Scanning the Technology

Applications of Asynchronous Circuits

C. H. (KEES) VAN BERKEL, MEMBER, IEEE, MARK B. JOSEPHS, AND STEVEN M. NOWICK

A comparison with synchronous circuits suggests four opportunities for the application of asynchronous circuits: high performance, low power, improved noise and electromagnetic compatibility (EMC) properties, and a natural match with heterogeneous system timing. In this overview article each opportunity is reviewed in some detail, illustrated by examples, compared with synchronous alternatives, and accompanied by numerous pointers to the literature. Conditions for applying asynchronous circuit technology, such as the existence and availability of computer-aided design (CAD) tools, circuit libraries, and effective test approaches, are discussed briefly. Asynchronous circuits do offer advantages for many applications, and their design methods and tools are now starting to become mature.

Keywords—*Asynchronous circuits, asynchronous systems, CAD tools, heterogeneous timing, high performance, low power, low noise, testability.*

I. INTRODUCTION

Today, the semiconductor industry is giving serious consideration to the adoption of asynchronous circuit technology. Up until now, asynchronous circuits have been applied commercially only as small subcircuits, often as peripherals to controllers. Examples include counters, timers, wake-up circuits, arbiters, interrupt controllers, first-in first-out (FIFO), bus controllers, and interfaces (e.g., RS-232, SCSI, UART). The need for such asynchronous circuits stems largely from intrinsically asynchronous specifications.

During the last decade there has been a revival in research on asynchronous circuits [1], [2]. Emphasis is now shifting from asynchronous-in-the-small to asynchronous very large scale integration (VLSI) circuits and systems. Asynchronous VLSI is now progressing from a fashionable
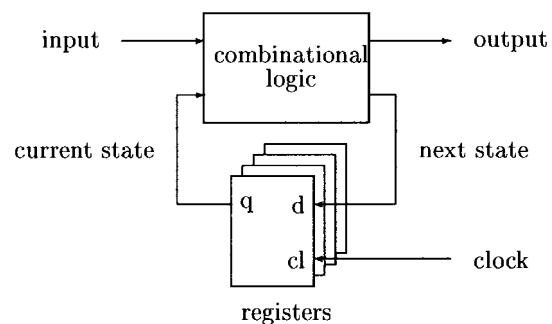
Fig. 1. Synchronous circuit.

academic research topic to a viable solution to a number of digital VLSI design challenges. A first, entirely asynchronous IC has recently appeared on the market (Section IV).

The added value of asynchronous circuit technology can best be understood by reviewing the key properties of synchronous circuits. A synchronous circuit in its simplest form is shown in Fig. 1. The current state of the circuit is stored in an array of registers. The next state is computed from the current state and inputs by a combinational logic circuit. When the clock signal makes a transition, say from low to high, the registers are enabled, and the next state is copied into the registers, thus becoming the current state. Each register bit generally comprises two latches, usually cascaded into a master–slave flipflop. In such a synchronous circuit, the following apply.

1) The longest path in the combinational logic determines the minimum clock period, whereas during a typical clock cycle the circuit may in fact become quiescent well before the next clock signal.

2) Each register dissipates energy during each clock cycle, regardless of the extent of the change in state. If dynamic logic is used, the combinational logic dissipates clock power during each clock cycle as well.

3) The clock modulates the overall supply current, causing peaks in power-supply noise and in electromag-

netic (EM) emission to occur at the clock frequency and higher harmonics thereof.

4) All functional submodules operate in lock step, a requirement that seems increasingly at odds with the growing significance of interconnect delays and the heterogeneous nature of systems-on-a-chip architecture.

The corresponding opportunities for application of asynchronous circuits are:

1) high performance (Section II);
2) low power dissipation (Section III);
3) low noise and low EM emission (Section IV);
4) a good match with heterogeneous system timing (Section V).

Section VI addresses design tools for asynchronous circuits, cell libraries, and testability issues. For an introduction to the modeling and design of asynchronous circuits the reader is referred to [3].

II. ASYNCHRONOUS FOR HIGH PERFORMANCE

In an asynchronous circuit the next computation step can start immediately after the previous step has completed: there is no need to wait for a transition of the clock signal. This leads, potentially, to a fundamental performance advantage for asynchronous circuits, an advantage that increases with the variability in delays associated with these computation steps. However, part of this advantage is canceled by the overhead required to detect the completion of a step. Furthermore, it may be difficult to translate local timing variability into a global system performance advantage. In this section we explore these data dependencies of delays and present a number of successful demonstrations of asynchronous performance advantages.

A. Data-Dependent Delays

The delay of the combinational logic circuit in Fig. 1 depends on the current state and the value of the primary inputs. The worst-case delay, plus some margin for flipflop delays and clock skew, is then a lower bound for the clock period of a synchronous circuit. Thus, the actual delay is always less (and sometimes much less) than the clock period.

A simple example is an N-bit ripple-carry adder (Fig. 2). The worst-case delay occurs when 1 is added to $2^N - 1$. Then the carry ripples from FA_1 to FA_N. In the best case there is no carry ripple at all, as, for example, when adding 1 to 0. Assuming random inputs, the average length of the longest carry-propagation chain is bounded by $\log_2 N$ [4].[1] For a 32-bit wide ripple-carry adder the average length is therefore five, but the clock period must be six times longer. On the other hand, the average length determines the average-case delay of an asynchronous ripple-carry adder, which we consider next.

In an asynchronous circuit this variation in delays can be exploited by detecting the actual completion of the addition.

[1] In practice this average may be somewhat larger, as shown in [5] for an arithmetic-logic unit (ALU) in a microprocessor.
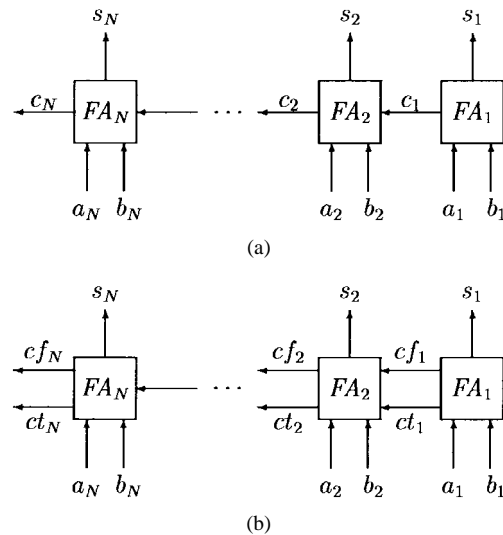

(a)

(b)

**Fig. 2.** (a) N-bit ripple-carry adder and (b) a self-timed version.

Most practical solutions use double-rail encoding of the carry signal [Fig. 2(b)]; the addition has completed when all internal carry-signals have been computed [5], [6]. That is, when each pair (cf_i, ct_i) has made a monotonous transition from $(0,0)$ to $(0,1)$ (carry = false) or to $(1,0)$ (carry = true). A variant of this scheme is applied in [7]. Double-rail encoding of the carry signal has also been applied to a carry bypass adder [8]. When inputs and outputs are double-rail encoded as well, the completion can be observed from the outputs of the adder [4], [9]. A quite different technique, with similar objectives, is speculative completion: so-called abort logic is used to select among a number of fixed delay-lines depending on the input values of a combinational circuit. In [10] this technique is introduced and applied to a carry look-ahead adder.

