

POWER IS FAST BECOMING THE BIGGEST HEADACHE FOR CHIP AND SYSTEMS DESIGNERS. BUT A STEP-BY-STEP APPROACH TO ARCHITECTURE AND DESIGN CAN YIELD DRAMATIC POWER SAVINGS.



In-car systems are using increasing amounts of electrical power.

With the advent of sub-100nm process technologies, the microelectronics business is facing major changes. System-on-chip 'super-integration' is now a reality. Chips with more than 1 billion transistors are becoming possible, running at clock frequencies greater than 3 GHz. As well as signal-integrity, reliability and statistical process-variation issues are becoming troublesome, a gap is also growing between the power needs of an embedded system and what the battery or power supply can deliver, or even what the IC package can dissipate, as seen in Figure 1.

The portable market is no longer the only one that needs low-power design efforts. Many applications, from telecom ICs to digital still cameras, PC peripherals and even automotive designs – primarily for reliability and electromagnetic compatibility reasons – now require a power-conscious design approach. The result is that system design is shifting its focus from performance alone to performance at the lowest possible power cost.

The war on power will not be won by simply selecting the best cell library and supply voltage strategy. Power-saving opportunities are available at each level of design and need to be exploited from the system level down to physical implementation, as can be seen in Figure 2. The

POWER AWARE DESIGN

by Luca Mazzoni

FOR EMBEDDED SYSTEMS

potential power savings reduce as the design phase moves downward.

This approach needs, in turn, high-level power analysis tools to identify whether a targeted macro dissipates a substantial amount of the design's power budget. Today, by the time the designers can accurately measure power consumption – at the transistor level – it is too late to act. Our internal design experience suggests that circuit and gate-level optimisation techniques typically have less than a two-fold impact on power, while results on architecture and system-level strategies offer savings of 10 to 100 times or even more.

The first opportunity for power savings comes at the application level. Optimising for power at the application level means exploring and finding ways to provide the end user with the same target functions, but at lower energy cost. It starts with a detailed analysis of the functions the system is expected to perform and how the system is exercised by the user and the environment.

CHANGING MODES

Once the application has been so profiled, it may be possible to alter the existing execution modes or to introduce new modes that execute the same job, at a lower energy cost. Although potential energy reductions are the greatest, the techniques are specific and depend on the product.

Take, as an example, a portable MP3 player. Evaluations may reveal that users will tolerate reduced playback quality as the result of reduced-precision decompression techniques that use less power than a full decompression algorithm. In environments with high background noise, the difference in quality between the two modes of decompression may not be perceptible.

It is not just portable systems that can take advantage of high-level power control. Modern cars have embedded networks that link tens of electronic controllers in the chassis, powertrain and body, as well as to telematics and entertainment systems. The growth of electronics content in cars is giving rise to power issues, as the active systems can draw tens of amps of current at any one time.

In a project for one automotive customer, the objective was to provide an efficient power-management strategy for each electronic controller unit (ECU) to reduce power consumption in stand-by mode. The problem was that stand-by current risked discharging the car battery in a relatively short time when the engines was off. Analysis of the system's various functions highlighted situations where it was possible to cut power consumption to zero in some cases, and introduce low-power states in others.

In certain cases, the controllers could be turned off completely using relays to disconnect them from the power supply. In other cases, a supply gating technique at the level of the electronic control unit was able to cut leakage power

Handheld systems need to extend battery life.



and ensure quick wake-up times and low-power switching. The power-reduction strategy also exploited voltage-scaling techniques, where the supply voltage is reduced rather than cut off completely during standby. Wake-up signals could be sent using the system bus – a master may wake up a slave ECU – or by local sources, such as cyclic timers or external events generated by sensors and switches.

The new power states introduced were able to reduce the actual stand-by current from a few milliamps to hundreds of microamps, providing up to a 20-fold reduction.

Significant opportunities for power reduction exist one level below the application layer, at the system-design level. An electronic system is a combination of hardware and software, and both provide opportunities for power optimisation. At the system-design level, the designer can use hardware-software codesign techniques to explore the allocation of intellectual property (IP) blocks, such as processors or dedicated hardware controllers, the binding of functional tasks to those blocks, and then the scheduling.

At this level, system architects compare hardwired, application-specific implementations of functions with software-programmable solutions. The scope of the analysis cannot be limited to the power consumption only, because a trade off between system flexibility, performance and cost has to be found.

Significant power reductions can also be achieved by choosing the most appropriate process technology, a choice generally made at the system-design level. As an example, the active power consumption of a commercial 32bit RISC processor core on a 130nm process can have 70% lower power consumption than its 180nm equivalent.

But the move to the 130nm process implies different IP licensing costs, silicon technology complexity, EDA tool costs. The total may prove to be excessive in terms of →

BY THE TIME THE DESIGNERS CAN ACCURATELY MEASURE POWER CONSUMPTION IT IS TOO LATE.

overall design cost. Once the system has been partitioned into hardware and software elements, the designer can begin to explore different implementations of each to determine the most power-efficient approach.

Behavioural-level analysis concerns the implementation of application specific computational kernels, such as algorithms. For example, a two-dimensional discrete cosine transform (DCT), used in the JPEG and MPEG compression schemes, can be implemented in tens of different ways. Some will support very high performance; others better power efficiency.

Both software and hardware design play a part in determining how power efficient an algorithm implementation will be. Optimising hardware for low power consumption at the behaviour level trades off factors such as parallelism against the core size, and the computation precision. Although parallelism increases the overall number of transistors in a core, it allows them to operate at a lower speed so that slower, less power-hungry cells can be used. The overall power saving can become even higher – by as much as an order of magnitude – if voltage scaling can be applied such that the hardware core is supplied with a voltage that is set at the minimum value that allows ‘just-in-time’ computation.

