switch does not depend on the delay, the CDF of the bus is Φ^m, where Φ is the CDF of a single wire and m is the number of wires switching in each code word. From Figure 4(a) one can see that the expected average delay improvement is very small for the codes where a large number of bits are switching (e.g. dual-rail, $\binom{2m}{m}$, etc.), and slightly better for the codes with a small number of such bits (e.g. 1-hot). The latter class of codes will produce faster implementations at the expense of using more wires (can be exponentially more!). Some of self-timed codes have limited error correction properties (e.g. the thermometer code), which might be possible to exploit in the implementation of the slowest bit discarding technique. However, discarding a small number of late bits does not improve the average case delay much (see Figure 5(a)).

To conclude, it is unlikely that the slowest bit discarding method will be used in self-timed designs, unless the design uses the bundled data approach, which is based on the worst-case timing assumptions similar to synchronous designs.

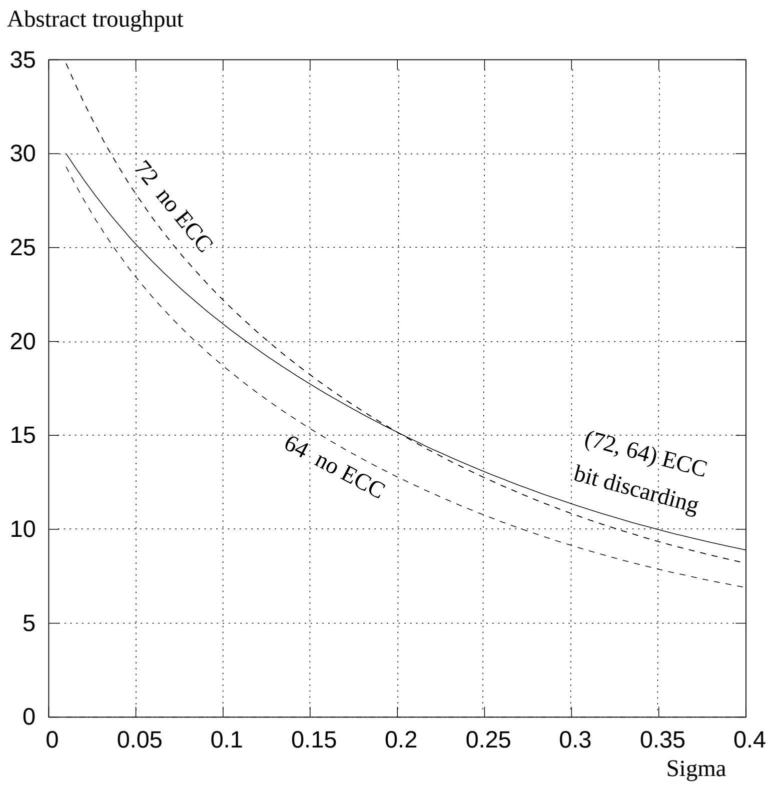
3.3 Gardbanding, Razor, memory, etc.

A significant limitation of the above analysis is complete disregard to any other contributors to the bus delay apart from random uncorrelated delay on the individual wires. In reality, there are many other delay components and guardbands added to the sampling delay (e.g. clock period). These components will make the gains of the bit discarding technique look less significant by reducing the relative improvement.

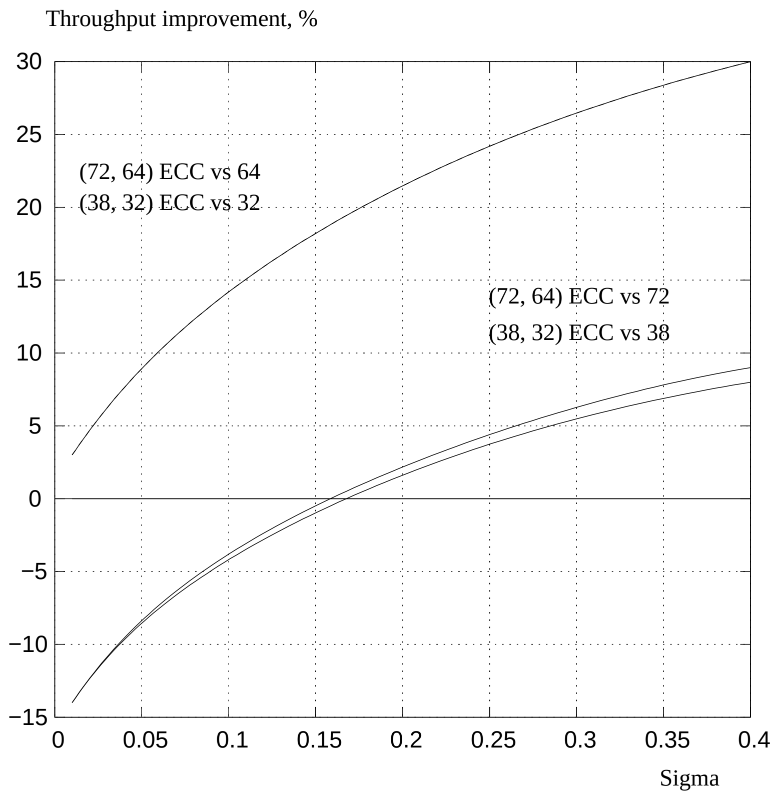
However, certain technologies such as Razor reduce the guard bands, thus making the proposed technique attractive. Furthermore, the ECC included into the bit discarding technique can be combined with Razor with the purpose of reducing the recovery delay (when Razor detects an error and falls back to the old data in the shadow register).

Various techniques using “elastic” clocks also aim at reduction of guardbands whilst relying on bundling delay to sample data. The delay elements are usually implemented as semiconductor devices characterised with very high levels of dynamic variability. This seems to be a good application area for the bit discarding technique as well.

One of possible applications of the proposed method is communication to and from memory. Memory cells may have huge variations in read delay and ECC is frequently used there, but not for performance improvement. This forms an attractive niche for the proposed method.



(a) Abstract throughput vs σ



(b) Throughput improvement by bit discarding with ECC

Figure 6: Efficiency of bit discarding with ECC

4 Conclusions

In this Report an idea of discarding the slowest bits on parallel buses (or other parallel structures) has been mathematically analysed and discussed. The main conclusion is that if only a small number of bits are discarded, then this mainly improves the worst-case delay, and does not improve much the average delay. This makes the method applicable to synchronous and “almost synchronous” (e.g. Razor and bundled data self-timed) design methodologies. One of possible applications of the method can be a memory bus, because the read access is usually subject to high delay variation due to nature of memory cells. The presented method of mathematical analysis can be applied to any specific configurations of buses and other parallel structures.