The asynchronous adders discussed above nicely demonstrate how data-dependent delays can be exploited to obtain a superior average-case delay compared to the fixed (worst-case) delay of the equivalent synchronous adder. This performance advantage is maximal for the ripple-carry adder and becomes more modest for adder organizations with carry acceleration, such as carry lookahead adders and carry select adders [11].

In some specific applications the large data-dependent variations in delays naturally lead to elegant and efficient asynchronous solutions. For example, Yun *et al.* [8] describe a differential equation solver based on adders and multipliers with superior average-case delays. Benes *et al.* [12] describe a high-speed software decompression engine for embedded processors. The engine exploits the large variations in delays so typical for Huffman decoders. For similar reasons Intel is investigating asynchronous instruction decoders [13]. In [14], performance benefits are pursued for microprogrammed control organizations.

B. Elastic Pipelines

In general, it is not easy to translate a local asynchronous advantage in average-case performance into a system-level performance advantage. Today's synchronous circuits are
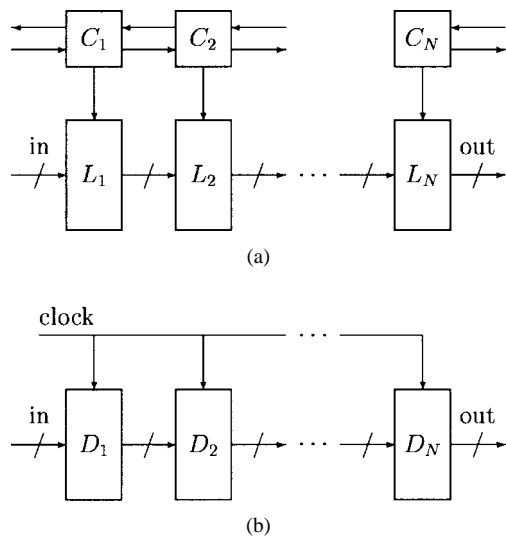
Fig. 3. (a) N-place micropipeline versus (b) an N-place clocked shift register.

heavily pipelined and retimed. Critical paths are nicely balanced and little room is left to obtain an asynchronous benefit. Moreover, an asynchronous benefit of this kind must be balanced against a possible overhead in completion signaling and asynchronous control.

The comparison of a so-called micropipeline and a clocked shift register is interesting in addressing performance issues. In its most basic form, a micropipeline [15] is an elastic FIFO buffer, constructed as a cascade of identical stages. Each stage consists of a latch L and a controller C, as in Fig. 3(a). The controller communicates exclusively with the controllers of the immediately preceding and succeeding stages by means of handshake signaling [4], and controls the state of the data latches (transparent or opaque). Between the request and the next acknowledge phase the corresponding data wires must be kept stable.

Maximum throughput of a micropipeline is obtained when it is half full, in which case alternatingly data are transferred from the even stages to the odd stages and from the odd to the even stages. When used as a high-throughput shift-register, the effective capacity of a $2N$-place micropipeline is thus reduced to N. One could say that the latches in stage $i + 1$ are used as slave latches to those in stage i.

Work at SUN Research (see [16]) shows that, with a carefully designed controller circuit, the throughput of a micropipeline can approach that of a synchronous shift register. The number of data latches of a $2N$-place micropipeline equals that of an N place shift register (a master latch plus a slave latch per bit). Hence the micropipeline solution is costlier, given the additional control circuitry C.

These additional costs, which can be relatively modest when compared to the costs of a wide data path, buy three interesting—and potentially useful—bonuses. Firstly, when not used at the maximum throughput, the $2N$-place micropipeline has a higher capacity, up to $2N$ places. (When completely full, a new vacancy created at the

FIFO's output will crawl back to the input). Secondly, the latency per stage is only a small fraction of the cycle time. For example, when a micropipeline is empty, data fall through the (then transparent) latches at a rate of a few gate delays per stage. Thirdly, there is the elasticity of an asynchronous FIFO. As a result, it can (for example) be used when connected to functional units with variable computation times, such as the adder discussed above. Also, micropipeline-like buffers can be used to interface between a data producer and a data consumer operating at different speeds. However, this elasticity also makes it harder in general to analyze the performance of asynchronous circuits, as is done in [17].

The self-timed dividers of [18] and [19] and the AMULET microprocessor [20] are based on micropipelines. Connecting two micropipelines running in opposite directions and connecting them stage-by-stage in fact results in an innovative concept for a microprocessor architecture [21]. A group at California Institute of Technology (Caltech) designed an asynchronous version of the MIPS R3000 [22]. They expect high performance in part by fine-grained pipelining techniques; the number of instructions in the pipeline is dynamic, depending on specific instruction sequences.

C. Quantifying Circuit Performance

Quantifying the performance of synchronous and asynchronous circuits can be tricky [23] and is often a source of confusion. A clocked circuit is usually guaranteed to run at a specified maximum frequency over some range in ambient temperature and supply voltage. Furthermore, there is also a considerable variation between the worst-case and the best-case CMOS process corners. When combined, this results in a safety margin or "derating factor" of about a factor two. This means that, under typical conditions, many chips could run at about twice the clock frequency specified.

An asynchronous circuit, in contrast, when not delayed by its environment, runs as fast as it goes. It slows down when heated, or when the supply voltage drops. Furthermore, by measuring its performance, one in effect also measures the quality of the CMOS processing. Hence, measured asynchronous performance will vary from one fabrication run to another. Note that when the circuit's specification contains strict requirements on throughputs or response times, the asynchronous performance is subject to the same derating factor as used for synchronous circuits. Beware!

III. ASYNCHRONOUS FOR LOW POWER

A quiescent circuit only consumes a leakage current. For most CMOS circuits this leakage current is negligible compared to the dynamic current for that circuit in an active mode. A synchronous circuit is either quiescent (i.e., the clock is turned off) or active entirely (i.e., clock on). An asynchronous circuit, in contrast, only consumes energy when and where active. Any subcircuit is quiescent until activated. After completion of its task, it returns to a quiescent, almost nondissipating state until a next activation.
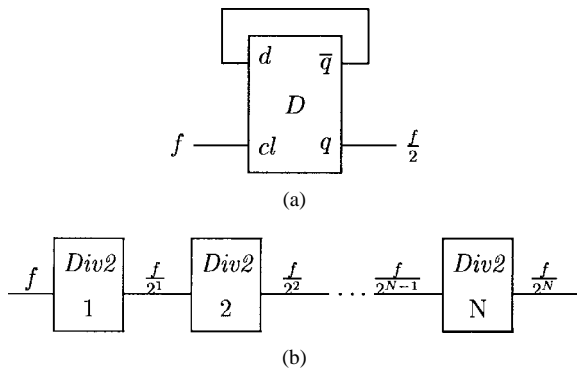
Fig. 4. (a) Divide-by-2 element and (b) divide-by-2^N circuit.

In this section the potential for low power consumption of asynchronous circuits is reviewed, including a number of successful demonstrations. However, it is not obvious to what extent this advantage is fundamentally asynchronous. Synchronous techniques such as clock gating may achieve similar benefits, but they have their limitations.

