Crosstalk Glitch Propagation Modeling for Asynchronous Interfaces in Globally Asynchronous Locally Synchronous Systems

Syed Rafay Hasan, Normand Bélanger, *Member, IEEE*, Yvon Savaria, *Fellow, IEEE*, and M. Omair Ahmad, *Fellow, IEEE*

***Abstract*—This paper characterizes the potentially catastrophic effect of crosstalk glitches on representative circuit implementations of two widely used asynchronous protocols. It is demonstrated that the crosstalk glitches can induce false events, which can undesirably propagate into asynchronous interface circuits and may cause system failure. Conventionally, to a circuit designer, glitch propagation (GP) due to aggressor-to-quiet-line crosstalk (AQX) in asynchronous handshake schemes can only be observed through circuit-level analysis/simulation. In this paper, circuit-level analysis is first performed to prove that even optimized conventional asynchronous circuits allow crosstalk glitches produced over moderate-length interconnects (1.5 mm) to propagate. This is a precursor to a more problematic crosstalk glitch occurrence due to further scaling of technologies. To warn the digital designers from GP due to AQX, a novel modeling technique is proposed. This modeling method works at the logic level to facilitate asserting asynchronous interface robustness to crosstalk glitches. This model can accurately identify the possibility of intrinsic (to the asynchronous interface) crosstalk GP in asynchronous circuits at the logic level and, hence, provides a foundation to formally verify such circuits. To our knowledge, this is the first work on modeling GP due to AQX at the logic level for asynchronous circuits.**

***Index Terms*—Asynchronous interface, crosstalk, deep submicrometer (DSM), multiple clock domains (MCDs).**

I. Introduction

MULTIPLE clock domains (MCDs) are inevitable constituents of complex system-on-chips [1]. Asynchronous interfaces are among the popular methods of interfacing modules within MCDs [1]–[4]. A system, in which individual modules are synchronous while the interfacing modules are asynchronous, is often called globally asynchronous locally synchronous (GALS) [5]. The asynchronous interfaces are conventionally defined at the gate or register–transistor logic levels [2], [3], [6], [7], [23]. Hence, some of the physical-level characteristics are ignored. Therefore, there is a gap to be bridged between these higher abstraction-level definitions of asynchronous interfaces and the problems associated with their physical-level implementation.

One of these physical-level problems is in relation to the rise in parasitic capacitance of interconnects in ultradeep-submicrometer (U-DSM) technologies, i.e., sub-100-nm technologies. Increased coupling capacitance due to technology scaling introduces unwanted crosstalk glitches in lines that affect the robustness of the system. Unwanted crosstalk glitches may produce false events, provided the glitch magnitude reaches the threshold value of subsequent gates [1]. Such a crosstalk glitch is called aggressor-to-quiet-line crosstalk (AQX) in this paper. The aggressor line (AL) is considered to be the one that inflicts crosstalk glitches, due to a transition of its own logic value, to an adjacent line. The signal line that is affected by glitches, in response to the aggressor, is called the victim line (VL) in this analysis, in compliance with the literature [8]–[10].

In U-DSM, reliability of signals is an issue of concern to designers [11], [27]. A false event in asynchronous systems, designed using U-DSM, may lead to system failure if subsequent signal transitions are triggered through this event. Hence, it makes AQX on a signal in event-driven self-timed designs (which encompass most of the asynchronous interfaces) a substantial reliability issue. AQX is more detrimental in asynchronous designs compared with synchronous designs, because clock frequency adjustment can take care of this effect in synchronous designs.

Conventional asynchronous circuits are designed to avoid data hazards and race conditions [12], [13], but design methods fail to address nonidealities introduced by crosstalk glitches. In [9], [10], [14], [15], [28], and [29], the crosstalk effects in DSM circuits were mitigated with the introduction of data bus encoding. Indeed, such encodings are very advantageous in alleviating delays introduced by crosstalk. On the other hand, crosstalk glitches on handshake signals in asynchronous interfacing circuits due to AQX cannot be resolved with this technique. In [16], a behavioral-level crosstalk detection methodology for sequential asynchronous circuits is developed. This design method checks for intrinsic transition faults in sandwiched wires only. However, in U-DSM technologies, as stated earlier, considerable glitches appear in quiet lines due to the transitions in ALs (AQX). Therefore, a need arises to investigate the conditions that model crosstalk glitch propagation (GP) due to AQX in asynchronous handshake schemes.

Manuscript received April 21, 2009; revised August 15, 2009 and October 30, 2009; accepted November 13, 2009. Date of publication February 05, 2010; date of current version August 11, 2010. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the Regroupement Stratégique en Microeléctronique du Québec (ReSMiQ). This paper was recommended by Associate Editor Y. Massoud.

S. R. Hasan and M. O. Ahmad are with the Center for Signal Processing and Communications, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: sr_hasan@ece.concordia.ca; omair@ece.concordia.ca).

N. Bélanger and Y. Savaria are with the Groupe de Recherche en Microélectronique et Microsystèmes, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada (e-mail: normand.belanger@polymtl.ca; yvon.savaria@polymtl.ca).

Digital Object Identifier 10.1109/TCSI.2009.2038553

In this paper, AQX effects in U-DSM are studied in the context of asynchronous interfacing protocols, and a GP model is proposed. To characterize the behavior of asynchronous circuits, due to AQX, a detailed analysis of the behavior of two protocols is performed. Circuit-level simulations of two typical protocols are used to validate this characterization. It is experimentally demonstrated that AQX can potentially lead to a sequence of erroneous transitions in asynchronous circuits. Hence, this characterization provides evidence that, if this phenomenon is not properly dealt with, then it may lead to the dire consequence of system malfunctioning. Currently, there is no mechanism or analysis method available other than circuit-level simulations that may detect such catastrophic outcome of AQX. No logic-level method is available to warn a design engineer before implementation of an interface exploiting asynchronous protocols. Toward that goal, we also propose a crosstalk GP modeling technique. The proposed technique includes a novel technology independent wire glitch element (WGE) to represent glitches at the logic level. The WGE translates the physical-level signals (glitches due to AQX) into pseudologic-level signals. A pseudologic-level signal can be treated at the logic abstraction level, although it does not necessarily do a full swing of the voltage levels. Utilizing AQX identification using the WGE, a novel GP set is proposed for each logic element (LE) to describe the conditions under which a crosstalk glitch may propagate. Leveraging these GP sets, a complete set of conditions is modeled for crosstalk GP. As these GP sets correspond to LEs, it makes these conditions applicable to most asynchronous handshake schemes used for interfacing MCD modules. This technique provides a framework that allows representing possible behaviors of a logic structure, in the presence of crosstalk glitches, without necessitating circuit-level simulations. This modeling can make a logic designer aware, at an earlier design flow stage, of the possible system conditions that lead to crosstalk GP. Alternatively, our proposed modeling method can be used to warn the physical designers so that measures can be taken to quench the potential AQX glitches in the identified vulnerable signals.

This paper is organized as follows: Section II describes the characterization of different asynchronous interfaces, in the presence of AQX, based on glitch magnitude. Section III illustrates the proposed crosstalk GP model that describes the possibility of crosstalk GP in asynchronous interfaces, at a logic level of abstraction. Section IV provides a design perspective of the proposed model, and Section V demonstrates the details of an experimental validation of the proposed technique using a Xilinx Virtex II-Pro field-programmable gate array (FPGA). Finally, Section VI concludes this paper.

II. Crosstalk Glitch Magnitude and its Impact on Asynchronous Interfaces

This section characterizes the effects of AQX on asynchronous interfaces. This characterization emphasizes the critical signals that are responsible for propagating glitches, due to AQX (and potentially lead to malfunction of the system), to the primary outputs (POs). Before going into the details of asynchronous circuit mechanisms that lead to identifying the critical signals, some general definitions and assumptions are stated.

Assumption: In this paper, only crosstalk glitches induced by an intrinsic signal transition within the same asynchronous domain, due to AQX, are analyzed. This assumption is justified, for instance, in GALS systems, for which the top metal layer is used to implement asynchronous interfacing signals. In this case, these signals are surrounded by signals that are typically least affected by AQX, i.e., VDD and GND. Hence, such an implementation reduces the possibility of any signal external to the asynchronous domain to cause AQX.

To simplify the analysis, it is also assumed that a good current-return path surrounds each wire, which alleviates inductive effects. Therefore, this paper is limited to the analysis of coupling capacitances only. This is in line with some crosstalk analysis in U-DSM technologies found in the recent literature [8].

Factors Affecting Crosstalk Glitch Magnitude: The magnitude of glitches in VLs due to a transition in an AL can be computed as follows [8]:

$$\frac{V_{\rm peak}}{V_{\rm DD}} \leq \frac{C_c}{C_c + C_{\rm gv}} \leq \frac{C_c/C_{\rm gv}}{(C_c/C_{\rm gv}) + 1} \tag{1}$$

where $V_{\rm peak}$ is the peak crosstalk glitch voltage, $V_{\rm DD}$ is the supply voltage, $C_{\rm gv}$ is the ground capacitances of the VLs, and C_c represents the coupling capacitance between two adjacent lines at steady state. When the aggressor switches very fast or the VL is floating, $V_{\rm peak}$ approaches the upper limit in (1). With the $C_c/C_{\rm gv}$ ratio getting as high as 4.5 in 32-nm technology [1], [8], crosstalk glitches for circuits designed in U-DSM technologies are more susceptible to reach the maximum theoretical value, i.e., $V_{\rm DD}$. In the International Technology Roadmap for Semiconductors (ITRS) 2007, it is noted that, by 2015, in 17-nm technologies, even the first metal layer will have glitches of 25% of the full switching voltage for 46-μm length of wire. Here, it is also to be noted that an increase in the coupling capacitance, as caused by reduced feature size, increases the settling time constant. This phenomenon contributes to longer crosstalk glitches, along with a higher crosstalk glitch magnitude in U-DSM technologies. In [1], examples are shown where such crosstalk glitches may reach a magnitude of $0.6V_{\rm DD}$, and the duration of glitches (larger than $V_{\rm DD}/2$) is reported to be more than three times the FO4 gate delay for the 90-nm STMicroelectronics CMOS technology [3]. Due to this high glitch magnitude and long glitch duration, it is predicted that crosstalk glitches due to AQX may propagate more easily into the logic and can lead to system malfunctioning. In the rest of this section, we provide detailed simulations that confirm the expected sensitivity of popular asynchronous interfaces.

