ABSTRACT

We present the implementation and analysis of a variation tolerant version of a switch-to-switch link in a NoC. The goal is to tolerate the effects of process variations on NoC architectures using self-correcting links that automatically detect delay variations and compensate them. The correction is applied without increasing the switch-to-switch latency by substituting the output flip-flops of the sending switch with a self-correcting flip-flop followed by an adaptive voltage swing selector. Higher delay variations will result in a smaller slack in the switch-to-switch path, but the adaptive voltage swing selector could mitigate its impact on the NoC communication by increasing the voltage swing on the link, thus allowing a compensation of the delay variation. As a result, it is possible to tolerate delay variations at the cost of additional power consumption.

1. INTRODUCTION

Due to silicon technology scaling effects, boosting clock frequency of monolithic high-performance microprocessors has clearly reached a point of diminishing returns. As an effect, we are experiencing a rapid shift towards massively parallel architectures with tens or even hundreds of cores integrated onto a single chip. Evidence of this trend is unmistakable as practically every commercial manufacturer of high performance processors is currently introducing products based on multi-core architectures [1, 2].

With the increasing number of cores, scalability of communication architectures becomes critical for system performance and traditional bus architectures are rapidly running out of steam [3]. We are moving from core-centric to interconnect-centric design flows and very likely buses will soon be replaced by a dedicated on-chip network providing point-to-point communications by means of dedicated routers [4].

As transistor dimensions continue to shrink, it becomes harder and harder to precisely control the fabrication process. Transistors and wires become faster, but less predictable and, after fabrication, it is becoming more and more difficult to guarantee timing closure, and hence performance, across multiple copies of the chip [5].

With traditional worst-case design techniques, the risk is to be too conservative or to discard too many chips, thus resulting in very low levels of yield.

The emerging issue now is how to design differently than worst-case [6]. Self-calibrating designs have been introduced as an alternative to worst-case characterization of silicon [7]. Instead of relying on over-conservative worst-case assumptions, self-calibrating circuits tune their operating parameters to in-situ actual conditions.

A run-time controller receives feedback from a checker that monitors correct operation of the circuit. When needed, circuit reliability can be restored at some power or performance cost, depending on actual silicon capabilities and noise conditions.

A milestone work exploring the use of self-calibrating links in the context of on-chip networks is [8]. It proposes dynamic voltage-swing scaling as a way to dynamically adapt to environmental variations, design uncertainties, manufacturing parameter deviations or communication bandwidth requirements.

While their transmission scheme adapts to variations of link physical parameters by discovering the real delay-voltage characteristic of the technology without any assumptions on it, there is no connection between link operating conditions and parameter variations affecting upstream or downstream functional units. In practice, self-calibrating links cannot prevent an upstream switch from violating its timing requirements due to delay variations.

In contrast, our work pioneers the use of digital self-calibrating links to compensate process variation induced delays in the upstream logic unit. We do not view the communication channel in isolation, but its operating parameters can be adapted to absorb and compensate the effect of process variations in connected modules.

We also select link voltage swing as our compensation knob. We envision a design methodology where process variations of link upstream modules are measured during the post-fabrication test phase, and clock skew of such modules tuned accordingly. Then we make up for the reduced propagation delay margins on the link through voltage swing adaptation.

We show that the energy overhead induced by the increased swing is offset by deploying low-swing signaling on the links under nominal conditions. This creates some margin for power savings which is the budget for process variation compensation during post-silicon testing and therefore for yield enhancement.

Our technique can also be used during the normal life of the circuit in order to address wear-out problems. Periodically, the chip could be re-tested and new problems caused by delay variations might arise. Our NoC links could be reconfigured in order to cope with the new conditions, thus resulting in the possibility to extend the lifetime of the chip by recharacterizing the links and their power consumption.

The rest of the paper is organized as follows. Related work are described in Section 2. Section 3 presents our guiding principles. Self-calibrating link implementation is then illustrated in Section 4. Finally, the experimental results in Section 5 explore the reliability - power trade-off and present an application case study of our technique to the NEC NoC prototype. Conclusions are drawn in Section 6.
2. RELATED WORK

Several variation-tolerant design techniques have been proposed so far and a good survey can be found in [9].