A. Dissipating When and Where Active

The classic example of a low-power asynchronous circuit is a frequency divider. A D-flipflop with its inverted output fed back to its input divides an incoming (clock) frequency by two [Fig. 4(a)]. A cascade of N such divide-by-two elements [Fig. 4(b)] divides the incoming frequency by 2^N. The second element runs at only half the rate of the first one and hence dissipates only half the power; the third one dissipates only a quarter, and so on. Hence, the entire asynchronous cascade consumes, over a given period of time, slightly less than twice the power of its head element, independent of N. That is, a fixed power dissipation is obtained. In contrast, a similar synchronous divider would dissipate in proportion to N. A cascade of 15 such divide-by-two elements is used in watches to convert a 32-kHz crystal clock down to a 1-Hz clock.

Similar asynchronous schemes with similar advantages have been applied to modulo-N counters [24], [25]. A counter with loadable N with these properties is presented in this Special Issue as part of a low-power asynchronous pager circuit [26].

One way of looking at this low-power property is that only active modules dissipate power. Modules that are not active resort automatically and instantaneously to a standby mode; this is true at arbitrary granularity both in time and in function. A number of examples of how asynchronous techniques help to reduce power consumption are explored in [27].

The potential of asynchronous for low-power depends on the application. For example, in a digital filter where the clock rate equals the data rate, all flipflops and all combinational circuits are active during each clock cycle. Then little or nothing can be gained by implementing the filter as an asynchronous circuit. However, in many digital-signal processing functions the clock rate exceeds the data (signal) rate by a large factor, sometimes by several orders

of magnitude.[2] In such circuits, only a small fraction of registers change state during a clock cycle. Furthermore, this fraction may be highly data dependent.

One application for which asynchronous circuits can save power is Reed–Solomon error correctors operating at audio rates [28], as demonstrated at Philips Research Laboratories. In [29], two different asynchronous realizations of this decoder (single-rail and double-rail) are compared with a synchronous (product) version. The single rail was clearly superior and consumed five times less power than the synchronous version.

A second example is the infrared communications receiver IC designed at Hewlett-Packard/Stanford [30]. The receiver IC draws only leakage current while waiting for incoming data but can start up as soon as a signal arrives so that it loses no data. Also, most modules operate well below the maximum frequency of operation.

The filter bank for a digital hearing aid was the subject of another successful demonstration, this time by the Technical University of Denmark in cooperation with Oticon, Inc. They reimplemented an existing filter bank as a fully asynchronous circuit [7], [31]. The result is a factor five less power consumption.

A fourth application is a pager in which several power-hungry subcircuits were redesigned as asynchronous circuits, as shown later in this Special Issue [26].

B. Low-Power Processors

Several groups have taken up the gauntlet to explore and exploit this low-power potential for full-fledged programmable processors. In such processors, circuit activity may vary considerably depending on the particular instruction (sequences) and on the occurrence of exceptions. The following are the promising results on four such processors are described: a Reduced Instruction Set Computer; a multimedia processor; a microcontroller; and a programmable digital signal processor.

The University of Manchester designed the AMULET2e, an embedded system chip incorporating a 32-bit ARM-compatible asynchronous core, a cache, and several other system functions [20], [32], [33]. Quite significant is that the synchronous versions of the ARM are already well known for their low power consumption. Accordingly, the reduction in power per MIPS is modest. However, power consumption in the asynchronous idle mode is a different story. The absence of a high-frequency oscillator and phase locked loop (PLL) offers a quite unique combination of two features: μW power consumption and instant response to an external interrupt. There is no need to stop an oscillator and a PLL and to deal with their slow restart and stabilization.

A collaborative effort of Sharp Corporation and the Universities of Osaka and Kochi resulted in a self-timed data-driven multimedia processor [34]–[36]. The processor comprises eight programmable, data-driven processing elements, connected by an elastic router. Target applica-

---

[2]The clock frequency is chosen that high to accommodate sequential algorithms that share resources over subsequent computation steps.

(a)

(b)
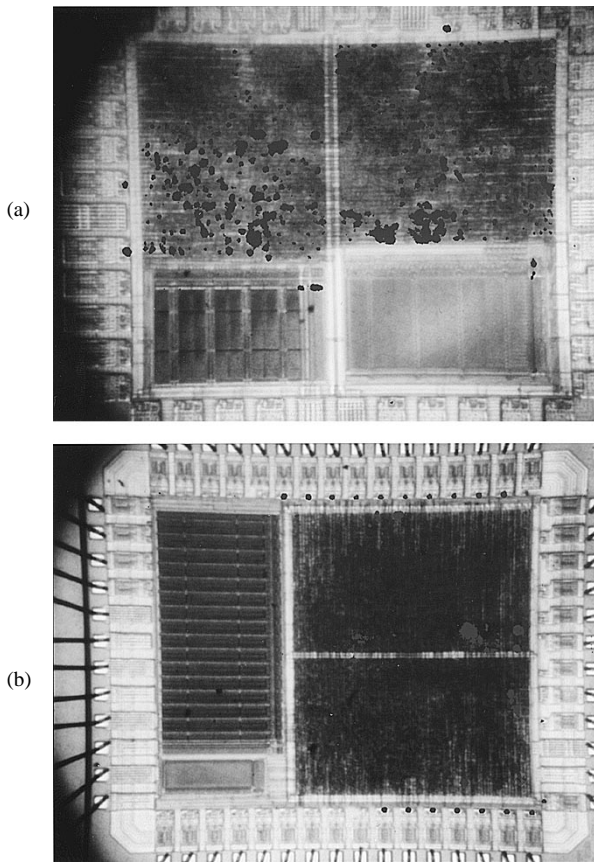
Fig. 5. The photographs show two versions of the 805C1 microcontroller. Both IC's have been realized in the same 0.5-μm CMOS process and run on the same test program at the same performance and under the same operating conditions. The pictures were made using photon-emission microscopy. The spots show emitted light and form an indication of the power dissipated by the two IC's. The (a) synchronous version shows much—and widespread—activity; the (b) asynchronous version clearly shows less—and more local—activity. (Photographs kindly provided by the Test and Diagnostics Group, Philips Research Labs.)

tions include future digital television receivers. It has an impressive peak performance of 8600 MOPS with a power consumption below 1 W (0.25-μm CMOS @ 2.5 V). The power consumption of the individual processing elements scales with their loads.

Philips Semiconductors together with Philips Research redesigned the 80C51 microcontroller. The asynchronous version [37] consumes about four times less power than its synchronous counterpart In Fig. 5, the nature of the differences in power consumption of the two IC's is visualized using photo-emission microscopy. This asynchronous version of the 80C51 microcontroller also has superior EM compatibility (EMC) properties (see also Section IV).

Finally, Cogency redesigned a programmable Digital Signal Processor [38], consuming about half the power of its synchronous counterpart.

Most asynchronous circuits have the property that their performance scales continuously with the supply voltage over a wide range. In a number of cases, correct circuit operation has been demonstrated from subthreshold to oxide-breakdown supply voltages! This makes asyn-

chronous circuits very suitable for adaptive scaling of the supply voltage [39], [40]. Such schemes can also work for synchronous circuits, but then the clock frequencies must scale simultaneously.

