

Asynchronous design gets a second look

By Bernard Cole
June 9, 2003

As system-on-chip (SoC) designs grow larger, designers must grapple with serious global timing problems, the effect of wire loading and timing delays and the performance hit associated with supporting on-chip communications between different clock domains.

To deal with them, developers are squeezing every trick and technique they can out of traditional architectures. They are also using the well understood synchronous logic design methodologies that have been in use since Carver Mead and Lynn Conway delineated and formalized them, and made then accessible to engineering students in their seminal 1980 textbook *Introduction to VLSI Design*.

By and large, most circuit designs are built with synchronous logic, small blocks of combinatorial logic separated by synchronously clocked registers. The biggest advantage — at least at SSI, MSI and LSI levels — was that synchronous logic made it easy to determine the maximum operating frequency of a design by finding and calculating the longest delay path between registers in a circuit.

At VLSI and now at the ultrascale levels with millions of gates it is becoming extraordinarily difficult to find and predict the critical path delays. And as process technology works further down though the micrometer to the nanometer level, things only get worse, with shot noise, charge sharing, thermal effects, supply voltage noise and process variations all making calculations of delay that more uncertain and difficult. Because synchronous logic designs are always on, balancing power and performance becomes critical as integration levels increase and the processing requirements of even embedded and small-footprint appliance designs become more demanding.

In a switch or router with dozens of boards each with several network processors operating in parallel to deliver the necessary 10 to 40 Gbit/seconds of performance, power consumption and dissipation is becoming critical, rivaling that of many enterprise server farms. The latter is also facing critical power issues as a result of the shift from centralized three and four CPU configurations to more distribution architectures consisting of dozens or hundreds of server blades, which together require the power output for many small towns.

According to Robert Payne, senior vice president/general manager in Advanced System Technology at Philips Semiconductor, (San Jose, Calif.), many consumer oriented embedded applications and small footprint computing and communications appliances are facing power issues as well. "In the desktop world, in exchange for processors that operate at two and three gigahertz, users usually have had to settle for systems that are hot to the touch, or which require a lot of external fans to bring power dissipation and heat down to tolerable levels," he said. Payne believes that with the new distributed computing and communications environment — in which one personal computer on the desktop is being replaced with one or more mobile and often wireless personal appliances, users want much of the same capabilities — without the size and certainly without the power issues.



Global timing squeeze

But "simple" power consumption and power dissipation are not the only problems, according to contributor Andrew Lines, cofounder and CTO of Fulcrum Microsystems Inc. (Calabasas Hills, Calif.). In many large SoC designs, with millions of transistors, large clock current surges are necessary: these tax a circuit's power distribution nets as well as the thermal stability of the circuit. Also, he said, there is the growing inability to control noise and metal integrity. And in some of today's SoC designs with not only millions of gates but millions of heterogeneous gate structures — DRAM, SRAM, ROM, flash, digital logic, programmable logic and analog circuitry — the job of maintaining the global clocking across the area of a chip is not easy.

As designs have gotten larger, and platforms more distributed, a significant portion of the design resources of many companies is focused on the power management issues in two areas, according to Scott Gary, group technical staff member at Texas Instrument,(Dallas), architectural level decisions implemented at system run time and those that are primarily up-front silicon gate level hardware design decisions.

Many of the run time fixes relating to power management are well known, he said. These include turning gate clocks off when not needed, activating peripheral low-power modes turning off everything except the core, the use of built-in activity detectors that can be programmed to power down the peripheral after a period of inactivity; efficient use of auto refresh modes; and at boot or during operation turning off unnecessary power consumers and sending power to subsystems only as needed and at least another dozen or so techniques.

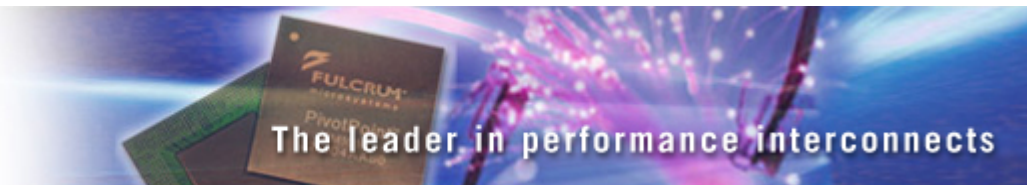
At the gate and transistor level a wide range of other techniques are employed, beyond just using a much more power efficient CMOS process. These include partitioning separate voltage and clock domains; selective scaling of voltage and frequency; gating different voltage levels to different blocks; and decreasing capacitive and DC loading on outputs, amongst other techniques.

Despite the availability of such an array of solutions, or perhaps because of the increasing complexity that they bring to a design, increasingly some designers are finding that the answers to some of their design dilemmas may lie in re-evaluating the understood and predictable synchronous methodologies and looking for other alternatives.

Async support

Now receiving a second look are a whole set of alternative asynchronous self-clocked, locally clocked and self-timed design methodologies. Contributors to this week's In Focus take a look at the growing concern in building synchronous circuits and the promise and practicality of asynchronous design.

