Challenges for the future

If asynchronous technology is to have a role in the future of the electronics industry there are several challenges which we, its principal proponents, must meet:

1. We must *demonstrate the technical merits* of asynchronous design. This requires that circuits of commercially relevant complexity are developed and that they show, convincingly, some advantage over comparable clocked designs. Building large-scale circuits is an expensive undertaking and beyond the capability of many (though not all) academic groups.
2. We must *develop efficient design methodologies* for asynchronous circuits. Even if asynchronous circuits can be shown to have advantages, this will not lead to commercial take-up if design costs are high. The Philips Tangram experience suggests that it is possible to build design tools for asynchronous circuits that are *more* efficient than those for clocked circuits, but much remains to be done.
3. We must *develop production testing strategies* for asynchronous circuits. Design for testability techniques such as built-in and scan-based test circuitry are beginning to appear, but again there is much to be done.

Underlying these front-line objectives are various supporting activities such as developing formal models of asynchronous systems, improving low-level circuits and inventing architectural techniques where existing techniques used in clocked circuits cannot readily be transferred.

In addition, those of us working in academia can begin to change the training of the next generation of designers to encourage them to be more open to asynchronous design.

Conclusions

The last decade has seen a remarkable growth in the interest in asynchronous techniques, despite the paucity of demonstrations of merit and the design and test difficulties. Some companies now appear ready to accept the risks for the potential benefits, and an atmosphere of expectancy has arisen which may not be sustainable for long. I think we will see asynchronous designs gain a significant commercial foothold over the next five years, attracting the level of resource which is required to make them competitive. If they fail, a new dark age will descend, but if they succeed we will have changed one of the fundamental precepts of our industry.

Whilst asynchronous design has perceived advantages over clocked design, there is a marked shortage of convincing demonstrations that these advantages apply to practical designs of commercial interest. We need considerably more concrete evidence that asynchronous circuits can be designed and tested at competitive cost and then have real benefits; without this, most of the world's design community will continue to view asynchronous techniques as academic curiosities pursued only for their qualities of 'truth and beauty'.

Unfortunately, developing competitive designs requires a level of resource which is rarely available to academics. We must therefore grasp the opportunity which is presented by the current level of industrial interest in our work and be prepared to face the commercial pressures which are associated with getting involved in product development. If we decline this opportunity, the next may be a long time coming!

References

1. van Berkel, K et al., "A Single-Rail Re-implementation of a DCC Error Detector Using a Generic Standard Cell Library", Proc. 2nd Working Conf. on Asynchronous Design Methodologies, London, May 30-31 1995, IEEE Computer Society Press.
2. Koche, A and Brunvand, E, "Testing Micropipelines", Proc. Intl. Symposium on Advanced Research in Asynchronous Circuits and Systems", Utah, November 1994.
3. Garside, J D, Temple, S and Mehra, R, "The AMULET2e Cache System", Proc. 2nd Intl. Symposium on Advanced Research in Asynchronous Circuits and Systems", Aizu, Japan, March 1996.

Obstacles to asynchronous design

The most frequently cited argument against asynchronous design is the difficulty of production testing asynchronous devices. Production test equipment is heavily committed to using a clock to regulate the application of stimuli and to time the sampling of responses from the device under test. In addition, asynchronous circuits tend to require logically redundant gates to remove hazards, and these are untestable by normal methods. Economic testability is an absolute requirement for volume production and much work remains to be done before we can be confident that asynchronous circuits can be made testable, but early results in the design for testability of asynchronous circuits [2] give cause for optimism that these problems are not insuperable.

Design tool support is another area of difficulty. Modern CAD tools are aimed at clocked circuits, and though it is clearly *possible* to design a clockless circuit using them, they could be a lot more helpful. For example, most synthesis tools will automatically remove logically redundant gates which have been carefully inserted to remove hazards.

Perhaps, though, the biggest obstacle to the widespread acceptance of asynchronous design techniques, always assuming that their merits can be demonstrated irrefutably, is the aversion of most of the electronic design community. Design engineers have been trained for the last twenty-five years to avoid asynchronous operation and most have a strong sense that though it was used in the distant past, it has been rendered obsolete by more recent developments in clocked design. We can expect these designers to be very resistant to suggestions that they should revert to design methodologies that were rejected a quarter of a century ago.

