Diagrammatic Reasoning for Delay-Insensitive Asynchronous Circuits

Dan R. Ghica

University of Birmimingham

Abstract. In this paper we construct a new trace model of delay-insensitive asynchronous circuits inspired by Ebergen's model in such a way that it satisfies the compositional properties of a category, with additional monoidal structure and further algebraic properties. These properties taken together lay a solid mathematical foundation for a diagrammatic approach to reasoning about asynchronous circuits, which represents a formalisation of common intuitions about asynchronous circuits and their properties.

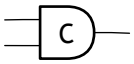
1 Asynchronous circuits

In the last few decades interest in asynchronous digital design ebbed and flowed. On the one hand, many studies have identified a great promise in asynchronous circuits, in particular low power consumption and modularity. On the other hand problems such as large silicone footprint and difficulties of fabrication hampered the adoption of asynchronous technology into the mainstream. These are just some of the well known advantages and disadvantages of the technology [1].

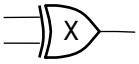
Another challenge raised by asynchronous design is that of reasoning about the correctness of circuits, and it has attracted a great deal of research interest. Several models of asynchronous circuits exists, such as Huffman [2] and burst-mode circuits [3], which fall in the broader category of *bounded-delay circuits*, and *delay-insensitive circuits*, of which a notable version are the so-called *micropipelines* [4].

The bounded-delay model takes explicitly into account the precise propagation delays of signals along circuit paths, or at least bounds on these delays. This is a fairly obvious model, but it has serious disadvantages. The first one is that computing delays is complicated, as propagation delays in a circuit can be data-dependent. The second one is that reasoning needs to be "geometric" rather than "topological", as wire lengths are highly relevant. This means that accurate reasoning can only be made after a circuit is placed and routed. Because one logical design can have a large number of concrete instantiations (*layouts*) this low-level way of reasoning is highly undersirable.

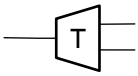
Far more attractive is the delay-insensitive model, which aims to design circuits that behave well no matter what the delays in the circuit. This is the model we will focus on. Typically, delay insensitive circuits are constructed out of a fixed set of primitive gates. Some of the most common are:



The *Muller C-element* is the typical synchronisation gate. It produces an output if it receives signals on both inputs.



The *exclusive or* is a merging gate, which outputs if it receives a signal on either input.



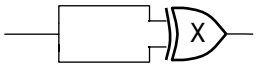
The *toggle* gate alternates (deterministically or nondeterministically) between the two outputs whenever it receives an input.

The *forking wire* can be seen as a gate which duplicates its input signal.

By *signal* we understand either a high-to-low or a low-to-high change in voltage on a pin. Other more complex gates can be introduced either as primitives or constructed out of these.

The main correctness challenge of the design of asynchronous circuits is to avoid so called "glitches": two signals which travel along the same wire can, if too close to each other, cancel each other out:

The reason is that the wires in a circuit are not ideal conductors but have capacitance, which acts like an inertial delay. If the signals are too close, they are "absorbed" by the capacitive inertia. A typical glitchy circuit is the one below:



If the two wires going into the XOR gate have different enough delays the circuit will output two signals, otherwise it will produce no output.

1.1 Ebergen's trace model

Glitchy circuits are obviously undersirable. In order to assess the absence of glitches a circuit must be modeled. The most widely used is a trace model due to Ebergen [5]. We will present it briefly below. By $K : A_1 \otimes \cdots \otimes A_m \to A'_1 \otimes \cdots \otimes A'_n$ let us denote a circuit K with inputs $A_1, \ldots, A_m$ and outputs $A'_1, \ldots, A'_n$ and let us denote by $[\![K]\!]$ the set of traces modelling that circuit, where each event represents an input/output on the port as identified by the label in the signature. Using the notation of regular expressions extended with interleaving $(- \mid -)$ and prefix closure $([-])$, the basic gates given above are modelled as:

$$
\begin{aligned}
[\![C : A_1 \otimes A_2 \to A']\!] &= \left[((A_1 \mid A_2) \cdot A')^*\right] \\
[\![X : A_1 \otimes A_2 \to A']\!] &= \left[(A_1 A' + A_2 A')^*\right] \\
[\![T : A \to A'_1 \otimes A'_2]\!] &= \left[(A A'_1 + A A'_2)^*\right] \\
[\![F : A \to A'_1 \otimes A'_2]\!] &= \left[(A \cdot (A'_1 \mid A'_2))^*\right].
\end{aligned}
$$

A trace model is also given for the wire:

$$[\![W : A \to A']\!] = \left[(AA')^*\right].$$

The composition on a common port is the usual "synchronisation and hiding" used in trace models of processes such as CSP [6, 7] and it can be used to model larger circuits.

Although the Ebergen trace model is useful and useable, it has practical and mathematical disadvantages. The behaviour of circuits is not fully defined, e.g. it is assumed that a Fork will not receive two consecutive inputs unless an output intervenes. This indeed corresponds to safe, glitch-free behaviour. However, in order to verify whether a circuit is safe, the formula for (de)composition needs to be elaborated and we must verify that, indeed, the outputs from one circuit do not violate the input assumptions on the other. Technically this is done by showing that there exists a correct projection of the composite traces onto the components. This is an awkward semantic condition of correctness, very difficult to check either by hand or automatically.

Another technical shortcoming of the Ebergen trace model is that the wire model is not an identity for composition because input-output alternation is not preserved automatically by composition. Composing $W : A_1 \to A_2$ with $W' : A_2 \to A_3$ allows the production of traces such as $A_1A_1A_3A_3$, which project correctly on the components but are not themselves wire-like in behaviour. This means that Wire is not even idempotent. This technical problem becomes an issue if we aim to structure asynchronous circuits into a category which, as discussed below, is highly desirable.