Choice of Asynchronous Interfaces: Several asynchronous interfaces have been proposed to implement GALS systems. They often adopt one of the two following protocols: bundled data protocol [3], [4], [5] or delay-insensitive (DI) data-encoded protocol [2], [7], [6]. To understand the effects of crosstalk glitches in these asynchronous handshake protocols, an investigation is performed in the following sections for representative circuit implementations of the asynchronous interfacing protocols. These asynchronous interfaces are widely cited and used as benchmarks in several recent papers [4]–[6].

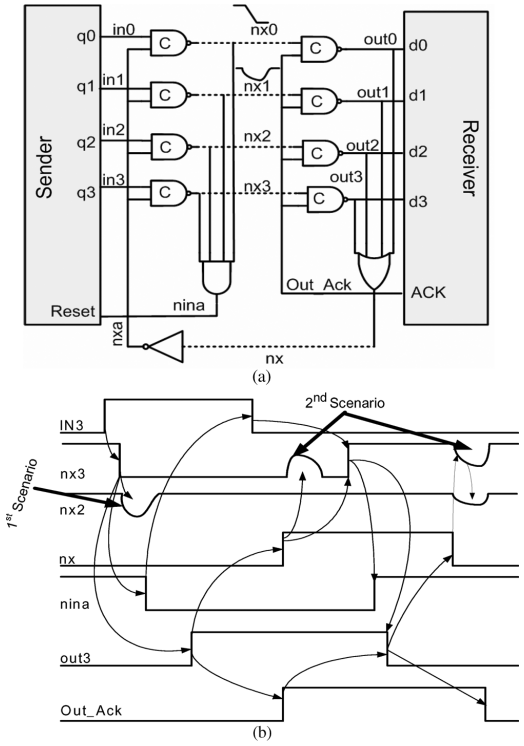


Fig. 1. (a) Hardware implementation of the data-encoded DI scheme. (b) Expected waveform of the design with glitch scenarios.

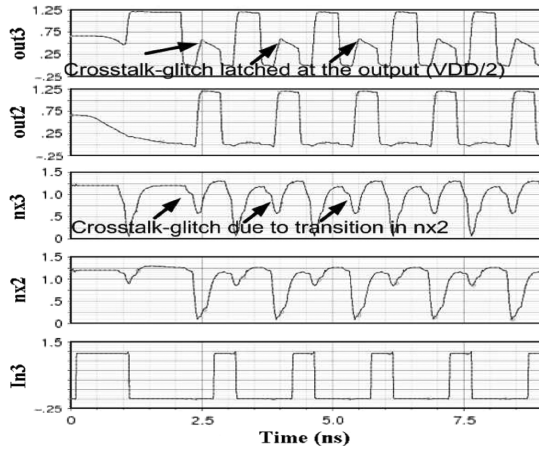


Fig. 2. Circuit-level simulation results for an optimized 1-of-4 data-encoded DI design scheme, as shown in Fig. 1(a), for an interconnect length of 1.5 mm.

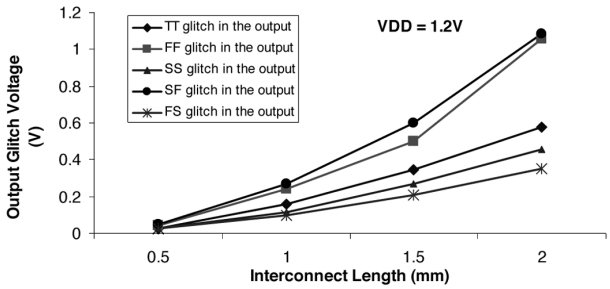


Fig. 3. Crosstalk glitch peak voltage in the 1-of-4 data-encoded DI design (having been optimized for latency in NXE simulations).

A. Crosstalk Glitch in a Conventional 1-of-4 DI Asynchronous Interface

This section analyzes the reasons for the vulnerability to AQX of a conventional 1-of-4 data-encoded DI asynchronous interface [2], [17], implemented as shown in Fig. 1(a). This analysis is based on the waveforms of Fig. 1(b) that illustrate possible AQX scenarios. Subsequently, these scenarios are examined in relation to the possible hardware implementation of Fig. 1(a) to identify the actual gates that can propagate crosstalk glitches.

To appreciate the effect of crosstalk glitches in such an interface, understanding the mechanics of the design is essential. The circuit of Fig. 1(a) shows the following scenarios of AQX occurrence during normal circuit operation. The nx0-to-nx3 group of signals is set to $V_{\rm DD}$ as an initial condition. This group of signals is represented as nx[0–3] in the rest of this paper. Similarly, in[0–3] represents the in0-to-in3 group of signals, and out[0–3] represents the out0-to-out3 group of signals. According to the 1-of-N DI data-encoded protocol, of which 1-of-4 is the special case shown in Fig. 1(a), only one of the input lines of in[0–3] can go high at a given time. When any of these input signals becomes high (to transfer some data), the corresponding line in the group nx[0–3] is pulled down to logic "0."

Two glitch generation scenarios are shown in Fig. 1(b). In the first scenario, a glitch is induced in nx2 due to a falling transition in nx3 in response to a rise in in3. The second glitch generation scenario arises when the transition in the nx signal, as shown at the bottom of Fig. 1(a), causes a glitch in a neighboring line (nx3 in this example). The second scenario appears twice in Fig. 1(b), one for each logic-level transition on the nx line.

Simulation Results: To validate the aforementioned analysis, two separate sets of circuit-level simulations were performed using 90-nm STMicroelectronics CMOS technology [1818]. Initially, a 1-of-4 data-encoded DI design is simulated with a model including the impedance of the lines, but without the coupling effect. This set of simulations is designated as no-crosstalk glitch effect (NXE) simulations. Although the coupling effect is not taken into consideration in the first simulations, for fair comparison with the simulations that include coupling capacitance, an additional ground capacitance of the same magnitude as the coupling capacitance is included in NXE simulations. Here, it is worth mentioning that, to avoid a pessimistic estimation (i.e., VL is not left as a floating line), we optimized the 1-of-4 data-encoded DI design for a particular worst-case delay for NXE simulations. This worst-case delay is measured from the rise of any of the in[0–3] signals to the latching of the data at the corresponding output. The value of this particular delay is close to 450 ps for a wire length of 1.5 mm. In simulation results that will be reported later in this paper, the 90-nm STMicroelectronics CMOS technology was used. In all these simulations, a top-level metal layer is used, and interconnects are modeled using a 5Π model, for which the relative delay errors associated with the use of lumped models are less than 3% [1919].

The second set of simulations includes coupling capacitances and is labeled "with crosstalk glitch effect" (WXE). This set of simulations is performed on the same optimized interface design (with no change in transistor sizing), whereas the line impedances include coupling effects. The results that are reported in Figs. 2 and 3 belong to WXE simulations only.

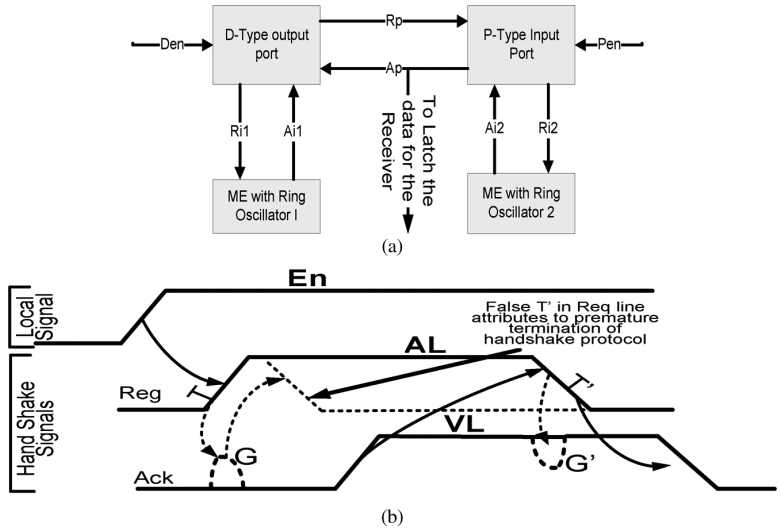


Fig. 4. (a) Conventional bundled data protocol: block diagram. (b) Critical handshaking signals. *Doutput port:* $R_{i1} = A_p + \mathrm{Den'Z1} + \mathrm{DenZ1'} + \mathrm{Den'}R_{i1}\mathrm{Z0'}$; $R_p = \mathrm{Den'}A_{i1}A'_p\mathrm{Z0'} + A_{i1}A'_p\mathrm{Z0'Z1'}$; $\mathrm{Z0} = \mathrm{Den'}A_p + A_{i1}\mathrm{Z0}$; $\mathrm{Z1} = \mathrm{Den}A_p + \mathrm{DenZ1} + A'_{i1}\mathrm{Z1}$. *Pinput port:* $R_{i2} = R_pR_{i2} + \mathrm{Pen'}R_p\mathrm{Ti}$; $A_p = A_{i2}$; $\mathrm{Ti} = \mathrm{Pen}A_{i2} + A'_{i2}\mathrm{Ti} + R'_{i2}\mathrm{Ti}$.