The state of the art

To see where asynchronous design has got to today we can look at one of the more complex asynchronous devices developed so far. The AMULET2e chip, designed at the University of Manchester, incorporates a self-timed 32-bit RISC processor core and a 4 Kbyte cache memory. The organisation of the device is shown in Figure 1. The processor core has organisational features such as register forwarding, out of order completion, data dependent addition timing and branch prediction. The cache [3] supports addressed word first refill, access behind line fetch and is 64-way associative. All on-chip activity is self-timed, but accesses to off-chip components are controlled by a reference delay built onto the chip. Typically, the reference delay will correspond to the off-chip RAM access time and other off-chip devices will be programmed as multiples of this delay. The memory interface provides support for byte-, halfword- (16-bit) and word-wide (32-bit) external devices, and offers a conventional bus interface to memory and peripheral parts.

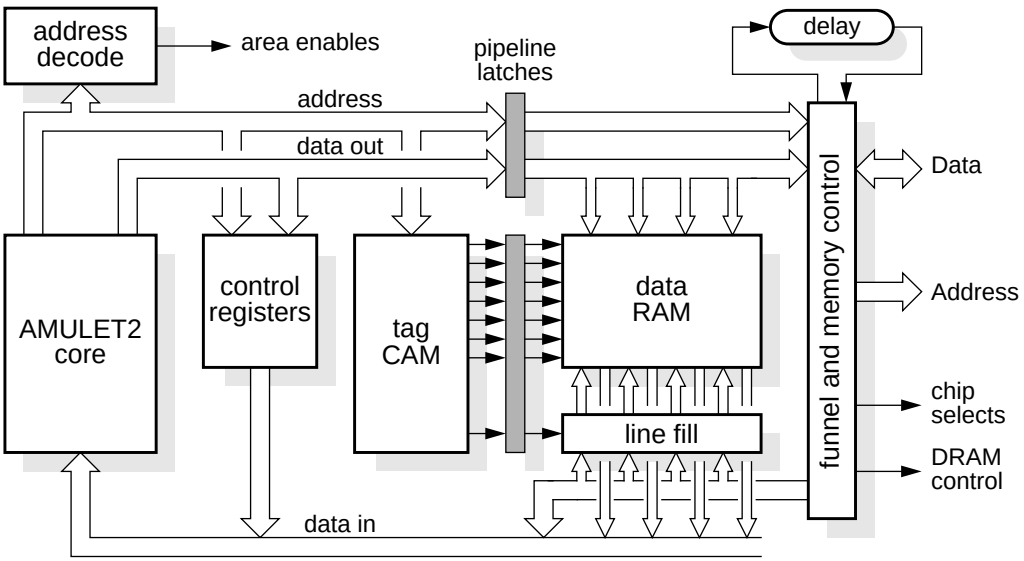

Figure 1 : *AMULET2e internal organisation.*

merits of asynchronous design to make our case irrefutable. Unfortunately, such large-scale demonstrators are few and far between, so we will for now have to be content with the rather diffuse arguments. These are, in order of importance, as follows:

1. *Power-efficiency.* Clocks beat away, causing power dissipation whether or not the circuit is doing useful work. Clock gating and dividing alleviates this problem, but only on a coarse granularity. An asynchronous circuit is inherently data driven and only uses power for useful work.
2. *Modularity.* A clock is a global variable; merging two design components which were developed with different clock requirements is very difficult. Asynchronous circuits have clean interfaces and avoid this problem.
3. *Performance*. The clock must be set so that the slowest logic stage can complete within a clock period under worst-case processing, data and environmental conditions. An asynchronous circuit delivers typical performance under typical conditions, which for a given logic function is twice the worst-case performance.
4. *Clock skew.* As clocks get faster, chips get bigger and tracks get thinner, it is increasingly hard to keep clock skew within tolerable limits. The engineering effort and silicon resource which goes into the clock distribution network on today's high-end processors illustrate the scale of the problem. Asynchronous circuits have no clock and therefore do not suffer from clock skew.
5. *Concurrency.* Most abstract representations of concurrent systems map far more naturally onto asynchronous circuits than they do onto clocked circuits.
6. *Metastability.* Any asynchronous input (which includes any input from the outside world) to a clocked circuit represents a source of unreliability, since there is always some residual probability that the synchronising circuit will fail. An asynchronous circuit can avoid incorrect operation by waiting until metastability has resolved. This potentially results in a failure to meet a real-time constraint, but in typical applications it should be possible to make the probability of failure very much lower than with a clocked system where reliability must often be traded-off against performance.

What evidence is there that these arguments hold up under close examination?