The number of published asynchronous low-power circuits is growing rapidly. In increasingly many cases there are careful comparisons with existing synchronous solutions. However, these comparisons are not always against an optimal low-power synchronous circuit. Moreover, there is also considerable progress in reducing the power in clocked circuits, for example by introducing multiple clocks or by locally gating clocks. Clock gating, also known as conditional clocking, has recently been applied to advanced high-performance microprocessors [41], [42]. Synthesis of clock-gating circuitry can to some extent be automated [43], [44]. Although the results are sometimes impressive (a four-fold reduction of the power consumption in a floating point unit [41]), it is also noted that clock gating complicates functional validation and timing verification, and that the extra gate used to qualify the clock can potentially introduce critical skews.

In summary, asynchronous operation by itself does not imply low power [45], but it often suggests low-power opportunities based on the observation that asynchronous circuits only consume power when and where active.

## IV. ASYNCHRONOUS FOR LOW NOISE AND LOW EMISSION

Subcircuits of a system may interact in unintended and often subtle ways. For example, a digital subcircuit generates voltage noise on the power-supply lines or induces currents in the silicon substrate. This noise may affect the performance of an analog-to-digital converter connected so as to draw power from the same source or that is integrated on the same substrate. Another example is that of a digital subcircuit that emits EM radiation at its clock frequency (and the higher harmonic frequencies), and a radio receiver subcircuit that mistakes this radiation for a radio signal.

Due to the absence of a clock, asynchronous circuits may have better noise and EMC properties [28] than synchronous circuits. This advantage can be appreciated by analyzing the supply current of a clocked circuit in both the time and frequency domains.

Circuit activity of a clocked circuit is usually maximal shortly after the productive clock edge. It gradually fades away and the circuit must become totally quiescent before the next productive clock edge. Viewed differently, the clock signal modulates the supply current as depicted schematically in Fig. 6(a). Due to parasitic resistance and inductance in the on-chip and off-chip supply wiring, this causes noise on the on-chip power and ground lines. Local drops in the supply voltage have impact on performance, and excessive noise may even impact circuit reliability.

Another problem becomes manifest when the supply current is analyzed in the frequency domain. The supply current of Fig. 6(a) can be rewritten as

$$\frac{1}{2} - \frac{1}{\pi}\left(\sin \omega_c t + \frac{1}{2}\sin 2\omega_c t + \frac{1}{3}\sin 3\omega_c t + \cdots\right)$$
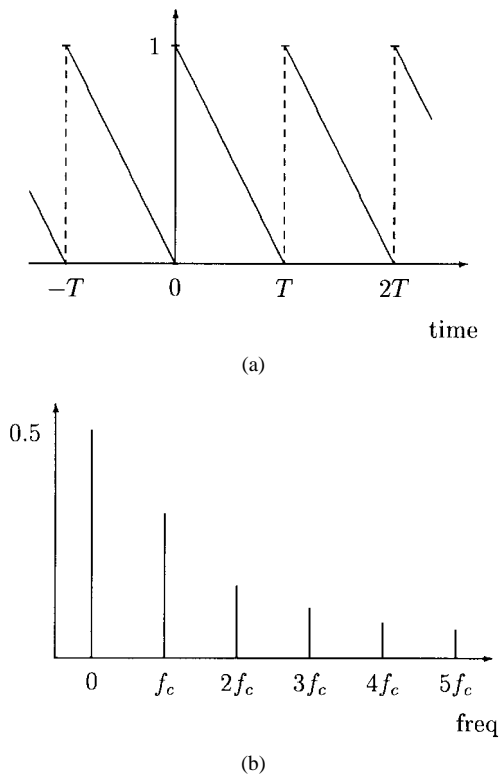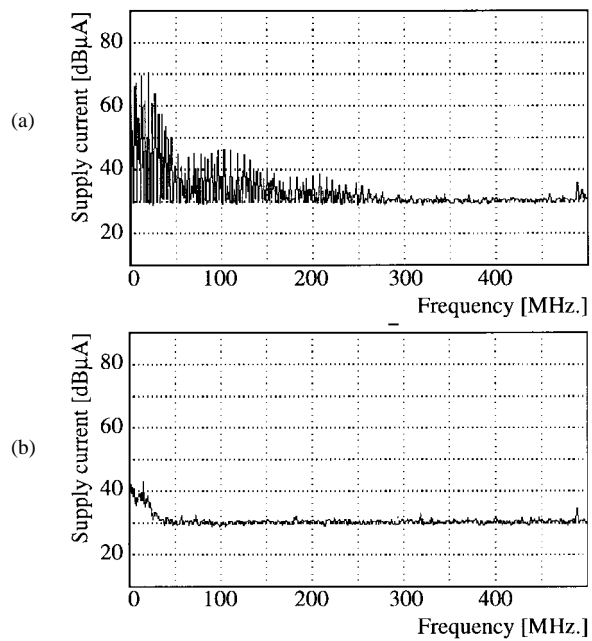
Fig. 6. Approximation of the supply current I_{dd} of a clocked circuit (a) in the time domain and (b) in the frequency domain.

Fig. 7. Frequency spectra of the two 80C51 microcontrollers of Fig. 5. The two controllers run the same test program at the same performance. The spectrum of (a) the synchronous version shows a long series of peaks at the harmonics of the 3.6-MHz clock frequency. The spectrum of the (b) asynchronous version does not show these peaks and clearly contains less energy. (Graphs kindly provided by Philips Semiconductors, Zürich, Switzerland.)

with $\omega_c = 2\pi f_c$. Hence, the clock causes a discrete contribution to the frequency spectrum of the supply current. The amplitude of this spectrum is shown in Fig. 6(b). The coefficients at nf_c with $(n > 1)$ denote the so-called harmonic amplitudes.[3] Voltage drops across parasitic inductances as well as the emitted EM fields are proportional to the first time derivative of the supply current. Hence, the amplitudes of the higher harmonics of the EM emission hardly drop at all below a few GHz. These higher harmonics may interfere with antennas and sensitive analogue circuits, including radio circuits from FM (100 MHz) to portable phones (1–2 GHz). For example, a tuner may mistake a particular harmonic of a clock frequency for a local FM station. The effects of interference can be reduced by means of (costly) shielding measures.

Note that harmonics are generally distinct sharp peaks because of the high quality of applied oscillators. In practice the spectrum is continuous and time varying due to nonperiodic components in the supply currents.

The frequency spectrum of the supply current of an asynchronous circuit obviously does not exhibit peaks at clock frequencies and multiples thereof. There may be spikes, but they tend to fade away when a longer integration interval is taken. Even periodic circuit behavior is less harmful; data-dependent delays invariably cause jitter, and

even a modest amount of jitter causes a rapid fall off of the harmonics.

Fig. 7 shows the frequency spectra of the supply current of both the synchronous version [Fig. 7(a)] and the asynchronous version [37] [Fig. 7(b)] of the 80C51 microcontroller. The synchronous version clearly shows a series of harmonics of the clock frequency (here about 3.6 MHz), dominating the spectrum up to about 300 MHz. In a pager product, the harmonics generated by such a synchronous microcontroller could interfere with the very sensitive analog radio circuits. In contrast, the low emission levels of the asynchronous 80C51 makes it possible to have the microcontroller active during reception of a paging message. For this reason Philips Semiconductors has developed a family of entirely asynchronous Pager Baseband Controller IC's, based on the cited asynchronous 80C51.[4] This IC has been put on the market successfully.[5]

Note that reducing power consumption generally also reduces the energy content of these spectra. An example of a measured EM emission spectrum of an asynchronous microprocessor can be found in [20].