Acknowledgments This paper is inspired in goals and methodology by Abramsky, Coecke and their collaborators work on categorical, algebraic and diagrammatic foundations for quantum computing [8]. This work greatly benefitted from conversations with Peter Selinger, John Baez, Bob Coecke, Samson Abramsky, Prakash Panangaden, Bertrfried Fauser, Alex Smith, Paul B. Levy, Claudio Hermida and Paul-André Melliès.

2 Preliminaries

Let $in : A \to A + B$ be the usual injection and $out : A + B \rightharpoonup A$ be its section (a partial function). If $f : A \rightharpoonup B$ is a (partial) function let $f^* : A^* \to B^*$ be the (total) function defined as its *point-wise lifting* to the corresponding free monoids.

$$\begin{aligned} f^*(\epsilon) &= \epsilon \\ f^*(a \cdot w) &= f(a) \cdot f^*(w) && \text{if } f \text{ is defined at } a \in A \\ f^*(a \cdot w) &= f^*(w) && \text{if } f \text{ is not defined at } a. \end{aligned}$$

Let $\iota : A^* \to (A + B)^*$ be the retraction of the point-wise lifting of $out : A+B \rightharpoonup A$ and let $\omega = out^* : (A+B)^* \to A^*$. If $f \subseteq (X+Y)^*$ and $g \subseteq (Y+Z)^*$

we define $f \,||_Y\, g = \omega(\iota_1(f) \cap \iota_2(g)) = (f; \iota_1 \cap g; \iota_2); \omega$ where $\iota_1 : (X+Y)^* \to (X+Y+Z)^*, \iota_2 : (Y+Z)^* \to (X+Y+Z)^*, \omega : (X+Y+Z)^* \to (X+Z)^*$. Note above that it is often convenient to write function application in *diagrammatic order*, i.e. $f(g(a)) = a; g; f$.

We call $\iota : A^* \to (A+B)^*$ the *injection* of A into $A+B$, $\omega : (A+B)^* \to A^*$ the *projection* of $A+B$ onto A, and $f \,||_Y\, g$ the *composition* of f and g on Y. In general we will use the following notations:

$$\omega^X_{XY} = \omega : (X+Y)^* \to X^* \qquad \iota^{XY}_X = \iota : X^* \to (X+Y)^*.$$

The following properties are immediate.

Lemma 2.01 *1.* $\omega^X_{XY}; \iota^{XZ}_X = \iota^{XYZ}_{XY}; \omega^{XZ}_{XYZ}$.
2. for any $f, g \subseteq X^*$, $(f \cap g); \iota^{XY}_X = f; \iota^{XY}_X \cap g; \iota^{XY}_X$.
3. for any $f \subseteq (X+Y)^*, g \subseteq Y^*$, $(f; \omega^X_{XY} \cap g) = (f \cap g; \iota^{XY}_X); \omega^X_{XY}$.

Lemma 2.02 *Composition is associative, i.e. for any* $f \subseteq (X+Y)^*, g \subseteq (Y+Z)^*, h \subseteq (Z+U)^*$ *we have that* $f \,||_Y\, g \,||_Z\, h = f \,||_Y\, (g \,||_Z\, h)$.

Proof.

$$\begin{aligned}
LHS &= ((f; \iota^{XYZ}_{XY} \cap g; \iota^{XYZ}_{YZ}); \omega^{XZ}_{XYZ}; \iota^{XZU}_{XZ} \cap h; \iota^{XZU}_{ZU}); \omega^{XU}_{XZU} \\
&= ((f; \iota^{XYZ}_{XY} \cap g; \iota^{XYZ}_{YZ}); \iota^{XYZU}_{XYZ}; \omega^{XZU}_{XYZU} \cap h; \iota^{XZU}_{ZU}); \omega^{XU}_{XZU} && \text{from Lem. 2.01(1)} \\
&= ((f; \iota^{XYZ}_{XY}; \iota^{XYZU}_{XYZ} \cap g; \iota^{XYZ}_{YZ}; \iota^{XYZU}_{XYZ}); \omega^{XZU}_{XYZU} \cap h; \iota^{XZU}_{ZU}); \omega^{XU}_{XZU} && \text{from Lem. 2.01(2)} \\
&= (f; \iota^{XYZ}_{XY}; \iota^{XYZU}_{XYZ} \cap g; \iota^{XYZ}_{YZ}; \iota^{XYZU}_{XYZ} \cap h; \iota^{XZU}_{ZU}); \omega^{XZU}_{XYZU}; \omega^{ZU}_{XZU} && \text{from Lem. 2.01(3)} \\
&= (f; \iota^{XYZU}_{XY} \cap g; \iota^{XYZU}_{YZ} \cap h; \iota^{XYZU}_{ZU}); \omega^{XU}_{XYZU} && \text{from Lem. 2.01(2)} \\
&= RHS,
\end{aligned}$$

which can be brought to the same "ternary composition" form following similar algebraic manipulations.

The interleaving of two strings is a language which we define inductively on the length of the two strings. For sets it is applied pointwise to all pairs of strings.

$$\begin{aligned}
\epsilon \,|\, \epsilon &= \epsilon \\
XW \,|\, X'W' &= X \cdot (W \,|\, X'W') + X' \cdot (XW \,|\, W') \\
f \,|\, g &= \bigcup_{W \in f, W' \in g} W \,|\, W'.
\end{aligned}$$

Note that:

Proposition 2.03 . *If* $f : X \to Y$ *and* $f' : X' \to Y'$ *with* X, X', Y, Y' *all mutually disjoint then* $f \,|\, f' = f \,||_\emptyset\, f'$.

3 An affine model

In this section we will develop a simple, idealised model of asynchronous circuits in which there exists an idealised wire component behaving like a genuine identity. The model is developed in two stages. First we examine an *affine use* model in which every input is received at most once. This model is a simplified version of Ebergen's trace model. In subsequent sections we *lift* the model to a setting in which inputs can be received an arbitrary number of times. Unlike Ebergen's model, there will be no assumption of *seriality*, i.e. several inputs can be processed before the output is issued. The idealised wire model is the key component that allows the structuring of the model in a category. However, as explained in the previous section, this model is physically unrealisable, a weakness which we remedy separately.