Doutput Port : R_{i1} = A_p + Den' Z1 + Den Z1' + Den' R_{i1} Z0'
R_p = Den' $A_{i1}A_p$' Z0' + A_{i1} A_p' Z0' Z1' Z0 = Den'A_p + A_{i1} Z0
Z1 = Den Ap + Den Z1 + A_{i1}'Z1
Pinput port: R_{i2} = R_pR_{i2} + Pen' R_p Ti A_p = A_{i2}
Ti = PenA_{i2} + A_{i2}'Ti+R_{i2}'Ti

Fig. 5. Conventional bundled data protocol: Boolean equations obtained from synthesizing the STG (using the 3-D tool).

Simulations on the circuit of Fig. 1(a) for the four process corner cases (SS, SF, FF, and FS)[1] and for a typical case (TT) were performed. These transistor-level simulations consider the case when in3 and in2, alternately, rise to logic "1" to measure the undesirable crosstalk glitch magnitude at output. The WXE transient circuit-level simulation results are shown in Fig. 2. These results show that such glitches may be latched as a valid output signal. For example, Fig. 2 shows that the out3 glitch magnitude is approximately $V_{\mathrm{DD}}/2$. This crosstalk GP indicates malfunction in the system, as the simulation shows that two of the out[0–3] signals are concurrently asserted, which is forbidden in the protocol.

The magnitude of the crosstalk glitch peak voltage on out3 is shown in Fig. 3 as a function of the interconnect length. This illustrates that the crosstalk glitch magnitude on signal out3 rises with the length of the wire. It is also observed that the crosstalk glitch magnitude may reach $V_{\mathrm{DD}}/2$ (0.6 V) for the interconnect length of 1.5 mm, which is high enough to turn on the subsequent gates and allow the glitches to propagate. Note that the variations in crosstalk glitch magnitude among different corner cases in Fig. 3 are due to the imbalanced sizing of PMOS and NMOS transistors, which is a requirement to optimize delays.

With this simulation, we have successfully demonstrated that crosstalk glitches can lead to malfunction in the 1-of-4 data-encoded DI asynchronous interfaces if no mitigation measures are taken. Further discussion on the implications of such a glitch generation and propagation scenario is provided at the end of Section II-B.

[1]SS means both PMOS and NMOS are slow; SF means slow PMOS, fast NMOS; FS means fast PMOS, slow NMOS; and FF means both PMOS and NMOS are Fast. TT represents both MOS having typical values.

B. Crosstalk Glitch in a Conventional Bundled Data Asynchronous Interface

This section examines the possibilities of AQX during the normal operation of a widely cited implementation of the bundled data protocol interface. Fig. 4(a) shows the generalized block-level structure for this asynchronous interface. Further details are provided in [3], where the state machines are described along with synthesis results produced with the 3-D synthesis tools [12], [13]. Fig. 4(b) shows the critical handshaking signals; dotted lines describe the signal corruption due to AQX glitches. The synthesized Boolean equations are shown in Fig. 5. From Fig. 4(a), it is obvious that the longest parallel-running handshake signals are R_p and A_p, and hence, arguably, they are the most glitch-prone signals. On closer inspection of this protocol, which is defined in [3], it is observed that the signal R_p undergoes a transition prior to A_p, as shown in Fig. 4(b). Therefore, there is a possibility of an AQX glitch occurrence in A_p. A large crosstalk glitch in A_P may lead to premature termination of the handshake protocol, which will be observed later in the simulation results shown in Fig. 6. It is also noticed that this glitch in A_p forces false latching of the data in A_p, which is an added adverse effect due to crosstalk glitches.

Simulation Results: To validate these analytical findings and to quantitatively characterize the crosstalk glitch effect, two sets of simulations were performed; results are shown in Fig. 6. Similar to the case of DI designs, these sets are designated as NXE and WXE simulations. As part of the first set of simulations with no crosstalk (i.e., NXE), the bundled data interface was optimized for a delay of approximately 3 ns when the interconnect length between the two communicating modules is 2 mm, and the typical delay model of the technology is used. The delay is measured from the assertion of the Den signal to the negation of A_{i1}, which indicates that the request R_p has been sent and acknowledged (A_p) (for a complete protocol specification,

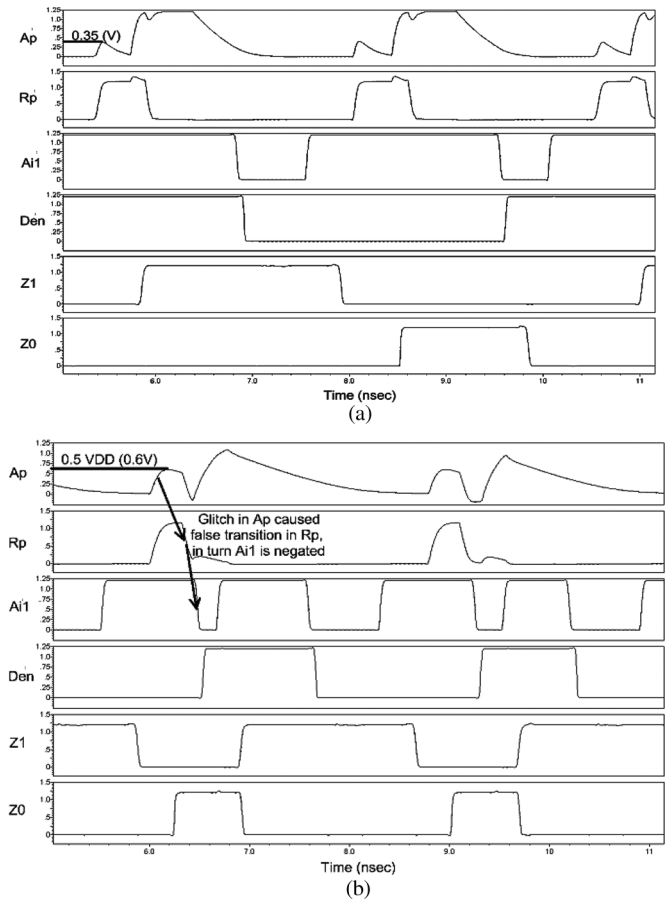


Fig. 6. Electrical simulation results for a circuit-level implementation (TT) of the bundled data protocol. (a) Interconnect length of 0.5 mm. (b) Interconnect length of 2 mm.

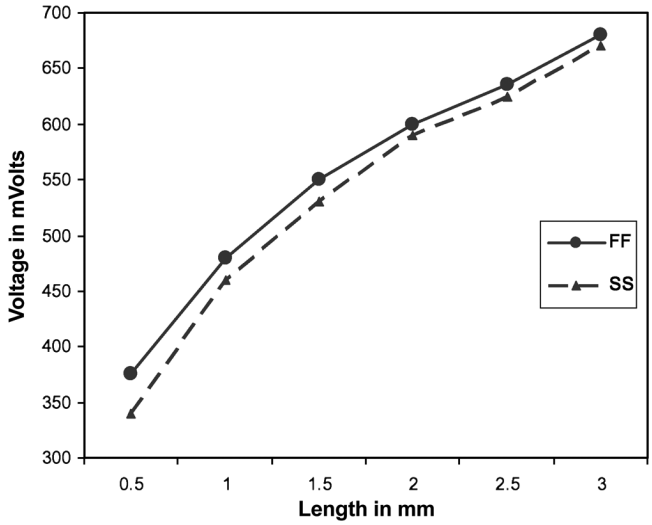


Fig. 7. Crosstalk glitch peak voltages in the bundled data protocol (having been optimized for latency in NXE simulations).

please see [3]). As usual with GALS systems, the sender clock is paused throughout the handshaking scheme.

To do a fair comparison between NXE and WXE, all interconnect parasitic capacitances are connected in both cases. However, in the NXE set of simulations, both nodes of the C_c capacitors are connected with an equal magnitude capacitor to the ground. By contrast, in the WXE simulation set, one C_c capacitor is connected between each pair of handshake signals. The WXE simulation set uses the same transistor sizing as the optimized NXE design. Fig. 6 shows the WXE transient circuit-level simulation results for this protocol. The simulated interconnect length between the two modules was varied from 0.5 and 3 mm. Fig. 6(a) shows the simulation results obtained with an interconnect length of 0.5 mm. It can be seen that A_p glitches with the rise of R_p, but this glitch magnitude is not large enough to cause a glitch to propagate in the circuit. However, in Fig. 6(b), when the interconnect length is increased to 2 mm, the magnitude of the glitch is large enough to falsely negate the R_p signal, and consequently, it prematurely terminates the handshake scheme (indicated by negation of the A_{i1} signal). Arrows in Fig. 6(b) indicate one such occurrence. Fig. 7 shows the rise in crosstalk glitch magnitude with the increase in interconnect length. This simulation shows that, conventionally, AQX glitches propagate and cause system malfunctioning. This identification of AQX effects was only possible through detailed electrical simulations, and these effects are otherwise hidden from logic designers.

Our foregoing simulations prove that conventional asynchronous interfaces are not immune to intrinsic crosstalk glitches due to AQX. They show that, if glitches cross the logic threshold and are able to switch logic modules, then conventional asynchronous protocols are vulnerable, as they are not designed to address the challenges associated with AQX glitches. Our concern is that, if a designer is unaware of what signals may contribute in propagating glitches in a harmful way, then the design process becomes either less reliable or forces adoption of unnecessary brute force solutions, even when possible glitches could have no significant consequence on circuit operation. Therefore, there is a requirement to develop a methodology that can predict the vulnerable nodes/conditions/primary inputs (PIs) that may be managed at the logic level. Alternately, if the phenomenon cannot be treated at the logic level, then this information can be relayed to the physical-level design, where some glitch quenching mechanism can be utilized. Thus, toward this goal, we present a novel modeling technique to predict possible glitch adverse effects (Section III).