The above suggests that asynchronous circuits often may be superior in EMC. In some specific cases, however, the opposite may be true. For example, a synchronous circuit is known to be quiescent just before the productive clock

---

[3] In contrast to Fig. 6(a) the circuit is generally quiescent well before the productive clock edge, often as early as halfway into the clock period to accommodate derating. In the frequency domain this manifests itself by relatively smaller coefficients for the even harmonics.

[4] The IC reported in [26] is not a member of this family of IC's.

[5] Personal communication of the authors with Francisco Ferrer of Philips Semiconductors, Zürich, Switzerland.
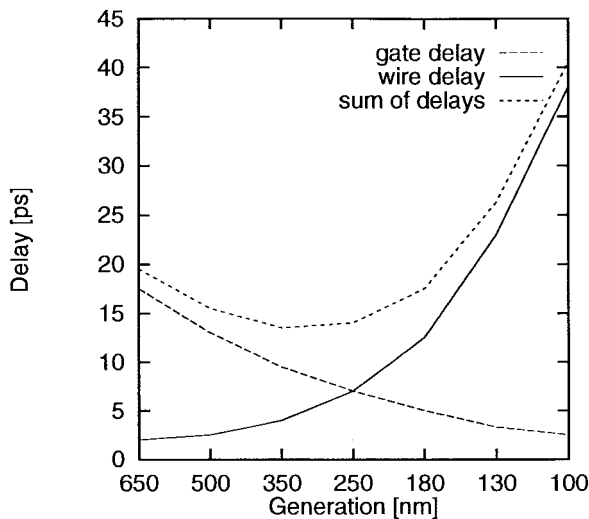
Fig. 8. Calculated gate and interconnect delay versus technology generation. The interconnect is an aluminum wire of 43-μm length and 0.8-μm thickness. Adapted from [46].

edge, providing an excellent moment to sample an analog signal for A-to-D conversion.

The above analysis is highly simplified, and it ignores for example the noise and EM emission associated with the simultaneous driving of a number of output loads. Still, EMC is becoming an increasingly important issue in electronic system design with respect to safety, reliability, and system costs. The relative EMC merits of asynchronous circuits clearly deserves more research attention.

V. HETEROGENEOUS TIMING

There are two ongoing trends that affect the timing of a system on a chip: the relative increase of interconnect delays versus gate delays and the rapid growth of design reuse. Their combined effect results in an increasingly heterogeneous organization of system-on-a-chip timing.

According to Fig. 8, gate delays rapidly decrease with each technology generation. By contrast, the delay of a piece of interconnect of fixed modest length increases, soon leading to a dominance of interconnect delay over gate delay [47], [46]. The introduction of additional interconnect layers and new materials (copper and low dielectric constant insulators) may slow down this trend somewhat. Nevertheless, new circuits and architectures are required to circumvent these parasitic limitations. For example, across-chip communication may no longer fit within a single clock period of a processor core. Accordingly, the 1997 edition of the SIA roadmap predicts a divergence between "on-chip local clock frequencies" and "across-chip clock frequencies." The former outperforms the latter by a factor that gradually grows to three. The increasing role of interconnect parasitics also makes it less and less practical to distribute a single high-frequency clock across the entire IC.

The same roadmap also predicts that the fraction of the die area covered by reusing existing circuit designs will increase to as much as 90%. Moreover, complex systems-on-a-chip will accommodate blocks from different design houses (DSP cores, microcontrollers, a variety of memory blocks, MPEG decoders, modems, etc.) and blocks that must conform to standardized off-chip interfaces. The result will be a plethora of (local) clock frequencies and circuit-level timing conventions.

Heterogeneous system timing will offer considerable design challenge for system-level interconnect, including buses, FIFO's, switch matrices, routers, and multiport memories. Asynchrony makes it easier to deal with interconnecting a variety of different clock frequencies, without worrying about synchronization problems, differences in clock phases and frequencies, and clock skew [48]. Hence, new opportunities will arise for asynchronous interconnect structures and protocols [49]–[51]. Once asynchronous on-chip interconnect structures are accepted, the threshold to introduce asynchronous clients to these interconnects is lowered as well. Also, mixed synchronous-asynchronous circuits hold promise [26], [52], [53].

VI. TOOLS, LIBRARIES, AND TESTABILITY

For a wider acceptance and application of asynchronous circuit technology, it is critical that tools for the synthesis and verification of asynchronous circuits become available. In order to make asynchronous circuits more competitive in cost, it would also be beneficial to extend standard-cell and gate-array libraries with a number of typical asynchronous circuits. Furthermore, it is absolutely essential that effective test approaches and tools be developed such that asynchronous circuits can be tested according to the same quality standards as synchronous circuits.

A. Tools

Fortunately, many of the conventional tools such as simulators, placement and routing tools, and delay extractors are also very effective in supporting the design of asynchronous circuits. This may even apply to logic synthesis tools and timing-analysis tools for certain asynchronous design styles. By relying on a systematic design method, an entire microprocessor has been designed successfully, without dedicated asynchronous design tools [32].

Nevertheless, the manual design of asynchronous control circuits is difficult and error prone. Hazards are easily introduced, and often very hard to recognize. A tool that verifies whether a given control circuit exhibits the specified behavior, and therefore is hazard free, is especially useful [54]. In many practical circuits, the absence of hazards depends on assumptions on the relative delays of the various circuit elements. Examples of tools to check these timing assumptions are [55] and [56].

Synthesis of asynchronous control circuits is becoming a mature technology. Most computer-aided design (CAD) tools synthesize these circuits from one of two specification styles: burst-mode, a Mealy-type state machine description [57]–[59], [52]; and signal transition graphs, or STG's, a Petri-net based formalism [60]–[63]. The synthesized

circuits are hazard free but differ in assumption on delays. STG-based circuits are often speed independent, allowing robust operation with weak assumptions about delays. In contrast, burst-mode circuits typically must meet somewhat stricter timing constraints (fundamental-mode assumption [64]) which in practice are often easily met, but allow greater flexibility in the synthesis path. A number of tools have been developed for both burst-mode [57], [59], [65]–[68] and STG [60], [69]–[73] synthesis; these have been applied to a number of real-world designs [8], [13], [73]–[76]. An alternative approach has also been proposed, called timed circuits [77], which incorporates user-specified timing information to optimize the circuits. Compiling asynchronous circuits from higher level programming languages has been extensively explored in [78]–[80] and these methods have been used to produce about two dozen functional IC's, including [26], [29], [37], [81].

It would be an error to apply a regular "synchronous" technology mapper [82] to asynchronous circuits, because the mapper may introduce hazards in an otherwise hazard-free circuit. Technology mapping for asynchronous circuits is addressed in, amongst others, [13] and [83]–[85].