The basic gates given above are modelled as before, except that Kleene and prefix closure are not required:

$$\begin{aligned}
[\![C : A_1 \otimes A_2 \to A']\!] &= (A_1 \mid A_2) \cdot A' \\
[\![X : A_1 \otimes A_2 \to A']\!] &= A_1 A' + A_2 A' \\
[\![T : A \to A'_1 \otimes A'_2]\!] &= A A'_1 + A A'_2 \\
[\![F : A \to A'_1 \otimes A'_2]\!] &= A \cdot (A'_1 \mid A'_2) \\
[\![W : A \to A']\!] &= A A' \\
[\![U : \emptyset \to A]\!] &= \epsilon \\
[\![E : A \to \emptyset]\!] &= A \\
[\![P : \emptyset \to A]\!] &= A.
\end{aligned}$$

In addition to the conventional gates we also have an open connector (W), a "dangling-input" connector (U) and a "dangling-output" connector (E). Finally, we have a one-pulse generator component P.

We introduce the following notations. If $f \subseteq (X + Y)^*$ we write $f : X \to Y$. If $f : X \to Y, g : Y \to Z$, with X, Y, Z mutually disjoint, then $f; g \stackrel{\text{def}}{=} f \,||_Y\, g$. For any $f : X \to Y, f' : X' \to Y'$, with X, X', Y, Y' mutually disjoint, then $f \otimes g \stackrel{\text{def}}{=} (f; inl^*) \,||_\emptyset\, (g; inr^*)$. It is immediate that in this case $f; g : X \to Z$ and $f \otimes g : X + X' \to Y + Y'$.

The definition of wire can be extended in the obvious way to that of a *bus*.

$$\begin{aligned}
W^0 &: \emptyset \to \emptyset, & W^0 &\stackrel{\text{def}}{=} id_\emptyset \\
W^k &: A_1 \otimes \cdots \otimes A_k \to A'_1 \otimes \cdots \otimes A'_k, & W^k &\stackrel{\text{def}}{=} W^{k-1} \otimes W.
\end{aligned}$$

Theorem 3.04 *Affinely-used asynchronous circuits form* category*, which we shall call* **AffAsy***, where*

1. *Objects are ports of shape $A_1 \otimes \cdots \otimes A_n$.*
2. *Morphisms $f : X \to Y$ are sets of traces $f \subseteq (X + Y)^*$.*
3. *Composition of morphisms $f : X \to Y, g : Y \to Z$ is defined as $f; g = f \,||_Y\, g$.*

4. *The identity morphism on X is W^n if $X = A_1 \otimes \cdots \otimes A_n$.*

Proof. The associativity of composition is Lem. 2.02. The fact that W^n is an identity is immediate.

Theorem 3.05 AffAsy *is a* symmetric monoidal category *where*

1. *The tensor of two objects is $X \otimes Y = X + Y$.*
2. *The tensor of two morphisms $f : X \to Y, g : Y \to Z$ is $f \otimes g : X \otimes X' \to Y \otimes Y'$.*
3. *The unit object is $\emptyset$.*
4. *The associator, left identity, right identity and symmetry are the corresponding isomorphisms for disjoint sum, lifted pointwise to sequences.*

Proof. We show that $\otimes$ is functorial for composition, i.e. if $f : X \to Y, g : Y \to Z, f' : X' \to Y', g' : Y' \to Z'$, then $(f \otimes f'); (g \otimes g') = f; f' \otimes g; g'$. Lets write $U = X + X' + Y + Y' + Z + Z'$. Expanding the definitions, the LHS is

$$((f; inl; \iota^{XX'YY'}_{XY} \cap f'; inr; \iota^{XX'YY'}_{X'Y'}); \iota^{U}_{XX'YY'} \cap (g; inl; \iota^{YY'ZZ'}_{YZ} \cap g'; inr; \iota^{YY'ZZ'}_{Y'Z'}); \iota^{U}_{YY'ZZ'}); \omega^{XX'ZZ'}_{U}$$

We use Lem. 2.01(2) and we combine consecutive injections to rewrite LHS as

$$LHS = (f; inl; \iota^{U}_{XY} \cap f'; inr; \iota^{U}_{X'Y'} \cap g; inl; \iota^{U}_{YZ} \cap g'; inr; \iota^{U}_{Y'Z'}); \omega^{XX'ZZ'}_{U}$$

Using similar algebraic manipulations the RHS can be brought to the same form.

The functoriality of $\otimes$ on identity is by definition. The coherence properties are the same as for disjoint sum and are preserved by point-wise lifting.

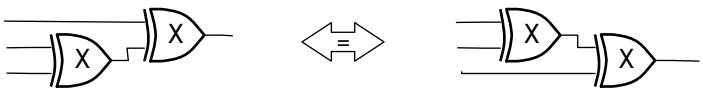
This model is, of course, limited in that it gives the wrong result for glitchy circuits, which do not behave in an affine way. For example $F; X = \emptyset$, which can be interpreted as the fact that this composition has no "safe" traces. An additional serious limitation is that linearity cannot model circuits where the output is fed back into an input port.

However, this model is a stepping stone which we shall elaborate towards more realistic behaviours in a way in which basic algebraic properties are preserved. Here are some of the main such properties.

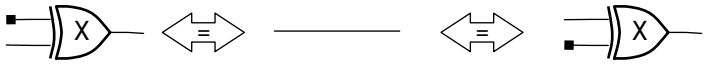
Theorem 3.06 *In* **AffAsy**

1. *(A, X, U) is a commutative monoid, with T a retract of X.*

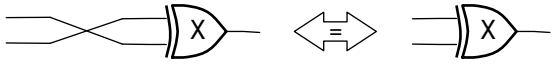
   Associativity $(W \otimes X); X = (X \otimes W); X$.