III. GP Modeling for Asynchronous Handshake Schemes

The proof-of-concept simulations of the preceding section illustrate the potential vulnerability to AQX glitches of asynchronous interfaces, particularly in future DSM technologies. In a complex integrated circuit, the asynchronous interfaces can be composed of hundreds of wires. Typically, glitches appear when wires have long parallel runs. It has recently been shown that, as technology scales, shorter and shorter wires become sensitive to AQX glitches (an example is discussed in Section II in relation to ITRS-2007). A possibility is to do systematic buffer insertion. The drawback of this approach is that a large number of signals could require many buffers in advanced DSM, which would require a lot of power and area. Therefore, there is a

growing need to identify AQX-glitch-affected signals in asynchronous interfaces before their physical-level implementation. This section elaborates on a novel modeling approach that helps a designer identify the AQX-sensitive signals that can propagate errors to the POs of asynchronous interfaces under certain operating conditions. These operating conditions include a set of PIs and/or initial conditions of the internal nodes of the interface. The benefit of such modeling is that this information becomes available at the logic abstraction level, and designers do not have to resort to detailed circuit-level simulations. Moreover, if the detected conditions cannot be addressed at the protocol design stage, then this information is relayed to the physical design stage where physical design measures can be taken to avoid these AQX glitches from propagating to become errors.

Inspired by the D-algebra, as proposed by the test community [1], [2], a new concept is introduced in this section for modeling the effect of glitches in asynchronous handshake schemes. As part of the proposed framework, it is essential to first define some notations.

As discussed in Section II, in the context of asynchronous handshake schemes, the AL and VL of interest are handshake signals that are communication lines between two mutually asynchronous communicating modules, which may be classified as sender and receiver modules. Hence, such lines (AL and VL) are either input or output to these modules. The analysis performed in this section only considers the modules where the VL is an inbound signal. This is because of the possibility that GP is only associated with such modules (this is further explained in Definition 5). Thus, the input and output notation is used with respect to the modules to which the VL is an input. Notice that, since asynchronous circuits, in general, consist of LEs or Muller "C" elements, the AL and VL can always be studied as inputs or outputs to various LEs. Therefore, the "_in" in $\mathrm{AL\_in}$ and $\mathrm{VL\_in}$, and the "_out" in $\mathrm{AL\_out}$ and $\mathrm{VL\_out}$, respectively, denote whether the AL (or VL) signal is an inbound signal or an outbound signal to the relevant asynchronous module.

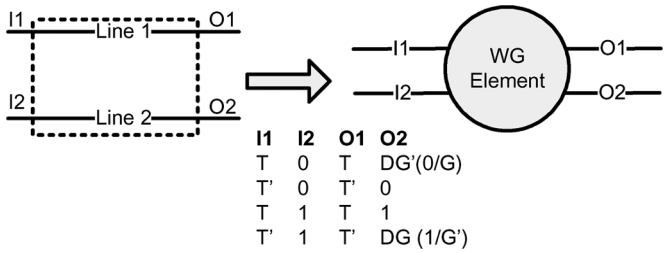
Definition 1: T (T') is a signal transition from logic level 0 to 1 (1 to 0).

Lemma: T (T') is deterministic (nondeterministic) if it is associated with $\mathrm{AL\_out}$ $(\mathrm{AL\_in})$. This follows from the fact that, when the AL is output to the asynchronous modules, then the conditions for asserting the signal are known to the module under study.

Definition 2: G (G') represents glitches in the VL due to the transition T (T') in the AL. A glitch on a particular VL, e.g., A, due to T (T') in the AL is represented as $\mathrm{G_A}$ (G'_A). The VL returns to its stable state after a bounded delay, i.e., Δt_G.

Note that, in the context of AQX, the VL may glitch, i.e., G (G'), only if the VL and AL are at the same logic level before the transition T (T') in the AL. This is further explained in Definition 4.

Definition 3: DG (DG') represents composite logic values of the form v/vg, where v and vg are the values of the same signal in the glitch-free and nonglitch-free circuits, respectively. The composite logic values that represent errors, i.e., $1/\mathrm{G}'$ and 0/G, are denoted by the symbols DG and DG', respectively. This notation has been chosen by analogy with D-algebra, where 1/0 and 0/1 are represented as D and D', respectively.



I1	I2	O1	O2
T	0	T	DG'(0/G)
T'	0	T'	0
T	1	T	1
T'	1	T'	DG (1/G')

Fig. 8. Logic-level representation of AQX, which is called the WGE.

Definition 4: A WGE is a pseudologic-level representation of the AQX phenomenon, as shown in Fig. 8. The WGE translates a physical-level AQX-affected signal into a representation that can be understood at the logic abstraction level. In Fig. 8, the values on I1 and I2 are the values imposed on the AL and VL, respectively. Provided there is no AQX phenomenon, then O1 and O2 keep the logic values that are imposed on I1 and I2, respectively. On the other hand, if the wires are subject to AQX, then O1 and O2 represent the resulting propagated values on the wire due to the AQX effect. The logic table in Fig. 8 shows that, whenever I1 and I2 are at the same initial logic levels, T (T') in I1 causes DG' (DG) in O2, whereas O1 keeps the same value T (T'). It is imperative for our modeling approach to define the criteria for WGE insertion.

Criteria to Insert a WGE: Any pair of intrinsic handshake signals that may utilize global or intermediate wires will require WGE insertion if one of the two signals, which can potentially run parallel, does a transition T (T'), whereas the other signal remains stationary at logic level 0 (1).

Since glitch magnitude is proportional to wire length with direct capacitive coupling between respective connected gates or buffers, consideration should also be given to the relevant length of wires. The simulation results for the technology that we used in the preceding sections, i.e., the 90-nm STMicroelectronics CMOS technology, allow us to safely conclude that, in an asynchronous interface, only the longest wires require WGE insertion. The envisioned design flow is as follows: Signal sensitivities are tested at the logical level before physical design. Before detailed placement and routing, but after a floor plan has been developed, a list of potentially adjacent long wires is generated. Analogous to buffer insertion and sizing during logic synthesis based on predicted load, temporarily assume that a pair of wires could become aggressor and victim and determine those where a glitch would propagate if generated. Such potentially sensitive signals would require some mitigation technique such as repeater insertion, route far away (tag, routing constraint), logic mitigation, or some phase-modulation-based technique [27]. A protocol designer may try to introduce logic techniques that help in mitigating/eliminating such glitches and may also relay the information related to the potential signals to the physical designer. The salient feature of this approach is the reduction in the required area and power, as compared with adopting brute force systematic repeater insertion.

It is of interest that, as we move deeper into technology scaling, WGE insertions will be required even for local length interconnects, as several factors become increasingly significant. Some expected relevant factors are given in the list that follows.

1 Knowledge of the maximum length of wire where a FO4 gate can keep the glitches within $V_{\rm DD}/4$. This information is usually available beforehand for well-characterized technologies. This metric proposed in ITRS-2007 [11], provides wire lengths for which a glitch of $V_{\rm DD}/4$ is observed for different metal layers. This is a means to characterize possible harmful crosstalk effects in future technologies.
2 Identify the potentially weak drivers for nontransitioning wires.
3 Special consideration should be given to feedback signals because they are usually driven by large drivers. Hence, glitches inflicted due to feedback signals are strong candidates for WGE insertion.

Leveraging the WGE to Identify the AL and VL: To model AQX glitches at the logic abstraction level, it is necessary to identify the AL and VL. The following steps are proposed for identifying the AL and VL in an asynchronous interface and thus utilize the WGE.

1) Identify the signals where WGE insertion is required.
2) Leverage the knowledge of the asynchronous protocol to find the sequence of transitions in the handshaking schemes.
3) For each transition, check the WGE truth table shown in Fig. 8 for a possible glitch occurrence; hence, identify the AL and VL.
4) If the asynchronous handshake scheme involves the RTZ signaling scheme, then such a glitch may occur twice (positive and negative transitions); therefore, reapply this method for the second transition.

Definition 5: A GP set is a set of conditions that allow a glitch, in different logic gates and Muller "C" elements, to propagate. For brevity, this paper focuses on four specific gate types: inverter, two-input AND and OR gates, and Muller C element. It is possible to express all circuits of interest using these gates. In addition, the analysis method used to derive the GP sets can easily be generalized to cover all other gates of interest. An example of the GP set for the AND gate that produces DG at the output is {(1, DG), (DG, 1), (DG, DG)}. Fig. 10 summarizes these conditions for the four considered LEs. Following this, we explain the GP sets for the Muller "C" element. It can be seen that the GP sets for the "C" element have a timing notation as well (t^- and t^+). Let us analyze the second GP set of the "C" element as an example. This set is written as follows: $\{(0, \mathrm{DG}), (\mathrm{DG}, 0), (\mathrm{DG}, \mathrm{DG})\} \rightarrow t^- = 1$ and $t^+ = \mathrm{DG}$. This can be read as follows: If the output of the "C" element is logic "1" before the occurrence of a composite logic value of DG in any or both inputs of the "C" element, then, due to the glitch, the output of the "C" element will have a composite logic value of DG at the output after the gate propagation delay at time t^+.