Circuit design techniques based on replacing regular flip-flops with more redundant versions able to increase fault tolerance have recently gained momentum as an alternative to methods relying on worst-case characterization of silicon. In particular, the Razor [10] technique has been proposed for catching dynamic errors. Razor can not only detect and correct errors, but can also allow the design to operate at optimum power and performance. The Razor technique is power efficient because it does not replicate all the hardware, but only those flip-flops that are critical and require checking for correctness. This technique is also capable of catching circuit marginalities arising from transistor performance variations. In addition, algorithmic techniques at the functional-block level might be applicable to improve noise marginalities as well ascope with dynamic variations, thus improving reliability.

Razor is compelling due to its low area overhead, but it can only operate correctly in a limited range of delay errors. The two extreme conditions can easily be explained: (a) if the signal arrives on time, the delayed latch detects a false error; (b) if the signal arrives too late, even the delayed latch will sample it incorrectly. To overcome this limitation, checker architectures [11] have been proposed in order to enhance the Razor scheme with the capability to detect near critical conditions.

A timing error tolerant design methodology for NoCs is presented in [12]. The normal FIFOs used in the network components (links, switches and network interfaces) are replaced by T-error FIFOs that add support for timing error recovery using razor flip-flops and backpressure links. In presence of delay errors, the correction can be performed using one extra cycle. In the meantime, the backpressure link gets activated in order to slow down the previous FIFO stage.

ReCycle [13] is an architecture-level technique complementary to circuit-level ones like Razor. It addresses process variations in processor pipelines. In presence of variations, some pipeline stages become faster, while others become slower. To mitigate these effects, time slack can be transferred from the faster stages to the slower ones. As a result, the pipeline can be clocked with a period close to the average stage delay rather than the longest one. ReCycle increases the frequency of a pipeline without changing its structure or depth, or the speed of the transistor. Moreover, donor (empty) pipeline stages can be added to the critical loop in the pipeline to “donate” slack to slow stages, enabling a higher pipeline frequency.

ReCycle has been very inspirational to this work. Our technique is based on tuning the skew of the clock signal in every switch-to-switch link of our NoC in order to compensate for delays caused by process variations. The amount of slack can be selected from a pool of available slacks provided by an enhanced razor flip-flop. The concept of slack donor stages introduced in ReCycle can also be applied to our NoC links. Long links could be segmented into shorter ones by means of registers and the length of those segments could be selected based on a planned amount of extra slack that we want to incorporate in the specific link. With respect to [12], our technique does avoid the need of one extra cycle for performing the correction. The error is corrected in the same cycle by increasing the clock skew and by compensating it, when needed, with an increased voltage swing on the link. Our technique could be enhanced further by the adoption of coding techniques like those proposed in [11], but this has not been the focus of this work.

3. SELF-CORRECTING FLIP-FLOP

A recently proposed technique to deal with process variations in pipelined designs consists of clock cycle time stealing, where the time slack of faster stages in the pipeline is transferred to the slower ones by skewing the clock arrival times to latching elements.

In a different context, a similar scenario occurs also in on-chip networks. Whenever the switch and the link can be viewed as two consecutive pipeline stages, a critical path delay degradation in the switch could be compensated through slack stealing from the link. However, there is a big difference between the NoC and the microarchitecture scenario. In this latter case, if not enough slack is available in other pipeline stages, donor pipeline stages should be inserted, which are empty stages added in the pipeline to donate slack to the slow stages. A larger flexibility does exist for NoC links, since such slack could be provided without increasing link latency (e.g., by means of pipelined links). Link operating conditions provide the needed tuning knob to adapt the link propagation delay to the skew applied to an upstream logic stage. In particular, electrical schemes to reduce the voltage swing of the interconnect are known and well studied [14]. Obviously, the variable voltage swing impacts the speed at which the interconnect driver is able to charge or discharge the load capacitance. In the past, this has been exploited to identify voltage/frequency pairs for reliable link operation in presence of changing environmental conditions or application requirements.

In this work, we intend to apply the slack stealing technique to switch-link pipelines in on-chip networks for process variation tolerant operation. In practice, we propose to skew the clock of the switch output latches to compensate for variation induced gate delays. We then propose to make up for the reduced margins for signal propagation across the link by increasing the voltage swing. Since the effects of process variations are typically known only during the post-silicon testing, we need a self-calibrating mechanism in the link that tunes clock skews and drives the proper voltage swing to the link driver and receiver.