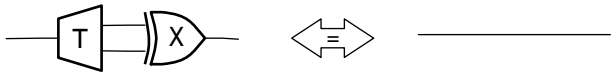
Unit $(U \otimes W); X = (W \otimes U); X = W$.



Commutativity $\gamma_A; X = X$.

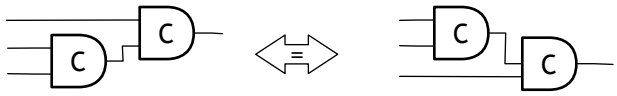


Retract $T; X = W$.



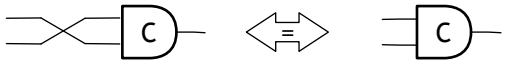
2. (A, C, P) *is a commutative monoid with* U *an absorbing element.*

Associativity $(W \otimes C); C = (C \otimes W); C$.

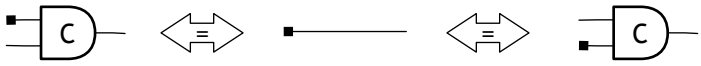


Unit $(P \otimes W); C = (W \otimes P); C = W$

Commutativity $\gamma_A; C = C$.

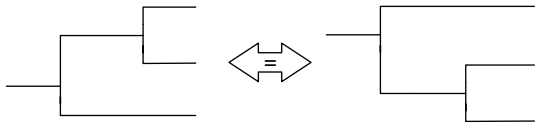


Absorbing element $(W \otimes U); C = (U \otimes W); C = U$



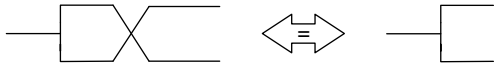
3. (A, F, E) *is a co-commutative co-monoid, with* C *a section of* F.

Co-associativity $F; (F \otimes W) = F; (W \otimes F)$.

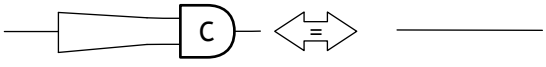


Co-unit $F; (W \otimes E) = F; (E \otimes W)$.

Co-commutativity $F; \gamma_A = F$.



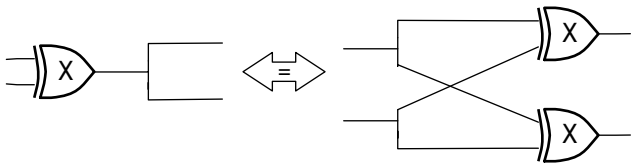
Section $F; C = W$



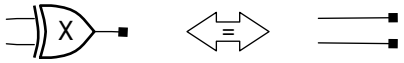
The non-trivial interplay of the basic gates gives rise to a richer algebraic structure:

Theorem 3.07 *1.* (A, X, E, F, U) *is a bialgebra.*

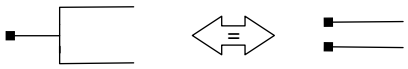
Distributivity $X; F = (F \otimes F); (W \otimes \gamma_A \otimes W); (X \otimes X)$.



Unit $E; F = E \otimes E$.

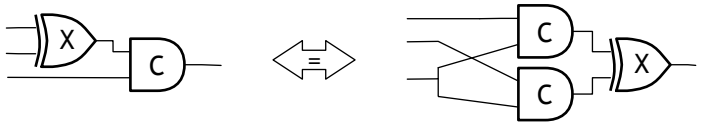


Co-unit $X; U = U \otimes U$.



2. (A, C, F, X) *is a Laplace pairing (in the sense of Rota, as per [9]).*

$$(X \otimes W); C = (W \otimes W \otimes F); (W \otimes \gamma_A \otimes W); (C \otimes C); X.$$



Proof. The proof is immediate from definitions.

The notion of *Laplace pairing* above is "categorified" in the obvious way from the conventional algebraic formulation: $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$.

The proofs of Thm. 3.06 and Thm. 3.07 are immediate from definitions and only involve simple calculations. Note that all the compositions above are "safe" in the sense that "no traces are lost in the composition". In Ebergen's terminology, the interaction between components has no *computation interference*. Exploiting equality in the presence of "interference" would allow us to introduce more equations (e.g. $T; C = \emptyset = F; X$), but we will see in the following section why such equations are not interesting.

4 An interleaved model

4.1 An idealised wire model

The next step is to "lift" the model from the previous affine use to unrestricted use. Given a set of traces f we define $!f$ as the smallest set of traces containing f, closed under self-interleaving. We define closure under self-interleaving as:

$f^0 = \emptyset, f^k = f \mid f^{k-1}, !f = \bigcup_{i \geq 0} f^i$. Note that if $f : X \to Y$ then $!f : X \to Y$. We define $\mathsf{C} = !C$, $\mathsf{X} = !X$, $\mathsf{T} = !T$, $\mathsf{F} = !F$, $\mathsf{W} = !W$, $\mathsf{U} = !U$, $\mathsf{E} = !E$, $\mathsf{P} = !P$, the models of basic components, closed under self-interleaving. This is a crucial difference between this model and Ebergen's, we do not assume serial use. However, the consequence is that the wire W has the physically unrealistic behaviour of an infinite-bounded buffer which can receive (and store) any n signals as inputs before issuing them as outputs.

Definition 4.11 *We say that* $f : X \to Y, g : Y \to Z$ compose safely *if and only if* $!(f;g) = !f;!g$.

Our notion of safety corresponds to Ebergen's computational noninterference. We can see that $T;C = !(T;C) = \emptyset$, whereas $\mathsf{T};\mathsf{C}$ includes traces such as AAA', with A the input and A' the output of the composition.

Lemma 4.12 *All the compositions in Thms. 3.06 and 3.07 are safe in the sense of Def. 4.11.*

Proof. Immediate, by inspection.