Required Steps for Modeling Crosstalk GP at the Logic Abstraction Level: Our modeling approach utilizes the aforementioned definitions to understand whether GP is possible in the given asynchronous interface. A designer begins the analysis by utilizing the proposed WGE to identify the AL and VL and their directions (input or output to the module). Next, the designer has to identify the LEs to which the VL is an input $(\mathrm{VL\_in})$. Further explanation of GP sets leads us to make a corollary with different cases (this is explained in the subsequent paragraphs). Using this explanation of GP sets, a designer can predict whether the glitch may propagate in the circuit under study. Hence, this modeling technique enables the designers to identify crosstalk GP at the logic abstraction level. Our modeling approach is broadly applicable, as it analyzes the constituents of asynchronous circuits (i.e., the LEs), and therefore, it can be applied to many asynchronous handshake interfaces. These steps are summarized in the flowchart shown in Fig. 9. In this figure, suppose that S is the total number of WGEs required for the interface. This number is obtained by applying the WGE insertion criteria to the given interface. The algorithm shown in Fig. 9 illustrates the conditions under which the AQX glitches propagate to POs. The final outputs of this flowchart are the PIs and/or initial conditions, along with the identification of the WGE node where the glitch is entered to the asynchronous interface.

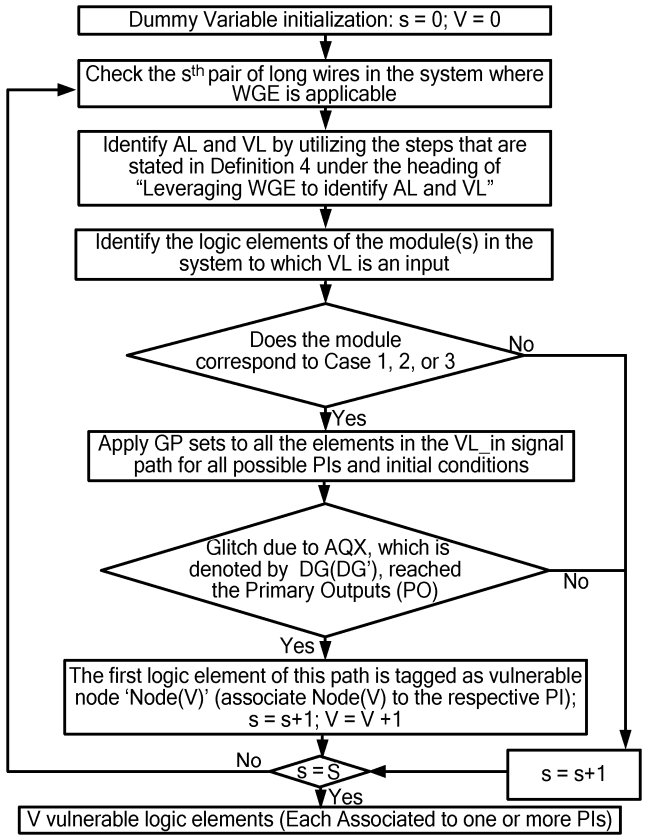


Fig. 9. Flowchart of the proposed modeling approach.

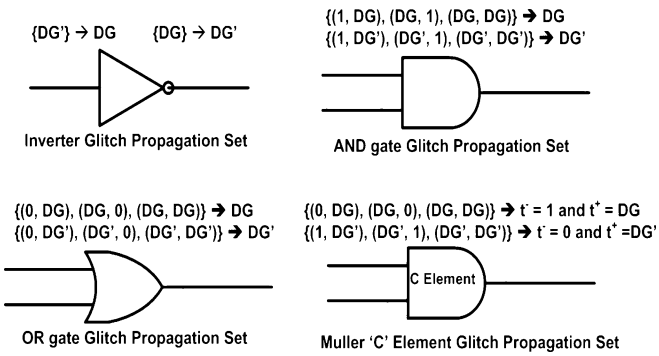


Fig. 10. GP sets for the inverter, AND and OR gates, and the Muller "C" element.

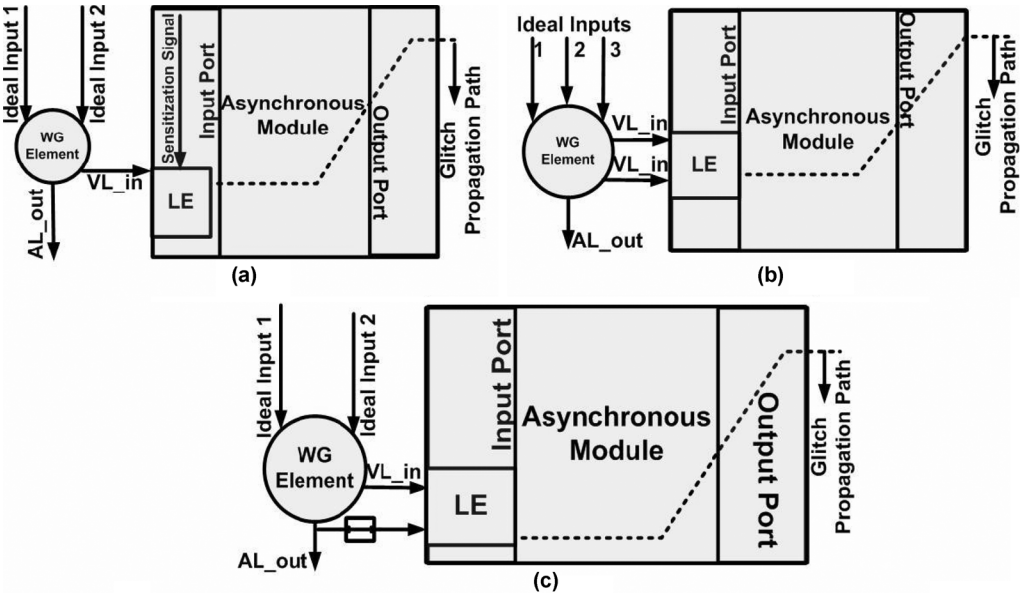


Fig. 11. Pictorial illustration of possible inputs to the glitch-propagating LE for the case $\mathrm{AL\_out} \wedge \mathrm{VL\_in}$.

Generalization of GP Sets: As stated in the paragraphs where we explained the required steps for modeling GP, this section derives a corollary and splits it into different cases to predict the possibility of GP. Note that each element of these GP sets always contains $\mathrm{VL\_in}$ (in the form of DG or DG'), as shown in Fig. 10. In the example of the GP set of the AND gate discussed in the previous section, each element of the set has at least one input as DG, which corresponds to $\mathrm{VL\_in}$, whereas the other input is either DG or the noncontrolling value of the gate, which is "1" for the AND gate. Therefore, the GP analysis is considered complete only if it contains all the modules to which $\mathrm{VL\_in}$ is one of the inputs. The VL may also be an output to the module, but it cannot contribute to GP in that module. However, it may act as an input to the opposite module; thus, it must be analyzed when the opposite module is investigated. DG or DG' on the right-hand side of GP sets in Fig. 10 shows that the output of the gate represents propagation of the glitch.

G (G') in the VL is a consequence of T (T') in the AL, which may be $\mathrm{AL\_in}$ or $\mathrm{AL\_out}$ to the module. Therefore, it can be concluded that "for all GP sets, there exists at least one AL, such that it is either $\mathrm{AL\_in}$ or $\mathrm{AL\_out}$, causing G or G' in $\mathrm{VL\_in}$." The statement in quotes can mathematically be written as the following corollary.

Corollary: $\forall \mathrm{GP} \exists \mathrm{AL} | ((\mathrm{AL\_in} \wedge \mathrm{VL\_in}) \vee (\mathrm{Al\_out} \wedge \mathrm{VL\_in}))$.

Explanation: Before elaborating on the preceding corollary, some background information is provided. Each asynchronous module may consist of many LEs. $\mathrm{AL\_out}$ and $\mathrm{VL\_in}$ can be associated with one or more logical elements inside the module. Note that the only necessary input signal for the glitch to propagate in an LE is $\mathrm{VL\_in}$, which is represented as DG or DG' in the GP sets. $\mathrm{AL\_in}$ and $\mathrm{AL\_out}$ signals can cause $\mathrm{VL\_in}$ to G (G') but may not necessarily be part of the same LE that propagates the glitch. The aforementioned corollary shows that GP is possible in any of the three cases.

1) $\mathrm{AL\_out} \wedge \mathrm{VL\_in}$, i.e., the module has $\mathrm{VL\_in}$ that glitches, i.e., G or G', due to T or T' in $\mathrm{AL\_out}$. Based on the possibilities of inputs to the LEs through which the glitch may propagate, this case can further be subdivided. It is evident from Definition 5 that one of the two inputs of the glitch-propagating LE should be $\mathrm{VL\_in}$. The other input should have a sensitizing value to fulfill the condition of a GP set. Physically, such a signal can be any of the following: 1) an independent sensitizing signal; 2) another handshaking signal, which is affected by crosstalk; or 3) a feedback signal, which is derived from the AL. These different conditions are elaborated in Fig. 11.
2) $\mathrm{AL\_in} \wedge \mathrm{VL\_in}$, i.e., the module has $\mathrm{VL\_in}$ that glitches (G or G') due to transition (T or T') in $\mathrm{AL\_in}$. Again, based on the possibilities of inputs to the LE through which glitches may propagate, this case can also be further subdivided into the following two cases (following the same guidelines as adopted for case 1): 1) GP set inputs are $\mathrm{VL\_in}$ and the relevant sensitizing signals; and 2) the GP set consists of $\mathrm{VL\_in}$ and $\mathrm{AL\_in}$ (or two $\mathrm{VL\_in}$ signals). Fig. 12 pictorially elaborates these cases.
3) $(\mathrm{AL\_in} \wedge \mathrm{VL\_in}) \wedge (\mathrm{AL\_out} \wedge \mathrm{VL\_in})$. There are several possibilities associated with this case. To simplify the analysis, it is treated as two different nonconcurrent events. This case is divided into two subcases as follows: 1) if $\mathrm{VL\_in}$ is the same physical line for both events (this is a possibility when the same $\mathrm{VL\_in}$ is subject to the glitch due to two physically different ALs) and 2) if two physically different $\mathrm{VL\_in}$'s are involved (i.e., the asynchronous module receives two or more VLs at two different time instances due to physically different ALs). The GP sets for these subcases can be obtained by utilizing the description of the preceding two cases (i.e., case 1 and case 2).