One of the main trends in the design of self-calibrating circuits is to empower a selected subset of standard flip-flops with error detection and correction capabilities. Figure 1 shows the architecture of the self-correcting FF that we have designed to recycle switch-link pipelines in on-chip networks. If the incoming data, due to process variations, arrives late, the main FF will sample it incorrectly, but the circuit is equipped with a certain number of additional detection and correction paths that will be able to tolerate delays up to approximately 50% of the clock period.

Our enhanced self-correcting flip flop has two execution modes, with well differentiated performance and power requirements: calibration mode and normal execution mode.

In calibration mode, an operating point controller compares the output of the main and of the delayed-sampling flip-flops, assuming that the worst-case transition delay of the input signal is within the correct sampling window of the flip flop with the most delayed clock. Contrarily to similar circuits that cope with dynamic timing errors[10], our enhanced flip flop needs not only to detect timing violations, but also to quantify them. In fact, the voltage swing on the link will have to be adjusted accordingly, and we should take care of not over-designing the voltage swing to avoid link power inefficiencies. In contrast, similarly to [10], we need to cope with metastability. This scenario, where multiple delayed FFs sample the same input signal, is particularly sensitive to metastability, and we accounted for this in the controller design.

The controller compares all FF outputs (MAIN, S1, S2) with that of S3. When the output of a FF differs from S3 or is metastable, then the next one in the timing chain is selected for reliable opera-
results in the possibility to effectively cope with wear-out effects. By recharacterizing the links and their power consumption, this could be reconfigured in order to cope with the new conditions, caused by delay variations might arise depending on the age and during normal execution mode.

It is important to stress that no power overhead is caused by our self-calibrating link. All FFs that do not belong to the selected path can be powered down using either clock- or power-gating techniques in order to save power. Active circuits during both calibration and execution modes are illustrated in Fig. 1(a) and (b) respectively. In addition, switches are easily testable as far as we are concerned. In fact, they usually exhibit a well identifiable number of critical paths going through the arbitration stage and through the crossbar selection signals. Moreover, a large timing gap does exist between the critical and the non-critical paths, thus making SSTA easier.

After calibration, the circuit enters in its normal execution mode. All FFs that do not belong to the selected path can be powered down using either clock- or power-gating techniques in order to save power. Active circuits during both calibration and execution modes are illustrated in Fig. 1(a) and (b) respectively. It is important to stress that no power overhead is caused by our self-calibrating link during normal execution mode.

Periodically, the chip could be re-tested and new problems caused by delay variations might arise depending on the age and the history of operating conditions of the chip. Our NoC links could be reconfigured in order to cope with the new conditions, thus resulting in the possibility to extend the lifetime of the chip by recharacterizing the links and their power consumption. This results in the possibility to effectively cope with wear-out effects.

### Table 1: Truth table of the operating point controller.

<table>
<thead>
<tr>
<th>S2</th>
<th>S1</th>
<th>MAIN</th>
<th>SELECTED FF</th>
<th>Sel1</th>
<th>Sel2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MAIN</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>S1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>S2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>S3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

3.1 Applicability

As regards applicability of our technique, a wide application domain can be envisioned in the NoC context. The requirement we pose on the NoC architecture is to expose the switch-link interface as a logic pipeline. As a consequence, switches implementing input buffering or virtual output queuing can be made variation tolerant with our circuit technique provided output latching is implemented. In principle, this might lead to an increased communication latency through the network, but a lack of output latching might give rise to long signal paths degrading global design performance. Circuit-switching architectures like the one reported in [16] are ideal candidate for our self-correcting FFs, since they implement end-to-end flow control and feature retiming stages at switch inputs and outputs. Finally, our technique applies to output-buffered switches as well with a few modifications. First, the input to output buffers should feed our enhanced FF as well. During self-calibration, the controller selects the proper voltage swing for the switch output links. Moreover, it should drive a multiplexer that selects a proper delayed version of the clock for the whole output buffer. Second, input latching may be required to break long timing paths. In this paper, we will demonstrate applicability of our variation-tolerant scheme to the NEC NoC in Section 5.4.