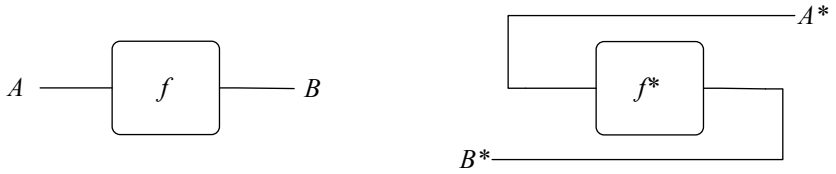
Lemma 4.13 *If* $f : X \to Y, f' : X' \to Y'$ *then* $!(f \otimes g) = !f \otimes !g$.

Proof. Immediate from Prop. 2.03.

Theorem 4.14 *Asynchronous circuits with an interleaved model form a compact closed category, called* **IdAsy** *where*

- *composition is defined as in* **AffAsy***;*
- *identity is* W*;*
- *the structural monoidal morphisms (associator, left identity, right identity, symmetry, unit, co-unit) are obtained by applying* $!-$ *to the corresponding structural morphisms in* **AffAsy***;*
- *objects are self-dual* $A^* = A$*;*
- *the unit* $\eta_A : I \to A_1^* \otimes A_2$ *and the co-unit* $\epsilon_A : A_1^* \otimes A_2 \to I$ *have the same sets of traces as the identity* $W : A_1 \to A_2$.

In the compact-closed category it is convenient to define the dual of a morphism $f : A \to B$ as $f^* : B^* \to A^*$, $f^* = (\eta_A \otimes id_{B^*});(id_{A^*} \otimes f \otimes id_{B^*});(id_{A^*} \otimes \epsilon_{B^*})$. This construct has an intuitive diagrammatic representation:



Note that:

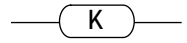
Lemma 4.15 $\mathsf{U}^* = \mathsf{E}$.

Theorem 4.16 *The algebraic structure of* **AffAsy** *is preserved by interleaving* $(!-)$ *in* **IdAsy***:*

- $(A, \mathsf{X}, \mathsf{U})$ *is a commutative monoid with* T *a retract of* X.
- $(A, \mathsf{C}, \mathsf{P})$ *is a commutative monoid with* U *an absorbing element.*
- $(A, \mathsf{F}, \mathsf{E})$ *is a co-commutative co-monoid with* C *a section of* F
- $(A, \mathsf{X}, \mathsf{E}, \mathsf{F}, \mathsf{U})$ *is a bialgebra.*
- $(A, \mathsf{C}, \mathsf{F}, \mathsf{X})$ *is a Laplace pairing.*

Proof. Composition in **IdAsy** is defined like in **AffAsy** and it is associative. W is an identity immediately from Lem. 4.14(1) because composition with W is safe for any morphism. Similarly, all the equations defining the symmetric compact closed structure involve only safe compositions, so are preserved by self-interleaving. The equations involved in Thms. 3.06 and 3.07 are also constructed out of safe compositions (Lem. 4.12).

The safety requirement in Lem. 4.12 is essential since self-interleaving may introduce new traces in unsafe compositions. For example $F; X = \emptyset$ but $F; X^2 = AA'A'$, and in fact $\mathsf{F}; \mathsf{X} = !(AA'A')$ where $F; X : A \to A'$. This example also illustrates the physically unrealistic nature of this model, because consecutive signals $A'A'$ are never absorbed by the wire capacitance. A realistic model should give $\mathsf{F}; \mathsf{X} = !(A \cdot (A'A' + \epsilon))$, reflecting the fact that in this composition consecutive A' signals may, non-deterministically, disappear.

To prepare the ground for a more realistic model we introduce a new component $\mathsf{K} : A \to A'$ defined as $[\![\mathsf{K}]\!] = !(AA' + AA)$. We call this component a *capacitive wire* and we represent it as 

. Not accidentally, this is reminiscent of the symbol conventionally used for unknown bounded delay; this is an unknown bounded capacitance. Its behaviour is to either propagate a signal correctly or to absorb consecutive inputs, non-deterministically.

Let $|t|$ be the length of a sequence. Let $t \sqsubseteq t'$ denote a prefix t of a sequence t'. It is easy to see that all sequences in K have more inputs than outputs in any prefix and, overall, an even number of outputs can be lost.

Lemma 4.17 *Let* $\iota_k : (A_1 + A_2)^* \to A_i$ *for* $k = 1, 2$. $t \in [\![\mathsf{K} : A_1 \to A_2]\!]$ *if and only if both these conditions hold:*

- *there exsists* $k \in \mathbb{N}$ *such that* $|t; \iota_1| - |t; \iota_2| = 2k$
- *for any prefix* $p \sqsubseteq t$, $|p; \iota_1| \geq |p; \iota_2|$.

The capacitive wire has the following important property:

Lemma 4.18 *1.* $\mathsf{K} : A \to A$ *is idempotent, i.e.* $\mathsf{K}; \mathsf{K} = \mathsf{K}$.
2. $\mathsf{K}^n : A^n \to A^n$ *is idempotent, i.e.* $\mathsf{K}^n; \mathsf{K}^n = \mathsf{K}^n$.

Proof. The idempotence of K is proved showing that the two conditions of Lem. 4.17 are preserved by composition. If the first capacitive wire loses $2k$ signals and the second capacitive wire loses $2k'$ signals then their composition loses $2(k + k')$ signals, which is also a valid trace in a capacitive wire.

The second property follows immediately from the naturality of $\otimes$.

4.2 A capacitive wire model

We are now in a position to give a more physically accurate account of asynchronous circuits by removing the idealised wire W from the set of basic components. However, this raises a technical problem because W plays a structural role in the category as the identity. Also, the traces of the compact-closed unit (η_A) and co-unit (ϵ_A) behave like idealised wires. In order to restore the categorical structure we use the following standard construction.