Performance analysis of synchronous circuits can cleanly be separated into two tasks: the measurement of the length of the critical path in the combinational logic, and the counting of the number of clock ticks required for a given task. In asynchronous circuits, however, delays are often data dependent and are not rounded to an integer number of clock periods. Therefore, timing and performance analysis of asynchronous circuits clearly requires different techniques and are the subject of [17] and [86].

The academic research community has been very active in developing CAD tools, and many tools that support the design of asynchronous circuits are available on the Internet [87]. So far, EDA vendors have monitored these developments, but they have not yet included such tools in their product portfolios.

B. Layout Libraries

Asynchronous circuits can be implemented using standard cells and gate arrays without major problems. Although common standard-cell libraries have been optimized for the realization of synchronous circuits, they turn out to be adequate for realizing asynchronous circuits as well [29]. Nevertheless, circuit-area reductions of, say, 10% can often be achieved by optimizing common asynchronous cells such as latches, various C-elements, and mutual-exclusion elements.

C. Testability

A synchronous circuit organized according to Fig. 1 has two features that simplify testing dramatically: it can be stopped during each clock cycle, and it is both simple and cheap to include a scan-chain through all flipflops. Asynchronous circuits exhibit more autonomy, and given the large variety of isolated latch elements it is harder and more costly to connect them into scan chains. Accordingly, test-ing asynchronous circuits is harder, and the cost overhead for design-for-testability measures is higher. Nevertheless, testing specific classes of asynchronous circuits appears feasible, and progress is being made to reduce testability costs. See, for example, [88] and [89].

VII. CONCLUSION

In this article we have reviewed four opportunities for the application of asynchronous circuits. Our findings are summarized below.

First, by avoiding the wait until the next clock edge, asynchronous circuits exhibit average-case performance rather than worst-case performance. Furthermore, asynchronous circuits avoid the overheads and problems associated with distributing clock signals. These potential advantages must, however, be balanced against some delay overheads introduced by asynchronous control and completion detection. This balance will weigh in favor of the asynchronous alternative when the delays of the combinational logic are highly data dependent, or when for a subfunction the optimal clock frequency is simply not available on chip. Indirectly, the elasticity of asynchronous pipelines may offer other advantages, such as free buffer capacity.

Second, by enabling latches and flipflops only when their state must be updated, asynchronous circuits consume power only when and where active. This type of power savings can also be realized by gating and pausing local clocks in synchronous circuits, however, at the expense of some additional clock skew. Also, asynchronous circuit technology naturally provides power savings at arbitrary levels of granularity and has methods and tools to guarantee absence of hazards on the latch enable wires.

Third, asynchronous circuits do not emit radiation at the clock frequency and harmonic frequencies thereof. This opportunity for asynchronous circuits is quite fundamental. Still, more study and measurements are required to quantify its significance.

Fourth, the increasing significance of parasitic interconnect resistance, the increasing reuse of building blocks, and the integration of entire systems on a few chips inevitably leads to a complex and heterogeneous on-chip timing organization. It will be less and less practical to clock all building blocks on a single, high-frequency, global clock. Here, the application of asynchronous circuits and subcircuits probably holds most promise. Their use may well turn out to be unavoidable.

As becomes apparent from the above discussion, for each opportunity one may argue that there exists also a synchronous alternative. Indeed, in many cases these alternatives may have the desired effect to some extent. In other cases asynchronous solutions may be more effective, cheaper, easier to realize, or simply more elegant.

Of course, an asynchronous (sub)circuit will only make it to the marketplace when there is no synchronous alternative or when the asynchronous solution has a clear and substantial advantage. The improved EMC properties of the asynchronous pager IC of Philips is a good example of such

an advantage. Given the rich experience developed over the last decade by a very active and productive asynchronous research community, we predict that asynchronous circuit technology will be applied more and more often. Based on this review we do not expect an asynchronous revolution, but rather a steady evolution.

REFERENCES

[1] A. Peeters, "The 'asynchronous' bibliography (BIBTEX) database file async.bib." [Online.] Available FTP: ftp://ftp.win.tue.nl/pub/tex/async. bib.Z, corresponding e-mail address: async-bib@win.tue.nl.

[2] J. Garside, *The Asynchronous Logic Homepage*. [Online.] Available WWW: http://www.cs. man.ac.uk/amulet/async/.

[3] M. B. Josephs, S. M. Nowick, and C. H. (Kees) van Berkel, "Modeling and design of asynchronous circuits," this issue, pp. 234-242.

[4] C. L. Seitz, "System timing," in *Introduction to VLSI Systems*, C. A. Mead and L. A. Conway, Eds. Reading, MA: Addison-Wesley, 1980, ch. 7.

[5] J. D. Garside, "A CMOS VLSI implementation of an asynchronous ALU," in *Asynchronous Design Methodologies*, vol. A-28 of *IFIP Transactions*, S. Furber and M. Edwards, Eds. London: Elsevier Science, 1993, pp. 181–207.

[6] B. Gilchrist, J. H. Pomerene, and S. Y. Wong, "Fast carry logic for digital computers," *IRE Trans. Electron. Comput.*, vol. EC-4, no. 4, pp. 133–136, Dec. 1955.

[7] L. S. Nielsen and J. Sparsø, "Designing asynchronous circuits for low power: An IFIR filter bank for a digital hearing aid," this issue, pp. 268-281.

[8] K. Y. Yun, A. E. Dooply, J. Arceo, P. A. Beerel, and V. Vakilotojar, "The design and verification of a high-performance low-control-overhead asynchronous differential equation solver," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Apr. 1997, pp. 140–153.

[9] A. J. Martin, "Asynchronous datapaths and the design of an asynchronous adder," *Formal Methods in Syst. Design*, vol. 1, no. 1, pp. 119–137, July 1992.

[10] S. M. Nowick, K. Y. Yun, and P. A. Beerel, "Speculative completion for the design of high-performance asynchronous dynamic adders," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Apr. 1997.

[11] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design; A System Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1993.

[12] M. Benes, A. Wolfe, and S. M. Nowick, "A high-speed asynchronous decompression circuit for embedded processors," *Adv. Res. VLSI*, pp. 219–236, Sept. 1997.

[13] W. Chou, P. A. Beerel, R. Ginosar, R. Kol, C. J. Myers, S. Rotem, K. Stevens, and K. Y. Yun, "Average-case optimized technology mapping of one-hot domino circuits," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 80–91.

[14] H. Jacobson and G. Gopalakrishnan, "Application-specific programmable control for high-performance asynchronous circuits," this issue, pp. 319-331.

[15] I. E. Sutherland, "Micropipelines," *Commun. ACM*, vol. 32, no. 6, pp. 720–738, June 1989.

[16] C. E. Molnar, I. W. Jones, W. S. Coates, J. K. Lexau, S. M. Fairbanks, and I. E. Sutherland, "Two FIFO ring performance experiments," this issue, pp. 297-316.

[17] J. Ebergen and R. Berks, "Response-time properties of linear asynchronous pipelines," this issue, pp. 308-318.

[18] T. E. Williams and M. A. Horowitz, "A zero-overhead self-timed 160 ns 54 b CMOS divider," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1651–1661, Nov. 1991.