The following section utilizes this modeling approach to obtain the conditions under which glitches may propagate in the asynchronous interfaces analyzed in Section II.

IV. Application of the Proposed Model

The proposed modeling scheme can be applied to most asynchronous interfacing circuits to identify the possibility of GP

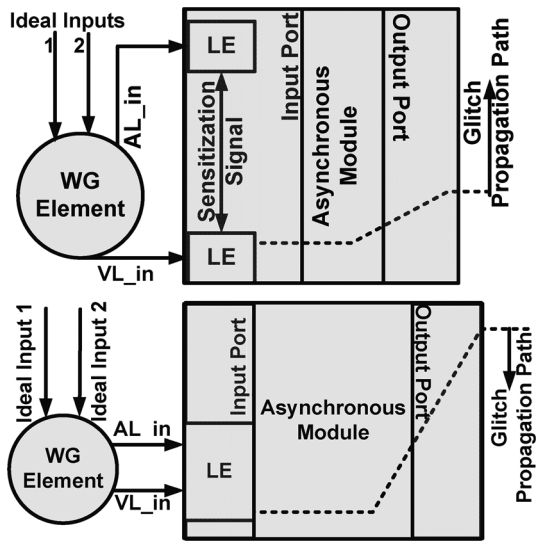


Fig. 12. Pictorial illustration of the possible inputs to the glitch-propagating LEs for the case $\mathrm{AL\_in}\hat{}\mathrm{VL\_in}$.

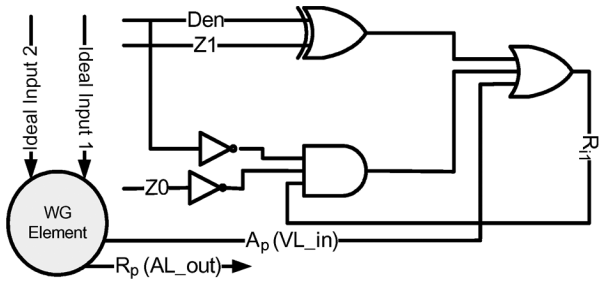


Fig. 13. Hardware implementation for R_{i1} signal generation of the Doutput port, with conceptual realization of the WGE.

at the logic abstraction level. In this section, the usefulness of the proposed technique is illustrated by applying it on the same interface circuits that were described in Section II. It is imperative, for establishing the validity of the proposed model, that this analysis, using our modeling approach, gives us the same results, at least in a conservative way, as achieved using circuit-level simulations. This is demonstrated in the following analysis.

Application to the Bundled Data Protocol: Fig. 5 provides the Boolean equations describing the two ports of an interface implemented according to the bundled data protocol. For convenience, these Boolean equations for the two outputs of the Doutput port are reproduced here: $R_{i1} = A_p + \mathrm{Den'Z1} + \mathrm{DenZ1'} + \mathrm{Den'}R_{i1}\mathrm{Z0'}$ and $R_p = \mathrm{Den'}A_{i1}A'_p\mathrm{Z0'} + A_{i1}A'_p\mathrm{Z0'Z1'}$. It is shown in Section II that A_p is the only $\mathrm{VL\_in}$ for the Doutput port, which is the sender-module output. In this section, the proposed modeling technique is applied to this interface. Considering the first WGE insertion criterion, the WGE is applicable on one set of parallel wires, which consists of signals R_p and A_p. Now, the application of the proposed modeling method (Fig. 9) identifies that R_p is the AL signal and A_p is the VL signal (as explained in the preceding section). A_p is an input (i.e., $\mathrm{VL\_in}$) to the logic gates that generate both PO signals. Therefore, a separate analysis is conducted on both (R_{i1} and R_p) the PO signals. This analysis, as the algorithm suggests, is used to predict the circuit behavior due to the crosstalk glitches (with A_p as $\mathrm{VL\_in}$) at the logic abstraction level for all the PIs and initial conditions. This is the same circuit that was described in Section II-B through circuit-level simulations.

R_{i1} Signal: The hardware implementation of the R_{i1}-generating circuit, along with the WGE for the glitch-affected signal A_p, is shown in Fig. 13. From the truth table of the WGE, the condition for A_p $(\mathrm{VL\_in})$ to glitch due to the transition in R_p $(\mathrm{AL\_out})$ can be obtained. $\mathrm{AL\_out}$ and $\mathrm{VL\_in}$ make this circuit an example of case 1 of the corollary. The GP set of the OR gate suggests that the glitch may propagate through the OR gate of Fig. 13 when the derived inputs (i.e., other than the $\mathrm{VL\_in}$ signal) act as sensitizing signals, whereas $\mathrm{VL\_in}$ is subject to $\mathrm{DG'}$. Hence, this example can further be subclassified as case 1-A, where the GP set inputs are $\mathrm{VL\_in}$ and a sensitizing signal. For the next step in the algorithm (application of the GP sets for all possible PIs), the GP sets for the OR gate (the LE to which the VL is an input) is utilized. It is shown in Fig. 10 that the set $\{(0,\mathrm{DG'}),(\mathrm{DG'},0),(\mathrm{DG'},\mathrm{DG'})\}$ will cause a glitch (composite logic $\mathrm{DG'}$) to propagate to the output of the OR gate. As the second part of this step, all the PI combinations for which the two derived inputs (i.e., the inputs other than $\mathrm{VL\_in}$) of the OR gate can become "00" (the noncontrolling value for the OR gate) are investigated. These PI vectors are found to be $\{\mathrm{Den},\mathrm{Z1},\mathrm{Z0}\} = \{001,110,111\}$. These three vectors are obtained from a classic digital design technique. Hence, we showed that our modeling approach provides digital designers the conditions that can lead to crosstalk GP.

Further inspection of the protocol under study [3] shows that, according to the Boolean equations of the state machine (Fig. 5), only vector "110" is allowed to occur in this protocol. The other elements are filtered out by the protocol itself. Hence, the proposed modeling technique allows identifying at least one possible scenario in which a crosstalk glitch may propagate to R_{i1}. An experimental validation is performed for this case in Section V.

R_p Signal: Fig. 14 shows the hardware implementation of the R_p signal (the Boolean equation is shown in Fig. 5) along with the WGE. A similar analysis (as exercised for the R_{i1} signal) led us to identify A_p as $\mathrm{VL\_in}$ and R_p as $\mathrm{AL\_out}$. It is seen from this figure that A_p $(\mathrm{VL\_in})$ is an input to both the AND gates (after passing through an inverter). As the next step, it is observed from Fig. 14 that this signal-generating circuit is an example of case 1-A of the corollary (where one of the inputs to the LE is $\mathrm{VL\_in}$ and the other is a sensitizing signal). The following step requires application of GP sets for all the PIs. The GP sets for the AND gates (Fig. 10) showed us that, when $\mathrm{Den},A_{i1},\mathrm{Z0},\mathrm{Z1} = \{1100,0100\}$, then they act as sensitizing signals for propagating the composite value of $\mathrm{DG'}$. Here, it is worth mentioning that the glitch occurrence that is shown in Fig. 6(b) is for the input vector (0100), which is identical to one of the vectors predicted by the proposed modeling. Hence, it is demonstrated through the aforementioned analysis that our proposed model has correctly predicted the outcome of the AQX-affected bundled data protocol at the logic abstraction level. An experimental validation is provided in the next section. To see the practicality of this result, it is observed that, in

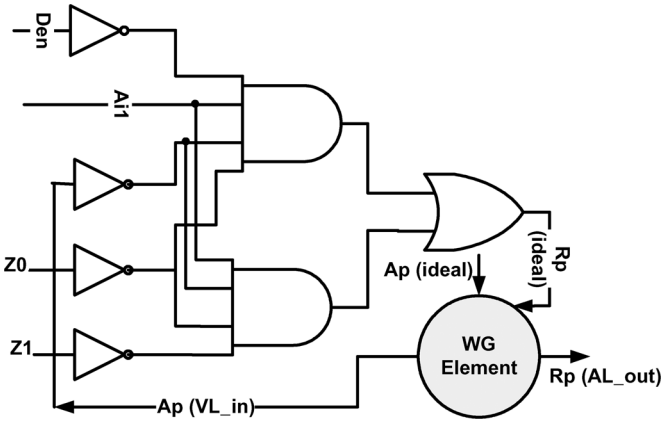


Fig. 14. Implementation for R_p signal generation with the WGE.

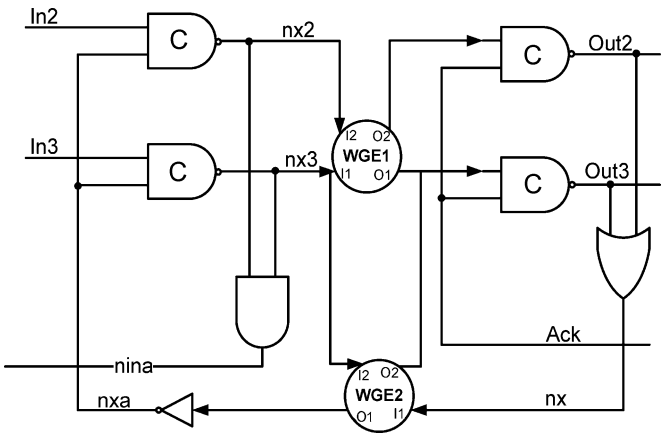


Fig. 15. Hardware implementation of the DI data-encoded interface with conceptual WGEs.

the protocol provided in [3], both conditions (i.e., both PI vectors) can occur.

