Asynchronous or synchronous? A misleading choice.

Scott Fairbanks and Simon Moore

15 May 2004

This work suggests that synchrony and asynchrony are not design choices but rather a collection of design choices. We believe the terms are actually counter-productive because they prevent designers from evaluating the full space of design options available to them.

Asynchronous design is often advertised as the technology that will progress the art of computation when the timing assumptions that hold the synchronous design paradigm together are no longer feasible. Recent papers from the asynchronous design community [1, 2] outline methods to build synchronous systems using asynchronous foundations. The authors in [1] described how to use asynchronous latch controllers to replace the clock distribution apparatus in a way that was transparent to the synchronous designer. The authors in [2] used locally communicating handshake circuitry to generate and globally distribute timing signals.

The first paper questions the synchronous assumptions of simultaneity in sequencing while retaining the rigid spacing of instructions. The second paper enforces global simultaneity while deriving timing from local conditions and handshake circuitry. Each paper chose to use elements of so-called asynchronous design in the construction of a so-called synchronous system.

These papers exposed the possibility that the terms synchronous and asynchronous have come to mean much more than how to handle time when sequencing data in a pipeline. Rather, we believe that these terms represent a collection of prejudices with respect to a number of design decisions. The categories of synchronous and asynchronous confines designers to one set of choices and we believe that these categories actually limit innovation.

For example, asynchronous designers spend most of their time tuning the bundling delays in the control path to model the delay found in the data path, while synchronous designers spend their time tuning the delays found in the data path to fit within a period provided by the clock. GasP [3] is a fairly recent " asynchronous" design style that broke this convention. The GasP design style advocates placing equal delays in all pipeline stages. GasP mixes design elements from the asynchronous design style, elasticity. and the synchronous design style, rigid periods. GasP reaps performance advantages by matching two design choices not previously paired.

But why not globally distribute some number of timing signals designed to bundle the delays of various data path stages with identical setup and hold conditions that are found in widely different locations? Here the asynchronous property of variable periods is coupled with the synchronous

property of global distribution. Could it be because global timing is in the scope of so called synchronous design, while bundled data is in the scope of asynchronous design and they shall not be mixed?

Categories are sometimes necessary to enable meaningful conversation but they can also hold innovation captive. This work attempts to tease apart some of the assumptions held in the terms synchrony and asynchrony. Once the different control factors are identified, such as as local versus global control, scheduling versus arbitrating, or matching the data path delays to control path delays or vice-versa, then each decision can be evaluated in isolation as opposed to as part of a package.

The authors in [1] argue that a favorable result of their work would be the fully asynchronous processor. They argue that incrementally introducing asynchronous techniques into the synchronous design flow will result in an increased comfort in using asynchronous design that might eventually yield a truly asynchronous design flow.

We find the potential implication of their work to be possibly far more revolutionary and impacting. We see their work as a first step in allowing great innovations as designers explore options and possibilities that our current categories forbid.

This work suggests that the 'best' design might be neither synchronous and asynchronous and that those terms are actually counter-productive. Rather we believe that the 'best' might be something that is not adequately described by the labels synchronous and asynchronous because it borrows heavily from both styles.

References

[1] I. Blunno, J. Cortadella, A. Kondratyev, L. Lavagno, K. Lwin, and C. Sotiriou. Handshake protocols for de-synchronization. In *10th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 149–158, 2004.

[2] S. Fairbanks and S. Moore. Analog micropipelines for high precision timing. In *10th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 41–50, 2004.

[3] I. Sutherland and S. Fairbanks. GasP: A minimal FIFO control. In *7th IEEE International Symposium on Asynchronous Circuits and Systems*, pages 46–53, Mar. 2001.