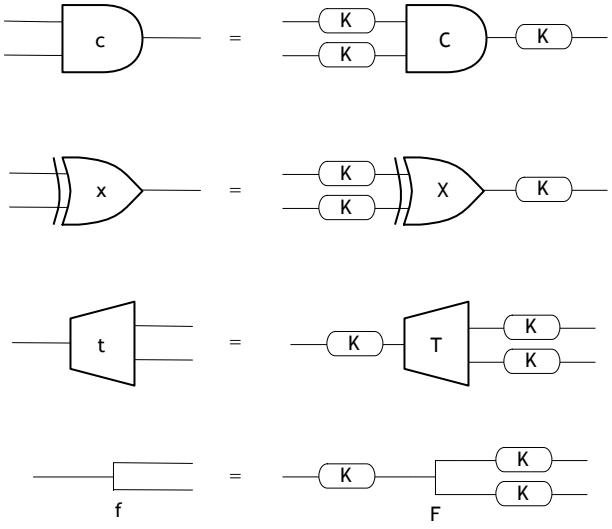
Definition 4.21 *The* Karoubi envelope *of category* $\mathbf{C}$*, sometimes written* $\mathbf{Split}(\mathbf{C})$*, is the category whose objects are pairs of the form* (A, e) *where* A *is an object of* $\mathbf{C}$ *and* $e : A \to A$ *is an idempotent of* $\mathbf{C}$*, and whose morphisms are triples of the form* $(e, f, e') : (A, e) \to (A, e')$ *where* $f : A \to A'$ *is a morphism of* $\mathbf{C}$ *satisfying* $f = e; f; e'$*.*

Definition 4.22 *The category of* delay-insensitive asynchronous circuits *is defined as* $\mathbf{DIAsy} = \mathbf{Split}(\mathbf{IdAsy})$*.*

The basic morphisms of the **DIAsy** category are

1. $\mathsf{c} = (\mathsf{K} \otimes \mathsf{K}); \mathsf{C}; \mathsf{K}$
2. $\mathsf{x} = (\mathsf{K} \otimes \mathsf{K}); \mathsf{X}; \mathsf{K}$
3. $\mathsf{t} = \mathsf{K}; \mathsf{T}; (\mathsf{K} \otimes \mathsf{K})$
4. $\mathsf{f} = \mathsf{K}; \mathsf{F}; (\mathsf{K} \otimes \mathsf{K})$.

The physical and diagrammatic representation of this construction is that all basic gates have capacitive wires as connectors:



Lemma 4.23 *1.* $\mathsf{c} = \mathsf{C}; \mathsf{K} = (\mathsf{K} \otimes \mathsf{K}); \mathsf{C}$
2. $\mathsf{x} = \mathsf{X}; \mathsf{K} = (\mathsf{K} \otimes \mathsf{K}); \mathsf{X}$

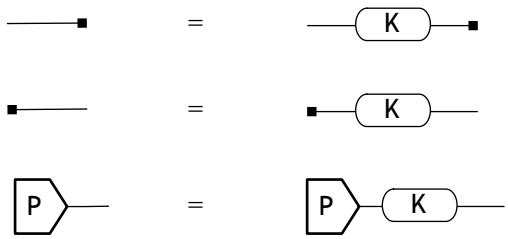
3. $\mathsf{t} = \mathsf{T}; (\mathsf{K} \otimes \mathsf{K}) = \mathsf{K}; \mathsf{T}$
4. $\mathsf{f} = \mathsf{F}; (\mathsf{K} \otimes \mathsf{K}) = \mathsf{K}; \mathsf{F}$.

Proof. From Lem. 4.17. For example, in the case of c if m respectively m' signals arrive as input, $m - 2k$ respectively $m' - 2k'$ reach the C-gate and $min(m - 2k, m' - 2k')$ are output. For post-composition with K, $min(m, m') - 2k'' = min(m - 2k'', m' - 2k'')$ signals are output. For any choice of k, k' we take $k'' = max(k, k')$.

We also have

Proposition 4.24 1. $\mathsf{e} = \mathsf{K}; \mathsf{E} = \mathsf{E}$
2. $\mathsf{u} = \mathsf{U}; \mathsf{K} = \mathsf{U}$
3. $\mathsf{p} = \mathsf{P}; \mathsf{K} = \mathsf{P}$.

Diagrammatically:



The first two equalities are obvious. For the last one, since P generates an *arbitrary* number of signals it does not matter than some of them are lost.

Theorem 4.25 *The category of delay-insensitive asynchronous circuits* **DIAsy** *is compact closed with*

1. *dual objects* $(A, \mathsf{K})^* = (A^*, \mathsf{K}^*)$
2. *unit* $\overline{\eta}_A : I \to A^* \otimes A$ *defined as* $\overline{\eta}_A = \eta_A; (\mathsf{K}^* \otimes \mathsf{K})$*;*
3. *co-unit* $\overline{\epsilon}_A : A^* \otimes A \to I$ *defined as* $\overline{\epsilon}_A = (\mathsf{K}^* \otimes \mathsf{K}); \epsilon_A$.

Diagrammatically, the unit and co-unit of the closed structure are:

Moreover, the fact that K is an idempotent means that the existing algebraic structure is preserved by the construction.

Theorem 4.26 *The algebraic structure of* **AffAsy** *and* **IdAsy** *is preserved in* **DIAsy***:*

– $(A, \mathsf{x}, \mathsf{u})$ *is a commutative monoid with* t *a retract of* x.

– $(A, \mathsf{c}, \mathsf{p})$ *is a commutative monoid with* u *an absorbing element.*
– $(A, \mathsf{f}, \mathsf{e})$ *is a co-commutative co-monoid with* c *a section of* f
– $(A, \mathsf{x}, \mathsf{e}, \mathsf{f}, \mathsf{u})$ *is a bialgebra.*
– $(A, \mathsf{c}, \mathsf{f}, \mathsf{x})$ *is a Laplace pairing.*

Proof. Immediate. For example distributivity in the bialgebra $((A, \mathsf{x}, \mathsf{e}, \mathsf{f}, \mathsf{u})$ is:

$$\begin{aligned}
&(\mathsf{f} \otimes \mathsf{f}); (\mathsf{K} \otimes ((\mathsf{K} \otimes \mathsf{K}); \gamma_A) \otimes \mathsf{K}); (\mathsf{x} \otimes \mathsf{x}) \\
&\quad = \mathsf{K}^2; (\mathsf{F} \otimes \mathsf{F}); (\mathsf{W} \otimes \gamma \otimes \mathsf{W}); \mathsf{X}^2; \mathsf{K}^2 && \text{(Lem. 4.23)} \\
&\quad = \mathsf{K}^2; \mathsf{X}; \mathsf{F}; \mathsf{K}^2 && \text{(Thm. 4.16)} \\
&\quad = \mathsf{x}; \mathsf{f}. && \text{(Lem. 4.23)}
\end{aligned}$$

Note that the proposition above involves circuits in a realistic model of glitchy circuits. The fact that we use the idealised connector W in proofs does not detract from the realism of the model.