[19] G. Matsubara and N. Ide, "A low power zero-overhead self-timed division and square root unit combining a single-rail static circuit with a dual-rail dynamic circuit," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Apr. 1997, pp. 198–209.

[20] S. B. Furber, J. D. Garside, P. Riocreux, and S. Temple, P. Day, J. Liu, and N. C. Paver, "AMULET2e: An asynchronous embedded controller," this issue, pp. 243-256.

[21] R. F. Sproull, I. E. Sutherland, and C. E. Molnar, "The counterflow pipeline processor architecture," *IEEE Design Test Comput.*, vol. 11, pp. 48–59, Fall 1994.

[22] A. J. Martin, A. Lines, R. Manohar, M. Nystroem, P. Penzes, R. Southworth, and U. Cummings, "The design of an asynchronous MIPS R3000 microprocessor," in *Adv. Res. VLSI*, pp. 164–181, Sept. 1997.

[23] M. E. Dean, "STRiP: A self-timed RISC processor architecture," Ph.D. dissertation, Stanford University, Stanford, CA, 1992.

[24] K. van Berkel, "VLSI programming of a modulo-N counter with constant response time and constant power," in *Asynchronous Design Methodologies*, vol. A-28 of *IFIP Transactions*, S. Furber and M. Edwards, Eds. London: Elsevier Science, pp. 1–11, 1993.

[25] J. L. W. Kessels, "Calculational derivation of a counter with bounded response time and bounded power dissipation," *Distrib. Comput.*, vol. 8, no. 3, pp. 143–149, 1995.

[26] J. Kessels and P. Marston, "Designing asynchronous standby circuits for a low-power pager," this issue, pp. 257-267.

[27] K. van Berkel and M. Rem, "VLSI programming of asynchronous circuits for low power," in *Asynchronous Digital Circuit Design* (Workshops in Computing), G. Birtwistle and A. Davis, Eds. New York: Springer-Verlag, 1995, pp. 152–210.

[28] K. van Berkel, R. Burgess, J. Kessels, A. Peeters, M. Roncken, and F. Schalij, "Asynchronous circuits for low power: A DCC error corrector," *IEEE Design Test Comput.*, vol. 11, pp. 22–32, Summer 1994.

[29] K. van Berkel, R. Burgess, J. Kessels, A. Peeters, M. Roncken, F. Schalij, and R. van de Wiel, "A single-rail re-implementation of a DCC error detector using a generic standard-cell library," in *Asynchronous Design Methodologies*. Los Alamitos, CA: IEEE Computer Society Press, 1995, pp. 72–79.

[30] A. Marshall, B. Coates, and P. Siegel, "Designing an asynchronous communications chip," *IEEE Design & Test of Comput.*, vol. 11, pp. 8–21, 1994.

[31] L. S. Nielsen and J. Sparsø, "An 85 μW asynchronous filterbank for a digital hearing aid," presented at International Solid-State Circuits Conf., Feb. 1998.

[32] S. Furber, "Computing without clocks: Micropipelining the ARM processor," in *Asynchronous Digital Circuit Design* (Workshops in Computing), G. Birtwistle and A. Davis, Eds. New York: Springer-Verlag, 1995, pp. 211–262.

[33] J. D. Garside, S. Temple, and R. Mehra, "The AMULET2e cache systems," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Mar. 1996, pp. 208–217.

[34] S. Komori, H. Takata, T. Tamura, F. Asai, T. Ohno, O. Tomisawa, T. Yamasaki, K. Shima, H. Nishikawa, and H. Terada, "A 40-MFLOPS 32-bit floating-point processor with elastic pipeline scheme," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1341–1347, Oct. 1989.

[35] H. Terada, M. Iwata, S. Miyata, and S. Komori, "Superpipelined dynamic data-driven VLSI processors," in *Advanced Topics in Dataflow Computing and Multithreading*. Los Alamitos, CA; IEEE Computer Society Press, 1995, pp. 75–85.

[36] H. Terada, S. Miyata, and M. Iwata, "DDMP's: Self-timed super-pipelined data-driven multimedia processors," this issue, pp. 282-296.

[37] H. van Gageldonk, D. Baumann, K. van Berkel, D. Gloor, A. Peeters, and G. Stegmann, "An asynchronous low-power 80c51 microcontroller," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 96–107.

[38] N. C. Paver, P. Day, C. Farnsworth, D. L. Jackson, W. A. Lien, and J. Liu, "A low-power, low-noise configurable self-timed DSP," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 32–42.

[39] L. S. Nielsen, C. Niessen, J. Sparsø, and C. H. van Berkel, "Low-power operation using self-timed and adaptive scaling of the supply voltage," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 391–397, Dec. 1994.

[40] J. Kessels, "VLSI programming of a low-power asynchronous Reed-Solomon decoder for the DCC player," in *Asynchronous Design Methodologies*. Los Alamitos, CA; IEEE Computer Society Press, May 1995, pp. 44–52.

[41] M. K. Gowan, L. L. Biro, and D. B. Jackson, "Power considerations in the design of the alpha 21264 microprocessor," in *Proc. ACM/IEEE Design Automation Conf.*, June 1998, pp. 726–731.

[42] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F.

Baez, "Reducing power in high-performance microprocessors," in *Proc. ACM/IEEE Design Automation Conf.*, June 1998, pp. 732–737.

[43] L. Beninni, G. De-Micheli, E. Macii, M. Pocino, and S. Scarsi, "Symbolic synthesis of clock-gating logic for power optimization of control-oriented synchronous networks," in *Proc. European Design and Test Conf.*, 1997, pp. 514–520.

[44] F. Theeuwen and E. Seelen, "Power reduction through clock gating by symbolic manipulation," in *VLSI: Integrated Systems on Silicon*, vol. A-28 of *IFIP Transactions*, R. Reis and L. Claesen, Eds. London, U.K.: Chapman & Hall, 1997, pp. 389–399.

[45] K. van Berkel, H. van Gageldonk, J. Kessels, C. Niessen, A. Peeters, M. Roncken, and R. van de Wiel, "Asynchronous does not *imply* low power, but . . . ," in *Low Power CMOS Design*, A. Chandrakasan and R. Brodersen, Eds. New York: IEEE Press, 1998.

[46] Semiconductor Industry Association, *The National Technology Roadmap For Semiconductors: Technology Needs*, 1997 ed. Austin, TX: SEMATECH, 1997.

[47] M. T. Bohr, "Interconnect scaling—The real limiter to high performance ulsi," in *Proc. 1995 IEEE Int. Electron Devices Meeting*, 1995, pp. 241–242.

[48] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Stanford University, Stanford, CA, Oct. 1984.

[49] I. E. Sutherland, C. E. Molnar, R. F. Sproull, and J. C. Mudge, "The trimosbus," in *Proc. 1st Caltech Conference on Very Large Scale Integration*, C. L. Seitz, Ed., 1979, pp. 395–427.

[50] W. J. Bainbridge and S. B. Furber, "Asynchronous macrocell interconnect using MARBLE," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 122–132.

[51] M. Greenstreet, "Implementing a STARI chip," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1995, pp. 38–43.

[52] K. Y. Yun and D. L. Dill, "Unifying synchronous/asynchronous state machine synthesis," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 255–260.