Application to the DI Data-Encoded Protocol: To provide an example with the DI data-encoded protocol, we use a simplified version of the interface shown in Fig. 1(a). This simplified interface is shown in Fig. 15. As can be seen, there are two WGEs (the relevant WGEs were obtained with the proposed insertion criteria) in this figure, one for each scenario depicted in Fig. 1(b). As a first step toward implementing our modeling approach, we need to know the AL and VL of the design. Here, the knowledge of the protocol plays a major role in understanding the nature of the AL and VL. Through the steps illustrated for "Leveraging the WGE to identify the AL and VL," which is defined in the preceding section under Definition 4, it is observed that both nx2 and nx3 may act as both the AL and VL at different stages of the protocol. Furthermore, the nx line can also act as an AL, which may inflict a glitch on the VL (nx3). In this protocol, VLs, under all the conditions, are inputs to the C elements at the receiving end. Hence, this interface can be treated as an example of case 3, where WGE1 depicts the case when both the AL and VL are inputs to the receiver module. Whereas, WGE2 shows that the AL is an output to the receiver module and the VL is an input to the receiver module. Therefore, GP involving WGE1 can be classified as an example of case 2-A, and GP at WGE2 can be categorized as case 1-A. Hence, both cases may

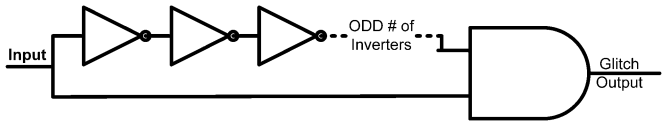


Fig. 16. Circuit used for simulating WGE behavior at the logic level.

contribute to the crosstalk GP and should accordingly be handled.

Following the steps of the algorithm for each WGE, GP sets are separately applied to the glitch output of both WGEs.

WGE1: According to the protocol, when idle, out[0–3] are at logic "0," and nx[0–3] are at logic "1." Therefore, the GP set for the inverted Muller "C" element that applies to this case is $\{(0, \mathrm{DG}), (\mathrm{DG}, 0), (\mathrm{DG}, \mathrm{DG})\} \rightarrow t^- = 0$ and $t^+ = \mathrm{DG}'$. Hence, if $\mathrm{Ack} = 0$, this glitch may propagate.

WGE2: A similar analysis is applied to WGE2. According to the protocol, it is known that nx only rises when one of the out[0–3] signal rises. The rise of out[0–3] is only possible when either nx2 or nx3 is at logic "0." Hence, we can say that GP is possible when the following two combinations of initial conditions, according to the GP sets, are observed: ($\mathrm{out3} = 0$ and $\mathrm{Ack} = 1$) and ($\mathrm{out3} = 1$ and $\mathrm{Ack} = 0$).

The aforementioned examples show that our modeling approach is flexible and can predict the conditions of GP on different asynchronous interface circuits. As our modeling approach is dependent on the GP sets that identify GP through LEs that are the constituent elements of the asynchronous interfaces, therefore it is applicable to many designs.

V. Experimental Validation

This section discusses the results of experiments that were performed to validate the proposed modeling method on the bundled data asynchronous interface. Its R_{i1}- and R_p-generating circuits shown in Figs. 13 and 14 are considered. We first focus on back-annotated simulation results, and we then confirm that the actual hardware exhibits the same behavior. These results were produced for a Xilinx Virtex II-Pro XC2VP30-7FF896. Glitches coming out of the conceptual WGE in Figs. 13 and 14 are introduced using the circuit shown in Fig. 16 and synthesized with the Xilinx Synthesis Tool using the "keep" attribute.

R_{i1} Signal: Fig. 17 shows back-annotated simulation results for the R_{i1} circuit (Fig. 13). This simulation is performed for two conditions, namely, with and without glitches, for the same input vector, to validate the proposed model. The left circle in Fig. 17 shows that, if there is no glitch, then R_{i1} is not asserted for the input vector $\{\mathrm{Den}, \mathrm{Z1}, \mathrm{Z0}\} =$ "110". The right circle shows the converse case, that is, when the circuit is subject to a glitch via the $\mathrm{glitch\_output}(A_p)$ signal for the same input vector, then this glitch is propagated to the PO R_{i1}. The count value shows the number of transitions in R_{i1}. This result is in agreement with the analysis provided in the preceding section.

The circuit was also implemented in hardware, using an FPGA board provided by the Xilinx University Program. For the input vector {"110"}, count_val (measuring the number of assertions in R_{i1}) is observed for both cases, with and without

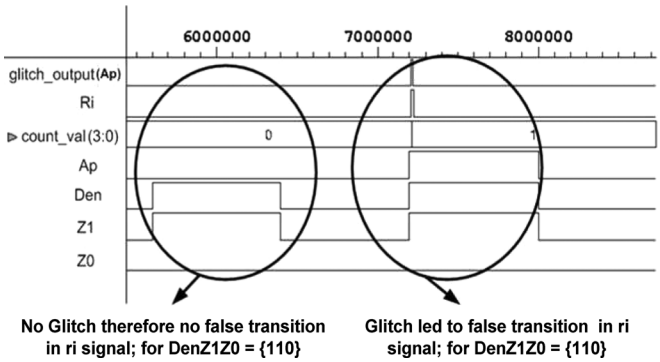


Fig. 17. Back-annotated simulation results of an FPGA (Virtex II-Pro) implementation of the design shown in Fig. 13.

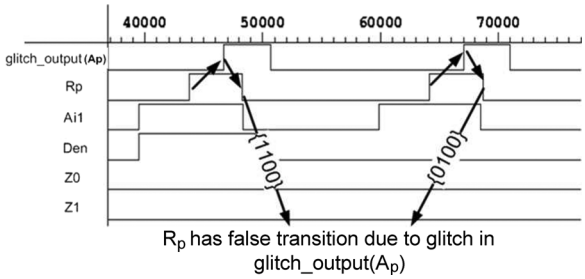


Fig. 18. Back-annotated simulation results of an FPGA (Virtex II-Pro) implementation of the design shown in Fig. 14.

glitch. The results were found to be in compliance with the simulation results.

R_p *Signal:* Fig. 18 shows the back-annotated simulation results for the FPGA implementation of the circuit shown in Fig. 14. The right-hand side of Fig. 18 shows that, when $\{\mathrm{Den}, \mathrm{A}_{i1}, \mathrm{Z0}, \mathrm{Z1}\} = \{(0100)\}$, a glitch in the A_p signal $(\mathrm{VL\_in})$ (labeled as $\mathrm{glitch\_output}(A_p)$) is propagated to R_p, thus forcing it to do a false transition from logic 1 to 0. This figure shows that the results obtained through the transistor-level simulation, as shown in Fig. 6(b), are reproduced here at the logic abstraction level, using the proposed modeling approach. Similarly, the transitions on the left side of Fig. 18 illustrate that the R_p signal does a false transition for the case $\{\mathrm{Den}, A_{i1}, \mathrm{Z0}, \mathrm{Z1}\} = \{(1100)\}$ as well.

VI. Conclusion

In this paper, it has been demonstrated that asynchronous handshake schemes can malfunction under normal operating conditions due to the influence of crosstalk (AQX). Representative circuits of two widely used classes of asynchronous handshake circuits were simulated using the 90-nm STMicroelectronics CMOS technologies. It was shown that false events due to AQX can propagate to POs of asynchronous interfaces. To understand the AQX effects at a higher abstraction level, a GP modeling technique was proposed. This model identifies the possibility of intrinsic crosstalk GP in asynchronous handshake interfaces at the logic abstraction level and helps a protocol designer avoiding the GP conditions or enables relaying the information to the physical designer for special consideration. This modeling technique is applied to asynchronous protocols under study to predict GP. Our approach has successfully predicted the conditions under which this interface may propagate glitches. The results obtained through the proposed modeling technique for the bundled data protocol were also experimentally validated with an FPGA implementation on a Xilinx Virtex II-Pro XC2VP30-7FF896. It has been demonstrated that the same glitch behavior, as observed using transistor-level simulations, is also obtained at the logic abstraction level through our modeling technique. To the authors' knowledge, this is a first work on a framework that allows representing the possible behavior of a logic structure, for asynchronous handshake schemes, in the presence of crosstalk glitches, without the help of circuit-level simulations. This framework can be used as a step toward formalizing the asynchronous circuit behavior under crosstalk glitches. Furthermore, it can be used to validate glitch mitigation techniques that will be explored in future work.

Acknowledgment

The authors would like to thank the following programs and organizations: the Canada Research Chair Program for financial support; CMC Microsystems and the Xilinx University Program for providing access to technologies, design tools, and hardware design technologies; and, finally, the anonymous reviewers for their constructive criticisms, which helped improve this paper.

References

[1] S. R. Hasan and Y. Savaria, "Crosstalk effects in event-driven self-timed circuits designed with 90 nm CMOS technology," in *Proc. IEEE ISCAS*, May 2007, pp. 629–632.

[2] W. J. Bainbridge and S. B. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," in *Proc. 7th Int. Symp. ASYNC*, Mar. 2001, pp. 118–126.

[3] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally-asynchronous locally-synchronous systems," in *Proc. 6th Int. Symp. Adv. Res. ASYNC*, Apr. 2000, pp. 52–59.