4. LINK DESIGN

Let us consider at first all circuit components that are active during normal execution mode and their relative design issues. We opted for full SPICE modeling of the whole communication channel, in order to accurately characterize the reliability-power trade-off. Our design effort can then be capitalized to develop cells for technology libraries to feed ASIC synthesis tools. We used 90nm Berkeley Predictive Technology device models for our design and analysis framework.

A number of well-known schemes for low-swing signaling have been presented in the open literature[14]. After evaluation of several alternatives, we selected the same low-swing transmission scheme as in [17], and adapted it to our needs. The circuit schematic is reported in Fig. 2, except for the multiplexer whose implementation was straightforward and which we could consider as a simple delay stage in what follows.

The interconnect is driven by a transmitter that takes the input stream from a flip flop. The receiver is composed by an amplifier and an output latch. We deployed a Sense-Amplifier Flip-Flop[18] scheme optimized for high-performance. It incorporates a precharged sense amplifier followed by a symmetric latch topology that significantly reduces delay and improves driving capability. The transmitter performs voltage-level conversion for low-swing signalling. The chosen receiver is a pseudo-differential circuit (PDFF), consisting of a clocked sense amplifier and a static NOR output latch. It uses single wire per bit while at the same time retaining most of the advantages of differential schemes, namely low input offset and good sensitivity. Its major reliability degra-
dation comes from the local device mismatch between the double input transistor pair, which usually can be controlled very well. The variation between distant REF’s of the driver and the receiver also contributes some reliability degradations. The operation of the receiver is not sensitive to the supply noise, as opposed to other schemes. Since the FF samples on the rising edge of the clock and the receiver on the falling edge, it is possible to apply a clock signal with a phase shift between these two modules in order to increase the link throughput and/or to let the receiver sample more stable input signals. In order to simplify the clock network, we fed the \texttt{PDIFF} receiver with the negated clock, thus giving the signal an entire clock period for propagation across the link. A π/3 model was used for electrical level simulation of the interconnect\cite{19}, due to the reasonable trade-off between computation complexity and accuracy.

Our SPICE characterization of this interconnect scheme for 90nm technology indicated a maximum operating frequency of 2 GHz (2mm wire, 200mV voltage swing). This frequency is much larger than those achievable by state-of-the-art NoC prototypes, which hardly achieve more than 1 GHz. Therefore, we set our nominal performance constraint to an aggressive value of 1 GHz. For transistor sizing, we used the optimization engine that comes with HSPICE. For each component in the channel, we kept requesting increasing performance to the optimizer till an abrupt increase of transistor sizes was returned by the optimizer. An interesting trade-off is implicit in the sizing process. On one hand, we would like to have large slacks so to be able to allow more slack stealing to the upstream switch. On the other hand, large transistor sizing results in more power consumption and area overhead. The driver was resized for different interconnect lengths and hence wire load capacitances. We considered link lengths ranging from 0.5 up to 4mm, which reflects typical NoC wiring scenarios.

\[ CNTRL \rightarrow S0 \rightarrow S3 \rightarrow OUT \rightarrow MUX \rightarrow OUT \]

\[ FF \]

\[ CLK \]

\[ 50\% \]

\[ S3 \]

\[ Inv \]

\[ Inv \ P \]

\[ Out 1 \]

\[ Out 2 \]

\[ Out 3 \]

\[ Sel 1 \]

\[ Sel 2 \]

\[ Figure 3: Operating point controller \]

Finally, area breakdown is illustrated in Fig.5. We can see that controller performance has been achieved at a significant area cost. Driver, receiver and FFs account for an almost equal share of area.

\[ Figure 4: Critical path breakdown \]
Total area incurred by the enhanced self-correcting FF amounts to 5 $\mu m^2$. Of course this depends on the number of instantiated FFs (4 in this case, as in Fig. 1 A).

5. EXPERIMENTAL RESULTS

5.1 Mode power

Our first experiment intends to characterize power consumption of our self-calibrating link in calibration and normal execution mode. We chose a target 2mm link and instantiated 4 FFs in the enhanced self-correcting FF. In all modes, the link works at 1 GHz.