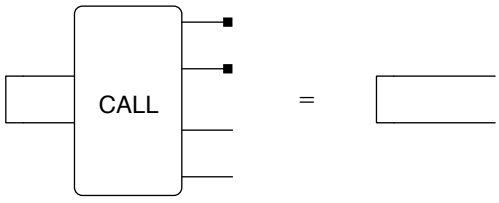
To conclude, the category we have constructed has a complex trace model, which incorporates circuits with glitches. Reasoning directly in the trace model is awkward. However, its compact closed structure and rich algebraic properties provide a useful framework in which reasoning can be carried out more abstractly, algebraically or diagrammatically.

5 Applications

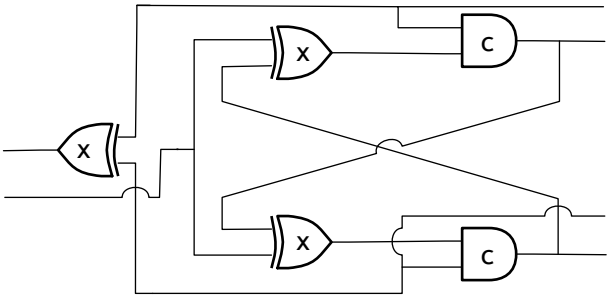
The Geometry of Synthesis project [10–12] shows how a higher-level programming language can be compiled directly into static asynchronous circuits, more specifically Event Logic, starting from its game semantic model [13]. The model of asynchronous circuits used there is based on the category **DIAsy** of delay-insensitive circuits but reasoning is carried out at the level of traces, and is tedious. As an application, we will show just one of the equivalences that needs to be proved in order to prove the soundness of the technique, and we do it in a purely algebraic or diagrammatic fashion [11].

We first introduce the Event Logic component $\mathsf{CALL} : A^* \otimes A \to (A^* \otimes A) \otimes (A^* \otimes A)$. It works as a stateful multiplexer-demultiplexer circuit between one occurrence of $A^* \otimes A$ on the left and two on the right. In [11] it is used to implement the *diagonal morphism* in a Cartesian category of circuits of particular shape. One of the required equations of the Cartesian product, mapped into circuits, amounts to showing that $\overline{\eta}_A; \mathsf{CALL}; (\mathsf{u}^* \otimes \mathsf{e} \otimes \overline{id}^2) = \overline{\eta}_A$. Diagrammatically, this is:

Lemma 5.07



The implementation of CALL, not a basic circuit, is given below.



In the category, the construction is:

$$\mathsf{CALL} = (\mathsf{x}^* \otimes \overline{id}); (\overline{id}^* \otimes \overline{\gamma}); (\overline{id}^* \otimes \overline{\eta} \otimes \overline{id} \otimes \overline{\eta} \otimes \overline{id}^*); \\ (\mathsf{f}^* \otimes \overline{id} \otimes \overline{id}^* \otimes \overline{id} \otimes \mathsf{f}^*); (\overline{id}^* \otimes \mathsf{DW} \otimes \overline{id}^*),$$

where

$$\mathsf{DW} = (\overline{id} \otimes \mathsf{f} \otimes \overline{id}); (\overline{id}^2 \otimes \overline{\eta}^2 \otimes \overline{id}^2); (\overline{id} \otimes \mathsf{x} \otimes \overline{\gamma} \otimes \mathsf{x} \otimes \overline{id}); \\ (\mathsf{c} \otimes \overline{id}^2 \otimes \mathsf{c}); (\mathsf{f} \otimes \overline{id}^2 \otimes \mathsf{f}); (\overline{id} \otimes \overline{\epsilon}^2 \otimes \overline{id})$$

In the structural morphisms, if the index is not specified, it is by default A, so $\overline{\eta} = \overline{\eta}_A$, and so on. It is quite clear that trace-level reasoning about such circuits is extremely difficult! On the other hand, the algebraic proof of Lem. 5.07 is a sequence of straightforward calculations using categorical and algebraic properties. The diagrammatic representation of the proof, as a rewriting of the circuit is given in Fig. 1.

The key circuit simplifications come out the fact that e and x are either unit or co-unit or absorbing element for x, c, f taken as (co)monoids. This process of reduction results in the circuit $\overline{\eta}; \overline{\gamma}; (\overline{id} \otimes \overline{\eta} \otimes \overline{id}); (\mathsf{c} \otimes \mathsf{f}^*)$. Standard diagrammatic reasoning using the compact-closed structure allows bringing the circuit to the simpler form $\overline{\eta}; ((\mathsf{f}; \mathsf{c}) \otimes \overline{id})$ where using the fact that c is a section of f completes the proof.

