Defining Elastic Circuits with Negative Delays

Sava Krstić[1], Jordi Cortadella[2], Mike Kishinevsky[1], and John O'Leary[1]

[1] Strategic CAD Labs, Intel Corporation, Hillsboro, Oregon, USA
[2] Universitat Politècnica de Catalunya, Barcelona, Spain

An elastic circuit can take an arbitrarily long time to compute its results and it can wait arbitrarily long for cooperating circuits to produce its inputs and to consume its outputs. For example, the streams $X = \mathbf{1}\square\mathbf{00}\square\square\mathbf{1}\square\cdots$ and $Y = \square\square\mathbf{0}\square\mathbf{1}\square\square\mathbf{1}\square\mathbf{0}\cdots$ could be an observed input/output behavior of an elastic inverter. The transfer of data on the input channel occurs at cycles 1,3,4,7,...; the absence of transfer is shown as the bubble symbol $\square$. When we ignore the bubbles, we see a behavior of an ordinary inverter: $X' = \mathbf{1001}\cdots$ and $Y' = \mathbf{0110}\cdots$.

Elastic circuits promise novel methods for microarchitectural design that can exploit variable latencies, while still employing standard synchronous design tools and methods [1]. The Intel project SELF (Synchronous Elastic Flow) [3, 2] demonstrates the industry interest. In SELF, every elastic circuit $\mathcal{E}$ is associated with an ordinary (non-elastic) system $\mathcal{S}$ in the "bubble removal" sense indicated in the inverter example. For each wire X of $\mathcal{S}$, there is a *channel* $\langle D_X, V_X, S_X\rangle$ in $\mathcal{E}$ consisting of the *data* wire D_X, and the handshake wires V_X and S_X (*valid*, *stop*). A transfer along the channel occurs when $V_X = 1$ and $S_X = 0$, thus requiring producer-consumer cooperation.

The first formal account of elasticity was given by the *patient processes* of [1]. Our *elastic machines* [5] are a more readily applicable theoretical foundation, but even this is insufficient for modeling relevant subtler aspects, notably anti-token counterflow [2]. Here, we would like to discuss the progress we have made in devising a more general theory and the challenges we encountered.

An adequate theory needs to provide a definiton of elastic circuits that makes two fundamental results provable: (1) the *liveness theorem* that guarantees the infinite flow of tokens whenever an elastic circuit is put in an environment that offers communication on each channel infinitely often; (2) the *elastic compositionality theorem* guaranteeing that, under reasonable assumptions, if elastic circuits $\mathcal{E}_1, \ldots, \mathcal{E}_n$ implement the systems $\mathcal{S}_1, \ldots, \mathcal{S}_n$ and if $\mathcal{S}$ is the network obtained by connecting some wires of the systems $\mathcal{S}_i$, then connecting the corresponding channels (wire triples) of the elastic circuits $\mathcal{E}_i$ produces a new elastic circuit which implements $\mathcal{S}$.

As in [5], we model ordinary (non-elastic) systems as stream transformers, but now we use a more precise notion of causality: given an inputs-by-outputs integer matrix $\delta = \|\delta_{XY}\|$, we call a system δ-*causal* if the nth Y-output depends only on the first $n - \delta_{XY}$ X-inputs. In [5], we dealt only with systems satisfying $\delta_{XY} \geq 0$ ("circuits") and had a compositionality theorem based on the absence of suitably defined "combinational loops". The new setup admits systems with negative delays (like the *tail* function on streams), but we can still prove compositionality, this time based on Wadge's *cycle sum condition* [6, 4].

Now, given a δ-causal system $\mathcal{S}$, when do we say that a system $\mathcal{E}$ is an elastic implementation of $\mathcal{S}$? In addition to a simple data-correctness property, $\mathcal{E}$ also needs to satisfy liveness properties $\mathsf{G}\,(\mathsf{G}\,\mathsf{en\_in}_X \rightarrow \mathsf{F}\,\bar{S}_X)$ and $\mathsf{G}\,(\mathsf{G}\,\mathsf{en\_out}_Y \rightarrow \mathsf{F}\,V_Y)$ for all inputs X and outputs Y. Surprisingly, the enabling predicates $\mathsf{en\_in}_X$ and $\mathsf{en\_out}_Y$ are difficult to guess. By introducing the "backward delay numbers" δ_{YX}, we can express (as $\delta_{XY}+\delta_{YX}$) the capacity of the fifos between X and Y, and make a formal connection with a marked graph model of $\mathcal{E}$. We expect the predicates $\mathsf{en\_in}_X$ and $\mathsf{en\_out}_Y$ should use no parameters other than the delay numbers, and the elastic compositionality theorem to be provable assuming only the absence of cycles with a negative δ-sum. With a current definition, however, we can prove the theorem only with additional assumptions. The right generalization of our enabling predicates, and with it, the right definition of elastic circuits, remains elusive.

References

1. L. Carloni, K. McMillan, and A. Sangiovanni-Vincentelli. Theory of latency-insensitive design. *IEEE Trans. Comput.-Aided Design Integrated Circuits*, 20(9):1059–1076, Sept. 2001.
2. J. Cortadella and M. Kishinevsky. Synchronous elastic circuits with early evaluation and token counterflow. In *DAC '07: Proceedings of the 44th annual Conference on Design Automation*, pages 416–419, New York, NY, USA, 2007. ACM.
3. J. Cortadella, M. Kishinevsky, and B. Grundmann. Synthesis of synchronous elastic architectures. In *DAC '06: Proceedings of the 43rd annual Conference on Design Automation*, pages 657–662, New York, NY, USA, 2006. ACM.
4. S. J. Gay and R. Nagarajan. Intensional and extensional semantics of dataflow programs. *Formal Asp. Comput.*, 15(4):299–318, 2003.
5. S. Krstić, J. Cortadella, M. Kishinevsky, and J. O'Leary. Synchronous elastic networks. In *FMCAD '06: Proceedings of the Formal Methods in Computer Aided Design*, pages 19–30, Washington, DC, USA, 2006. IEEE Computer Society.
6. W. W. Wadge. An extensional treatment of dataflow deadlock. In *Proceedings of the International Sympoisum on Semantics of Concurrent Computation*, pages 285–299, London, UK, 1979. Springer-Verlag.