In calibration mode, measured power was 0.388 mW. Please consider that this value includes not only link power, but the power of the entire communication channel, from the enhanced FF to the PDIFF receiver and NOR latch. Full swing voltage was enforced to meet the single cycle specification on calibration execution. In contrast, in normal mode the enhanced FF consumes like a normal flip flop, and power goes down to 50 uW. In this condition, the voltage swing was kept at 200mV, since even with 50% skew the nominal swing was able to meet timing constraints. Even raising the voltage swing to 1V, in normal mode power would be bounded to 0.15 mW.

5.2 Link power margins

The next experiment intends to characterize the margins for slack stealing made available by low-swing signaling interconnects and the link power overhead as a function of clock skew of the upstream logic stage. The plot in Fig.6 reports power consumption of our low-swing self-calibrating channel in three configurations and for different wire load scenarios. The leftmost bars indicate power when no clock skewing is applied, and hence the minimum voltage swing of 200mV is used. The rightmost bar reports power in the same case, but when voltage swing is arbitrarily raised to the full swing 1V value. Finally, the intermediate bar describes the case where the maximum tolerable skew is applied in each configuration (up to a maximum of half the clock period), and the relative power is highly impacted by the voltage swing needed to compensate that skew.

First, let us observe the first and the third columns. It is evident that low-swing signaling achieves a much smoother power behavior as a function of increasing wireload capacitance. The main difference is due to the interconnect capacitance being switched at its maximum swing. In practice, by implementing low-swing signalling in on-chip links we can accumulate power budgets that could be erased by process variations later on. In fact, the intermediate bar illustrates the way in which such budget is consumed by clock skew. For small wireload capacitances (from 0.5mm to 2mm links), the budget needs not to be consumed, since clock skew simply reduces the available slack. For longer links, such slack is not enough, and voltage swing needs to be increased with a significant reduction of power margins.

5.3 Voltage swing selection policy

The third experiment aims at deriving a selection policy of the proper voltage swing on the link as a function of clock skew. The plot in Fig. 7 shows the power incurred by a link as a function of increasing clock skew in order to be able to continue working at the target frequency of 1GHz in normal execution mode.

A few things are evident. First, as the link length increases, even raising the voltage swing to 1V does not allow to compensate up to 50% clock skew. In contrast, for small wire loads the clock skew is compensated by absorbing the available slack, without the need for voltage swing raising. Second, as soon as the voltage swing is increased, power goes up rapidly. So, when possible, it might be better to drive a given wireload capacitance with a slightly overdimensioned driver instead of having a weak driver which is then forced to raise the voltage swing to compensate large clock skews. Unfortunately, this was not feasible for the 4mm link, since the optimizer would have returned overly large transistor sizes to appropriately drive its wire load. As a result, with 350ps skew, power incurred by the link is more than 4x with respect to the zero skew case.

5.4 Case Study

We applied our process variation compensation technique to the NEC NoC architecture [20]. It is a heterogeneous tile-based archi-

![Figure 5: Area breakdown](image)

![Figure 6: Link power saving reduction with clock skew](image)

![Figure 7: Voltage swing selection strategy](image)
tecture where a two-dimensional fabric of tiles is connected to form a mesh or torus architecture (Fig. 8). Each tile typically consists of one or more bus based subsystems and each subsystem can contain multiple IP cores (processors, memory modules and dedicated hardware components).

In order to implement our technique, we replace each output flip of the upstream tile wrapper can be potentially compensated in the EQ output links.

We measured the area overhead of our self-calibrating link with respect to the area of the EQ module. The induced area overhead amounts to about 3%, considering the implementation of 3 FFs inside our enhanced self-correcting FF.

6. CONCLUSIONS

We present the implementation and analysis of a variation tolerant version of a switch-to-switch link in a NoC. The goal is to tolerate the effects of process variations on NoC architectures using self-correcting links that automatically detect delay variations in upstream logic units and compensate them. The correction is applied without increasing the switch-to-switch latency by substituting the output flip-flops of the sending switch with a self-correcting flip-flop followed by an adaptive voltage swing selector. As a result, it is possible to tolerate delay variations at the cost of additional power consumption. We proved the viability of our technique on a case study NoC architecture from NEC.

7. REFERENCES