[4] R. Mullins and S. Moore, "Demystifying data driven and pausible clocking scheme," in *Proc. 13th IEEE Int. Symp. ASYNC*, Mar. 2007, pp. 175–185.

[5] D. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, , Stanford Univ., Stanford, CA, Oct. 1984.

[6] M. De Clercq and R. Negulescu, "1.1-GDI/s transmission between pausible clock domains," in *Proc. IEEE ISCAS*, 2002, pp. II-768–II-771.

[7] W. J. Bainbridge, W. B. Toms, D. A. Edwards, and S. B. Furber, "Delay-insensitive, point-to-point interconnect using m-of- n codes," in *Proc. 9th Int. Symp. ASYNC*, May 2003, pp. 132–140.

[8] B. P. Wong, A. Mittal, Y. Cao, and G. Starr, *Nano-CMOS Circuit and Physical Design*. New York: Wiley-Interscience, 2005.

[9] E. G. Jung, B. S. Choi, Y. G. Won, and D. I. Lee, "Handshake protocol using return-to-zero data encoding for high performance asynchronous bus," *Proc. Inst. Elect. Eng.—Comput. Digit. Tech.*, vol. 150, no. 4, pp. 245–251, Jul. 2003.

[10] E.-G. Jung, J.-G. Lee, K.-S. Jhang, and D.-S. Har, "Differential value encoding for delay insensitive handshake protocol," *IEICE Trans. Inf. Syst.*, vol. E88, no. 7, pp. 1437–1444, Jul. 2005.

[11] Semiconductor Industry Association, International Technology Roadmap for Semiconductors 2007 [Online]. Available: http://www.itrs.net/Links/2007ITRS/Home2007.htm

[12] K. Y. Yun and D. L. Dill, "Automatic synthesis of extended burst-mode circuits: Part I. (Specification and hazard-free implementations)," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 2, pp. 101–117, Feb. 1999.

[13] K. Y. Yun and D. L. Dill, "Automatic synthesis of extended burst-mode circuits: Part II. (Automatic synthesis)," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 2, pp. 118–132, Feb. 1999.

[14] A. Katoch, M. Meijer, and S. K. Jain, "Active noise cancellation using aggressor-aware clamping circuit for robust on-chip communication," in *Proc. 18th Int. Conf. VLSI Des.*, 2005, pp. 325–329.

[15] J.-S. Huang, S.-W. Tu, and J.-Y. Jou, "On-chip bus encoding for LC cross-talk reduction," in *Proc. IEEE VLSI-TSA-DAT*, Apr. 2005, pp. 233–236.

[16] M. R. Becer, D. Blaauw, V. Zolotov, R. Panda, and I. N. Hajj, "Analysis of noise avoidance techniques in DSM interconnects using a complete crosstalk noise model," in *Proc. DATE Conf. Exhib.*, 2002, pp. 456–463.
[17] J. Bainbridge and S. Furber, "Chain: A delay-insensitive chip area interconnect," *IEEE Micro*, vol. 22, no. 5, pp. 16–23, Sep./Oct. 2002.
[18] *"STMicroelectronics Design Manual for CMOS 90 nm Technology,"* STMicroelectronics, Schaumburg, IL, 2005.
[19] H. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
[20] J. Sparsö and S. Furber, *Principles of Asynchronous Circuit Design*. Boston, MA: Kluwer, 2001.
[21] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Designs*. Piscataway, NJ: IEEE Press, 1990.
[22] J. P. Roth, "Diagnosis of automata failures: A calculus and a method," *IBM J. Res. Develop.*, vol. 10, no. 4, pp. 278–291, Jul. 1966.
[23] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.
[24] G. Birtwistle and K. S. Stevens, "The family of 4-phase latch protocols," in *Proc. 14th IEEE Int. Symp. ASYNC*, Apr. 2008, pp. 71–82.
[25] E. Amini, M. Najibi, Z. Jeddi, and H. Pedram, "FPGA implementation of gated clock based globally asynchronous locally synchronous wrapper circuits," in *Proc. ISSCS*, Jul. 2007, pp. 1–4.
[26] W. J. Bainbridge and S. J. Salisbury, "Glitch sensitivity and defense of quasi delay-insensitive network-on-chip links," in *Proc. 15th IEEE Int. Symp. ASYNC*, May 2009, pp. 35–44.
[27] C. S. D'Alessandro, A. Delong Shang Bystrov, A. V. Yakovlev, and O. Maevsky, "Phase-encoding for on-chip signalling," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 2, pp. 535–545, Mar. 2008.
[28] R. B. Lin, "Inter-wire coupling reduction analysis of bus-invert coding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 7, pp. 1911–1920, Aug. 2008.
[29] B. Fu and P. Ampadu, "On hamming product codes with type-II hybrid ARQ for on-chip interconnects," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 2042–2054, Sep. 2009.

Syed Rafay Hasan received the B.Eng. degree from NED University of Engineering and Technology, Karachi, Pakistan, in 1997 and the M.Eng. degree from Concordia University, Montreal, QC, Canada, in 2002. He is currently working toward the Ph.D. degree in the electrical and computer engineering with Concordia University.

He has been a part-time Professor with Concordia University since 2006. He has published ten peer-reviewed conference papers, abstracts, and technical reports. His research interests include design reliability of system-on-chips (SoCs), soft-error effects in VLSI circuits at terrestrial altitudes, globally asynchronous locally synchronous systems, and clock synchronization problems in SoCs designed in modern deep-submicrometer technologies.

Normand Bélanger (M'05) received the B.Ing., M.Sc.A., and Ph.D. degrees from École Polytechnique de Montréal, Montreal, QC, Canada, in 1987, 1989, and 1998, respectively.

After working for six years in industry, he joined academia as a Research Assistant with École Polytechnique de Montréal. His research interests include high-performance IC architectures, multiprocessor architectures, and application acceleration through hardware customization.

Yvon Savaria (S'77–M'86–SM'97–F'08) received the B.Ing. and M.Sc.A. degrees in electrical engineering from École Polytechnique de Montréal, Montreal, QC, Canada, in 1980 and 1982, respectively, and the Ph.D. degree in electrical engineering from McGill University, Montreal, in 1985.

Since 1985, he has been with École Polytechnique de Montréal, where he is currently a Professor and the Chairman of the Department of Electrical Engineering. He has published 82 journal papers and 322 conference papers and was the thesis advisor of 123 graduate students who completed their studies. He is the holder of 15 patents. He has carried work in several areas related to microelectronic circuits and microsystems such as testing, verification, validation, clocking methods, defect and fault tolerance, high-speed interconnects and circuit design techniques, CAD methods, reconfigurable computing and applications of microelectronics to telecommunications, image processing, video processing, radar signal processing, and digital signal processing acceleration.

Dr. Savaria was the Program Cochairman of the 1992 edition and the Chairman of the 1993 edition of the IEEE Workshop on Defect and Fault Tolerance in VLSI Systems. He was a Program Cochairman of the IEEE International Conference on Application-Specific Systems, Architecture and Processors (ASAP) 2006 and the General Cochair of ASAP'2007. He has been working as a Consultant or was sponsored for carrying research by CNRC, Design Workshop, Dolphin, DREO, Genesis, Gennum, Hyperchip, LTRIM, Miranda, MiroTech, Nortel, Octasic, PMC-Sierra, Technocap, Tundra, and VXP. He is a member of the Regroupement Stratégique en Microélectronique du Québec (RESMIQ) and the Ordre des Ingénieurs du Québec (OIQ) and is the Chairman of the Board of CMC Microsystems. He was awarded a Canada Research Chair (http://www.chairs.gc.ca) on design and architectures of advanced microelectronic systems in 2001. He was the recipient of the Synergy Award of the Natural Sciences and Engineering Research Council of Canada in 2006.

M. Omair Ahmad (S'69–M'78–SM'83–F'01) received the B.Eng. degree in electrical engineering from Sir George Williams University, Montreal, QC, Canada, and the Ph.D. degree in electrical engineering from Concordia University, Montreal.

From 1978 to 1979, he was a Faculty Member with the New York University College, Buffalo. In September 1979, he joined the Faculty of Concordia University as an Assistant Professor of computer science. Subsequently, he joined the Department of Electrical and Computer Engineering, Concordia University, where he was the Chair of the department from June 2002 to May 2005 and is currently a Professor. He holds the Concordia University Research Chair (Tier I) in Multimedia Signal Processing. He was a Founding Researcher at Micronet from its inception in 1990 as a Canadian Network of Centers of Excellence until its expiration in 2004. He was an Examiner of the Order of Engineers of Quebec. He has extensively published in the area of signal processing. He is the holder of four patents. His current research interests include the areas of multidimensional filter design, speech, image and video processing, nonlinear signal processing, communication DSP, artificial neural networks, and VLSI circuits for signal processing.

Dr. Ahmad was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I: FUNDAMENTAL THEORY AND APPLICATIONS from June 1999 to December 2001. He was the Local Arrangements Chairman of the 1984 IEEE International Symposium on Circuits and Systems. In 1988, he was a member of the Admission and Advancement Committee of the IEEE. He also served as the Program Cochair for the 1995 IEEE International Conference on Neural Networks and Signal Processing, the 2003 IEEE International Conference on Neural Networks and Signal Processing, and the 2004 IEEE International Midwest Symposium on Circuits and Systems. He was the General Cochair for the 2008 IEEE International Conference on Neural Networks and Signal Processing. He is currently the Chair of the Montreal Chapter IEEE Circuits and Systems Society. He was the recipient of numerous honors and awards, including the Wighton Fellowship from the Sandford Fleming Foundation, an induction to Provost's Circle of Distinction for career achievements, and the Award of Excellence in Doctoral Supervision from the Faculty of Engineering and Computer Science, Concordia University.