The Philips digital audio error detector chip [1] supports the power-efficiency claim; it demonstrates a factor five power reduction over clocked chips built on the same technology. The Tangram HDL and synthesis system used to design their asynchronous chips also supports the concurrency case; it is a far more elegant and straightforward synthesis route than anything available to designers of clocked chips.

The modularity problems with clocked circuits only arise if design components are ported at or below the register transfer level. If they are ported as behavioural descriptions, a synthesis tool *should* be able to generate a new register transfer structure to suit a new clock period. A perspective which is often applied is to look to a future where a billion transistors are available on a chip and to argue that the modularity case will then be compelling, but even here the problem can be removed by sufficiently powerful abstraction and synthesis tools.

The performance and clock-skew arguments are related, and are confused by the huge resource which is currently thrown at making clocked circuits perform well. Recent high-end processors demonstrate how clock problems can be engineered away. Their very high performance does appear in general to be incompatible with power-efficiency, and adding clock gating to improve power-efficiency compromises clock skew and thereby adversely affects performance, but a direct comparison between high-performance clocked and asynchronous circuits is difficult because of the imbalance in the resources available to the two methodologies.

Metastability should not be a problem in synchronous circuits in practice since a good clocked circuit designer can make the probability of failure very small indeed. Bad designers can get it very wrong, but they could probably do even worse with an asynchronous design!

Overall, the power-efficiency argument has the most immediate promise, with modularity and concurrency depending on tools support which is not yet widely available, and the hypothetical performance advantage has still to be demonstrated. Despite the rather weak case, though, there is growing industrial interest in asynchronous techniques and a feeling that asynchronous designs may be poised to claim a market share.

BREAKING STEP - THE RETURN OF ASYNCHRONOUS LOGIC

S B Furber

Introduction

Early computers employed a variety of techniques to control the flow of instructions and data through their various memory and processing units, but for the last quarter of a century one particular approach has come to dominate. Today, almost all digital design is based around the clock, where an external agent supplies a regular beat which all internal operations must follow. The term ‘clock’ is something of a misnomer, since this conjures up images of a dial on a wall to which reference may be made whenever one feels the need to know how long it is to lunchtime. An electronic clock is more akin to the conductor of an orchestra, providing the timing reference for every note that the system must play.

Under the dictatorship of the clock, the digital electronic civilisation has progressed apace. The rigid discipline imposed by the clock has enabled designers to make good use of the ever-increasing resources placed at their disposal. Computer Aided Design tools automatically generate VLSI circuits of a complexity which was unimaginable only a few years ago, and the progress in the cost-effectiveness of electronics puts computing resources into digital cellular telephone handsets that would be the envy of any 1970s minicomputer designer.

There is no denying that this progress owes a great deal to the simple design framework which is the result of basing all the control and communication mechanisms in a chip around a single clock signal. However, a digital electronic system does not *need* to be organised around a clock. Unlike an orchestra, where keeping all the playing units synchronised is essential to the correct functioning of the system as a whole, an integrated circuit need not be so regimented. A chip can be viewed as more like a production line, where adjacent units must cooperate when part-finished goods are passed from one to the other, but there is no requirement that this transaction happens at the same time as a similar transaction in a different part of the factory.

Chips which operate by local negotiation rather than under the control of a central clock are termed ‘asynchronous’ or ‘self-timed’. This is the digital electronic equivalent of political devolution and brings with it many of the same advantages and many of the same problems. Over the last ten years there has been a strong resurgence of interest in what to many designers, taught at university to avoid anything asynchronous, resembles electronic anarchy. This interest manifests itself in escalating numbers of publications on asynchronous design, in growing industrial interest and in specialised conferences, workshops and colloquia such as this. Are these just small, underground gatherings for those anarchists who have been rejected from the major party conferences? I think not, and in this paper I will attempt to explain why.

Why not just use a clock?

Since the clock has served the industry so well for so long, one must have good reasons to consider dispensing with its services. The rationale for the renewed interest in asynchronous design is, however, somewhat diffuse. There are several arguments which have been used to justify abandoning the known and trusted synchronous design methodology, none of which is wholly convincing on its own, but taken together they make a reasonable case for further research. Perhaps this is why much of the work is still carried out in academia, since we only require a good case for further research whereas our industrial colleagues require rather firmer ground to justify redirecting their development resources. What is really needed is convincing large-scale demonstrators of the

Steve Furber is with the Department of Computer Science, the University of Manchester (UK).