6 Conclusion

In this paper we have constructed a trace model for delay-insensitive asynchronous circuits similar to Ebergen's, but generalised to handle glitchy behaviour. We showed that even in the absence of an idealised connector, which would behave naturally as an identity for compositions, such circuits can be structured in a category by taking advantage of the Karoubi envelope construction where the idempotent is a realistic connector of unknown capacitance. We further show that even though the trace model is complicated and very awkward as a basis for reasoning about such circuits, they enjoy many algebraic properties

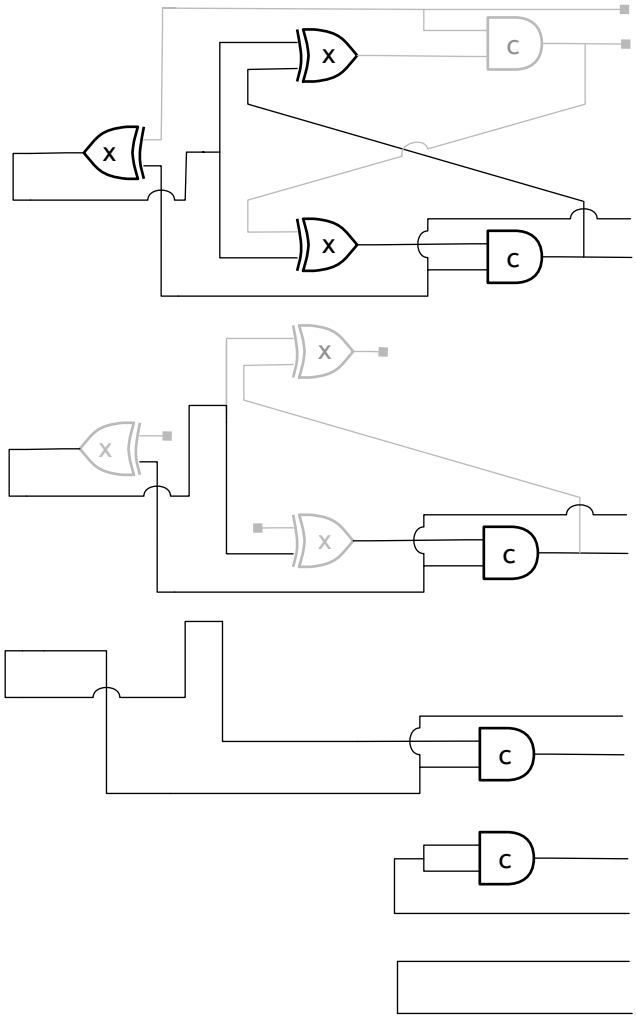
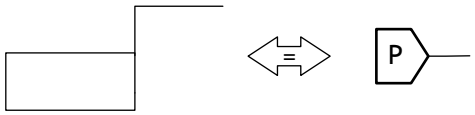


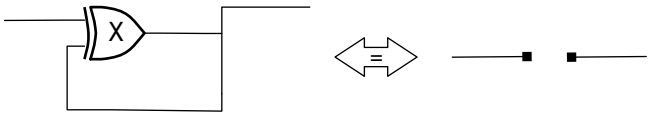
Fig. 1. Proof of Lem. 5.07

which allow diagrammatic reasoning consistent with common intuitions about such circuits. Such properties seem promising as a starting point for mechanised reasoning via, for example, circuit rewriting.

The most severe limitation of this model is its handling of circuits where feedback leads to non-terminating behavior. It is only a model of terminating computations, represented as complete traces. Technically, this is due to the fact that, unlike Ebergen, we do not adopt prefix closure. Prefix-closure cannot be naively introduced because causality loops created by feedback can lead to unrealistic solutions. For example this circuit



would be trace-equivalent to p (pulse), when in fact it is equivalent to u (dangling input). Our model instead equates



with $\mathsf{u};\mathsf{e}$ (an unresponsive circuit) which is sound but incomplete. To fix this problem causality, which we currently ignore, needs to be introduced in the model.

In a more theoretical direction it would be interesting to examine how the specific algebraic structures arising in asynchronous circuits interact with the generic framework introduced by Burroni [14] and further developed by Lafont [15], in which boolean circuits can be reduced to unique canonical forms. These notions are essential if we aim to automate reasoning about asynchronous circuits.

References

1. Hauck, S.: Asynchronous design methodologies: An overview. Proceedings of the IEEE **83**(1) (1995) 69–93
2. Huffman, D.: The synthesis of sequential switching circuits. Journal of the Franklin Institute **257**(3) (1954) 161–190
3. Yun, K., Dill, D.: Automatic synthesis of extended burst-mode circuits I (specification and hazard-free implementations). Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **18**(2) (1999) 101–117
4. Sutherland, I.: Micropipelines. Communications of the ACM **32**(6) (1989) 720–738
5. Ebergen, J.: A formal approach to designing delay-insensitive circuits. Distributed Computing **5**(3) (1991) 107–119
6. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall (1985)
7. Josephs, M.B., Udding, J.T.: Delay-insensitive circuits: An algebraic approach to their design. In Baeten, J.C.M., Klop, J.W., eds.: CONCUR. Volume 458 of Lecture Notes in Computer Science., Springer (1990) 342–366

8. Abramsky, S., Coecke, B.: Categorical quantum mechanics. Handbook of quantum logic and quantum structures: quantum logic (2008) 261–324
9. Fauser, B.: On the Hopf-algebraic origin of Wick normal-ordering. Journal of Physics A: Mathematical and General **34**(105) (2001)
10. Ghica, D.R.: Geometry of Synthesis: A structured approach to VLSI design. In Hofmann, M., Felleisen, M., eds.: The ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), ACM (2007) 363–375
11. Ghica, D.R., Smith, A.: Geometry of Synthesis II: From games to delay-insensitive circuits. Electr. Notes Theor. Comput. Sci. **265** (2010) 301–324
12. Ghica, D.R., Smith, A.: Geometry of Synthesis III: Resource management through type inference. In Ball, T., Sagiv, M., eds.: The ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), ACM (2011) 345–356
13. Ghica, D.R., Murawski, A.S., Ong, C.H.L.: Syntactic control of concurrency. Theor. Comput. Sci. **350**(2-3) (2006) 234–251
14. Burroni, A.: Higher-dimensional word problems with applications to equational logic. Theoretical computer science **115**(1) (1993) 43–62
15. Lafont, Y.: Towards an algebraic theory of boolean circuits. Journal of Pure and Applied Algebra **184**(2) (2003) 257–310