Asynchronous techniques also got considerable coverage in the Mead-Conway text, except for very specific design problems in which nothing else would work. Still, most designers, such as online contributor Shekhar Borkar, Intel Fellow and director of circuit research at Intel Corp.'s Microprocessor Research Lab (Hillsboro, Ore.), rule out its use because the circuits designed with them have not been fast enough, are not compatible with existing EDA tools and require unfamiliar design flows.



Although it is being resisted fiercely as an all-encompassing design methodology, nonetheless major semiconductor firms such as IBM, Intel, Sun, TI and others have selectively made use of asynchronous self-clocking mechanisms in their designs. Researchers at universities continue to work on full designs and the supporting infrastructure to make asynchronous logic more acceptable to mainstream computer design. And a coterie of companies is emerging to develop the necessary technical infrastructure, including Fulcrum Microsystems, Theseus Logic, Self-timed Solutions, Sun Microsystems and, most recently, the Tangram Handshake Technology group at Philips, which is being spun out as an independent company.

Giving such companies increased motivation, said Ryan Jorgenson, vice president of engineering at Theseus (Orlando, Fla.) and a contributor to the section, is Darpa, the Defense Advanced Research Projects Agency, which has recognized the need to accelerate the EDA support for clockless approaches and has funded a new program focused on improving clockless design tools and making them usable by the defense community. The aim of this Class (Clockless Logic, Analysis, Synthesis and Systems) program will be to produce tools and systems capable of easily developing high-performance, low-design-cost, large-scale SoC designs.

Unlike the familiar design that synchronous methodologies used, asynchronous circuits (also called, variously, self-timed, locally timed and a number of other names descriptive of the different approaches) remove the need for a global synchronizing clock. Instead, the process of computation is controlled though local clocks and local handshaking and handoff between adjacent units.

What this means for high performance and power-efficient design is that such local control permits resources to be used only when they are necessary, similar to data flow architecture in the highly parallel designs used in embedded network processors. Although asynchronous designs usually require more transitions on a computational path than synchronously designed CPUs, the transitions usually only occur in areas involved in the current computational task.

According to Philips' Robert Payne, from the power management point of view, the methodology offers a lot of benefits to designers of the new generation of ubiquitously connected embedded devices. Because self-clocking allows more fine grained control over the logic, only those portions of the chip that are actively involved in a particular operation are on at any one time. This means that the majority of resources of a circuit can be focused on that chore without driving up power dissipation and consumption, the limiting factors in almost any limited resource design. "And that is the name of the game in the new computing environment," he said.

EDA backing

There is enough promise in the approach so that Philips is continuing to develop not only the underlying methodology, but the EDA tools needed to streamline designs with self-clocked logic, said Ad Peeters, senior scientist and project leader of the Tangram effort at Philips Research (Eindhoven, Holland). He said that while it is not clear to him whether the methodology will be useful in high-performance CPU design, it is obvious that there is now sufficient evidence that self-timed circuits do have advantages in the area of medium-complexity ICs in connectivity, secure identification and possibly automotive. "The technology, design method and tool set that we have developed over the years is now being applied in several product groups at Philips

Semiconductors," he said. "This has led to several successful market introductions of ICs that are completely asynchronous.

"The challenge for us typically is to translate the potential advantages of self-timed [hand-shaking] circuit operation into an advantage for an end product," noted Peeters. "The low-power advantage can be exploited to the extreme in situations where exceptions have to be monitored and handled. A clocked solution will have to set the clock-speed to the required reaction speed, whereas a self-timed solution can simply wait until the exception happens and then react immediately."

Intel's Borkar said that while Intel has used local clocking and self-clocking methodologies in the past as solutions to specific design problems in specific circuit designs, he does not feel that asynchronous logic methodologies will replace synchronous design as an all-encompassing solution. Not only are designers at the architectural and process level finding a number of innovative ways to deal with both the power and clocking issues, he said, but there are other techniques in development that offer more promise than asynchronous logic and which would be less disruptive to the existing synchronous framework.

"There are several new classes of logic circuits that are essentially extensions of synchronous logic that accomplish much that clockless logic proponents promise," he explained. "One method, called mesochronous, involves logic designs in which various portions of an SoC design are not synched independently of the logic signals, but where the clock signals accompany the data." Alternatively, he noted, Intel has been looking at plesio-synchronous logic, in which multiple clock domains distributed throughout a chip share the same clock, but where the timing is available from a separate parallel clock signal distribution system.

Another methodology that is more attractive to synchronous designers such as Heinz P. Holzapfel, senior director of technology partnership programs at CoWare Inc., is adiabatic logic, where instead of supplying a constant voltage to a chip and then clocking signals through, circuits have a periodic, sinusoidal power system that activates logic gates as needed. "While there is always the possibility we may have to adopt a design methodologies that are predicated on asynchronous, clockless logic," he said, "there are still many alternative extensions to synchronous design that can be explored."

However, for now the last word on the subject is Darpa's. "One of the areas we have been considering to support the DoD [Department of Defense] needs in advanced ICs is asynchronous logic," said Bob Reuss, who manages the Tactical Technology Office of Darpa in the Microsystems Technology Office (Arlington, Va.). "It is my belief that while Intel, et al., may be able to continue to use existing chip-design technologies because of the large amount of design resources available to them, the DoD community will not be so lucky."