Synchronous Elastic Circuits

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

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

Synchronous elastic circuits (also known as latency-insensitive and latency-tolerant) behave independently of the latencies of computations and communication channels. For example, the three sequences

$$X = \langle 1, *, *, 2, *, 5, 3, \ldots \rangle \quad Y = \langle 2, *, 0, *, 1, *, 4, \ldots \rangle \quad Z = \langle *, 3, *, 2, *, *, 6, *, 7, \ldots \rangle$$

are an acceptable behavior of an elastic adder with input channels X, Y and output channel Z, where the absence of transfer on a particular channel at a given cycle is indicated by $*$. Indeed, the associated transfer subsequences (obtained by deleting the $*$'s) make up a behavior of an ordinary (non-elastic) adder:

$$X' = \langle 1, 2, 5, 3, \ldots \rangle \quad Y' = \langle 2, 0, 1, 4, \ldots \rangle \quad Z' = \langle 3, 2, 6, 7, \ldots \rangle$$

Current interest in elasticity is motivated by the difficulties with timing and communication in large synchronous designs in nanoscale technologies. The time discretization imposed by synchronicity forces to take early decisions that often complicate changes at the latest stages of the design or efficient design scaling. In modern technologies, calculating the number of cycles required to transmit data from a sender to a receiver is a problem that often cannot be solved until the final layout has been generated. Elastic circuits promise novel methods for microarchitectural design that can use variable latency components and tolerate static and dynamic changes in communication latencies, while still employing standard synchronous design tools and methods.

We will first present a simple elastic protocol, called SELF (Synchronous Elastic Flow) and describes methods for an efficient implementation of elastic systems and for the conversion of regular synchronous designs into an elastic form. Every elastic circuit $\mathcal{E}$ implements the behavior of an associated standard (non-elastic) circuit $\mathcal{C}$, as in the adder example above. For each wire X of $\mathcal{C}$, there are three in $\mathcal{E}$: the *data* wire D_X, and the single-bit control wires V_X and S_X (*valid* and *stop*). This triple of wires is a *channel* of $\mathcal{E}$. A transfer along the channel occurs when $V_X = 1$ and $S_X = 0$, thus requiring cooperation of the producer and the consumer. [CKG06] provides more details on the implementation of SELF.

We will next review theoretical foundations of SELF. Our main result states that (under favorable circumstances) "the network of elasticizations is an elasticization of the given network": if we have elastic circuits $\mathcal{E}_1, \ldots, \mathcal{E}_n$ implementing standard circuits $\mathcal{C}_1, \ldots, \mathcal{C}_n$ and if $\mathcal{C}$ is a standard network obtained by connecting some wires of the circuits $\mathcal{C}_i$, then connecting the corresponding channels

(wire triples) of the elastic circuits $\mathcal{E}_i$ will produce a new elastic circuit which implements $\mathcal{C}$. As a special case, we prove the characteristic property of elastic circuits: plugging an empty elastic buffer in a channel of an elastic network produces an equivalent elastic network. The details of the theory can be found in [KCKO06].

Related Work

Some researchers advocate for the modularity and efficiency of asynchronous circuits to devise a beter methodology for complex digital systems. However, asynchronous circuits require a significantly different design style and the CAD support for such circuits is still in its prehistory.

Our work addresses the following question: is there an *efficient* scheme that combines the modularity of asynchronous systems with the simplicity of synchronous implementations?

Other authors have been working towards this direction. Latency-insensitive (LI) schemes [CMSV01] were proposed to separate communication from computation and make the systems insensitive to the latencies of the computational units and channels. The implementation of LI systems is synchronous [CSV02, CN01] and uses *relay stations* at the interfaces between computational units.

In a different scenario, *synchronous interlocked pipelines* [JKB+02] were proposed to achieve fine-grained local handshaking at the level of stages. The implementation is conceptually similar to a discretized version of traditional asynchronous pipelines with request/acknowledge handshake signals.

A *de-synchronization* [HDGC04,BCK+04] approach automatically transforms synchronous specifications into asynchronous implementations by replacing the clock network with an asynchronous controller. The success of this paradigm will depend on the attitude of designers towards accepting asynchrony in their design flow.

References

[BCK+04] I. Blunno, J. Cortadella, A. Kondratyev, L. Lavagno, K. Lwin, and C. Sotiriou. Handshake protocols for de-synchronization. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 149–158. IEEE Computer Society Press, April 2004.

[CKG06] J. Cortadella, M. Kishinevsky, and B. Grundmann. SELF: Specification and design of a synchronous elastic architecture for DSM systems. In *TAU'2006: Handouts of the International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, February 2006. Available at `www.lsi.upc.edu/~jordicf/gavina/BIB/reports/self_tr.pdf`.

[CMSV01] L. Carloni, K.L. McMillan, and A.L. Sangiovanni-Vincentelli. Theory of latency-insensitive design. *IEEE Transactions on Computer-Aided Design*, 20(9):1059–1076, September 2001.

[CN01] Tiberiu Chelcea and Steven M. Nowick. Robust interfaces for mixed-timing systems with application to latency-insensitive protocols. In *Proc. ACM/IEEE Design Automation Conference*, June 2001.

[CSV02] L.P. Carloni and A.L. Sangiovanni-Vincentelli. Coping with latency in SoC design. *IEEE Micro, Special Issue on Systems on Chip*, 22(5):12, October 2002.

[HDGC04] G.T. Hazari, M.P. Desai, A. Gupta, and S. Chakraborty. A novel technique towards eliminating the global clock in VLSI circuits. In *Int. Conf. on VLSI Design*, pages 565–570, January 2004.

[JKB+02] Hans M. Jacobson, Prabhakar N. Kudva, Pradip Bose, Peter W. Cook, Stanley E. Schuster, Eric G. Mercer, and Chris J. Myers. Synchronous interlocked pipelines. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pages 3–12, April 2002.

[KCKO06] S. Krstić, J. Cortadella, M. Kishinevsky, and J. O'Leary. Synchronous elastic networks. Technical Report. Available at `www.lsi.upc.edu/~jordicf/gavina/BIB/reports/elastic_nets.pdf`, 2006.