[53] A. E. Sjogren and C. J. Myers, "Interfacing synchronous and asynchronous modules within a high-speed pipeline," in *Adv. Res. VLSI*, pp. 47–61, Sept. 1997.

[54] D. L. Dill, "Trace theory for automatic hierarchical verification of speed-independent circuits," in *ACM Distinguished Dissertations*, Cambridge, MA: MIT Press, 1989.

[55] A. Semenov and A. Yakovlev, "Verification of asynchronous circuits using time petri net unfolding," in *Proc. 33rd ACM/IEEE Design Automation Conf.*, June 1996, pp. 59–63.

[56] R. Negulescu and A. Peeters, "Verification of speed-dependences in single-rail handshake circuits," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 159–170.

[57] S. M. Nowick and D. L. Dill, "Synthesis of asynchronous state machines using a local clock," in *Proc. Int. Conf. Computer Design (ICCD)*, Oct. 1991, pp. 192–197.

[58] S. M. Nowick, "Automatic synthesis of burst-mode asynchronous controllers," Ph.D. dissertation, Stanford Univ., Stanford, CA, Mar. 1993. Also available as Stanford Univ. Comput. Syst. Lab. Tech. Rep. CSL-TR-95-686, Dec. 1995.

[59] A. Davis, B. Coates, and K. Stevens, "Automatic synthesis of fast compact self-timed control circuits," in *Proc. 1993 IFIP Working Conf. Asynchronous Design Methodologies*, Manchester, U.K., 1993, pp. 193–207.

[60] L. Lavagno and A. Sangiovanni-Vincentelli, *Algorithms for Synthesis and Testing of Asynchronous Circuits*. Norwell, MA: Kluwer, 1993.

[61] T.-A. Chu, "On the models for designing VLSI asynchronous digital systems," *Integr. VLSI J.*, no. 4, pp. 99–113, 1986.

[62] T. H.-Y. Meng, R. W. Brodersen, and D. G. Messerschmitt, "Automatic synthesis of asynchronous circuits from high-level specifications," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1185–1205, Nov. 1989.

[63] M. A. Kishinevsky, A. Y. Kondratyev, A. R. Taubin, and V. I. Varshavsky, *Concurrent Hardware: The Theory and Practice of Self-Timed Design*. New York: Wiley, 1994.

[64] S. H. Unger, *Asynchronous Sequential Switching Circuits*. New York, Wiley-Interscience, 1969.

[65] P. Kudva, G. Gopalakrishnan, and H. Jacobson, "A technique for synthesizing distributed burst-mode circuits," in *Proc. 33rd ACM/IEEE Design Automation Conf.*, June 1996, pp. 67–70.

[66] R. M. Fuhrer, B. Lin, and S. M. Nowick, "Symbolic hazard-free minimization and encoding of asynchronous finite state machines," in *IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 1995, pp. 604–611.

[67] K. Y. Yun and D. L. Dill, "Automatic synthesis of 3D asynchronous finite-state machines," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 576–580.

[68] M. Theobald and S. M. Nowick, "An implicit method for hazard-free two-level minimization," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 1998, pp. 58–69.

[69] P. Beerel and T. H.-Y. Meng, "Automatic gate-level synthesis of speed-independent circuits," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 1992, pp. 581–587.

[70] C. Ykman-Couvreur, B. Lin, and H. De Man, "ASSASSIN: A synthesis system for asynchronous control circuits," IMEC Lab., Tech. Rep./user and tutorial manual, Sept. 1994.

[71] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Complete state encoding based on the theory of regions," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems (Async96)*, Nov. 1996, pp. 6–47.

[72] A. Kondratyev, M. Kishinevsky, B. Lin, P. Vanbekbergen, and A. Yakovlev, "Basic gate implementation of speed-independent circuits," in *Proc. 31st ACM/IEEE Design Automation Conf.*, ACM, June 1994, pp. 56–62.

[73] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Trans. Inform. Syst.*, vol. E80-D, no. 3, pp. 315–325, Mar. 1997.

[74] B. Coates, A. Davis, and K. Stevens, "The Post Office experience: Designing a large asynchronous chip," *Integr. VLSI J.*, vol. 15, no. 3, pp. 341–366, Oct. 1993.

[75] S. M. Nowick, M. E. Dean, D. L. Dill, and M. Horowitz, "The design of a high-performance cache controller: A case study in asynchronous synthesis," *Integr. VLSI J.*, vol. 15, no. 3, pp. 241–262, Oct. 1993.

[76] K. Y. Yun and D. L. Dill, "A high-performance asynchronous SCSI controller," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1995, pp. 44–49.

[77] C. J. Myers and T. H.-Y. Meng, "Synthesis of timed asynchronous circuits," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 106–119, June 1993.

[78] E. Brunvand and R. F. Sproull, "Translating concurrent programs into delay-insensitive circuits," in *Proc. Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 1989, pp. 262–265.

[79] A. J. Martin, "Programming in VLSI: From communicating processes to delay-insensitive circuits," in *Developments in Concurrency and Communication* (UT Year of Programming Series), C. A. R. Hoare, Ed. Reading, MA: Addison-Wesley, 1990, pp. 1–64,.

[80] K. van Berkel, "Handshake circuits: An asynchronous architecture for VLSI programming," *International Series on Parallel Computation*, vol. 5. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[81] A. J. Martin, S. M. Burns, T. K. Lee, D. Borkovic, and P. J. Hazewindus, "The design of an asynchronous microprocessor," in *Advanced Research in VLSI: Proceedings of the Decennial Caltech Conference on VLSI*, C. L. Seitz, Ed. Cambridge, MA: MIT Press, 1989, pp. 351–373.

[82] K. Keutzer, "Dagon: Technology binding and local optimization by dag matching," in *Proc. ACM/IEEE Design Automation Conf.*, June 1987, pp. 341–347.

[83] P. Siegel, G. De Micheli, and D. Dill, "Automatic technology mapping for generalized fundamental-mode asynchronous designs," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 61–67.

[84] P. A. Beerel, K. Y. Yun, and W. C. Chou, "Optimizing average-case delay in technology mapping of burst-mode circuits," in *Proc. Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, Mar. 1996, pp. 244–260.

[85] A. Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev, "Logic decomposition of speed-independent circuits," this issue, pp. 347-362.

[86] S. Chakraborty, D. L. Dill, and K. Y. Yun, "Min–max timing analysis and an application to asynchronous circuits," this issue, pp. 332-346.

[87] B. Rahardjo and J. Garside, *Asynchronous Tools Available on the Internet*. [Online.] Available WWW: http://www.cs.man.ac.uk/amulet/async/async_ tools.html.

[88] H. Hulgaard, S. M. Burns, and G. Borriello, "Testing asynchronous circuits: A survey," *Integr. VLSI J.*, vol. 19, no. 3, pp. 111–131, Nov. 1995.

[89] M. Roncken, "Defect-oriented testability for asynchronous IC's," this issue, pp. 363-375.

C. H. (Kees) van Berkel (Member, IEEE), for a photograph and biography, see this issue, p. 222.

M. B. Josephs, for a photograph and biography, see this issue, p. 222.

S. M. Nowick, for a photograph and biography, see this issue, p. 221.