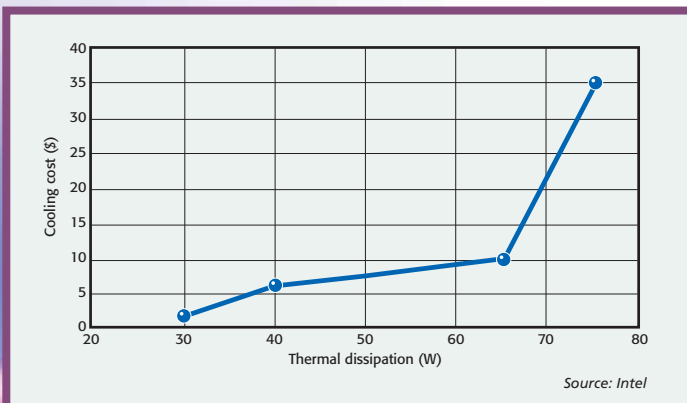


FIG 1: COOLING COST VERSUS THERMAL DISSIPATION

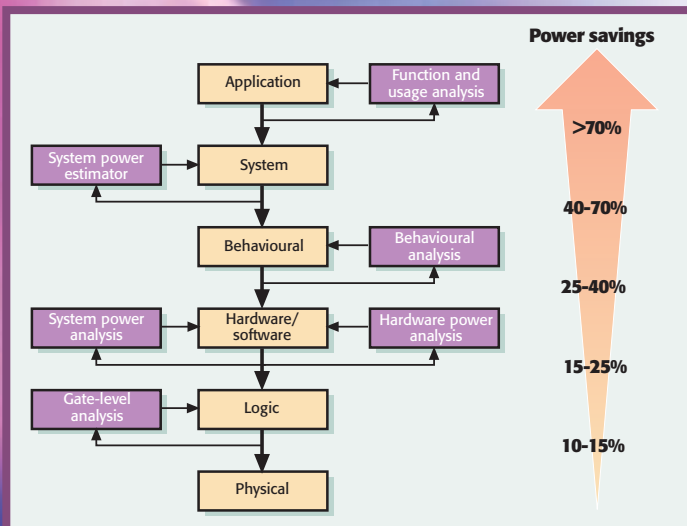
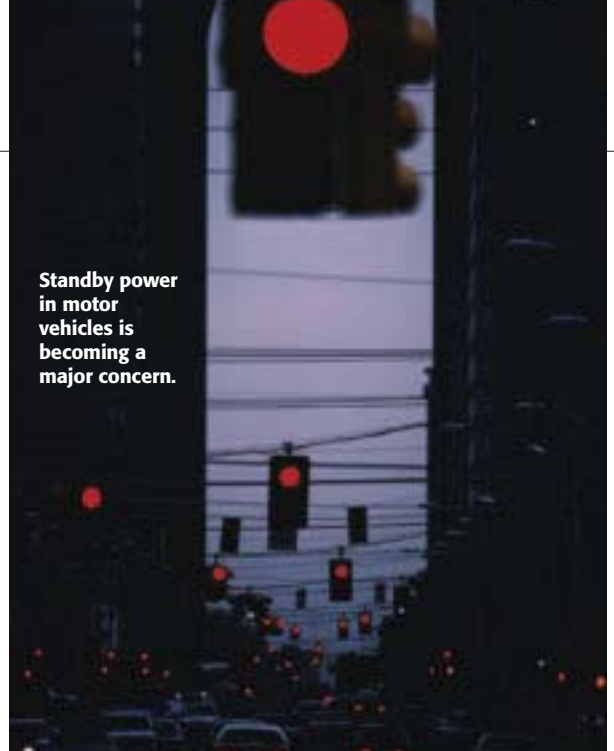


FIG 2: POWER SAVINGS POSSIBLE AT DIFFERENT LEVELS OF ABSTRACTION



Standby power in motor vehicles is becoming a major concern.

Memory accesses demand a lot of power that can be reduced by looking for a perfect match between the data structures handled by the application software code and the memory organisation. Power consumption associated with data storage and data transfers can be addressed by a careful partitioning of the memory into a hierarchy of caches, into smaller physical banks and by optimising the data packet size in terms of burst size and data width. A smaller bank of memory will have a lower load capacitance that needs driving than that of a larger bank. Also, when such bank is accessed, the other banks can be kept in a low-power standby state.

POWER-HUNGRY BUSES

Buses are among the main contributors to overall power consumption in a system design because of the capacitance effects of long, heavily loaded lines in a large SoC. Careful choice of encoding and access schemes can reduce the amount of power needed to transfer a piece of data.

Bus-invert encoding is an adaptive technique that minimises the number of transitions needed to transmit two consecutive words on a data bus. An encoder analyses on the fly if it is more convenient to transmit the second word normally or in inverted form. An extra transition line indicates to the receiver if such inversion happened, as shown in Figure 3. On a 180nm process, the encoding delay of such scheme for a 32bit bus is about 2ns, worst case.

The encoding works well where the data appearing on the bus can be treated as effectively random. But there are cases where the data words are correlated, such as samples within an audio stream. By choosing an appropriate data representation it is possible to exploit the correlation and minimise transitions between consecutive samples. Effective encoding techniques for correlated data include two’s complement and sign-and-magnitude codes on one side, and Gray, T0, Beach codes representations on another.

One highly effective behavioural-level power-reduction technique is operating system-driven dynamic power and energy management. Circuits are generally designed to deliver peak performance but the systems they drive do not need peak performance most of the time. In such cases,

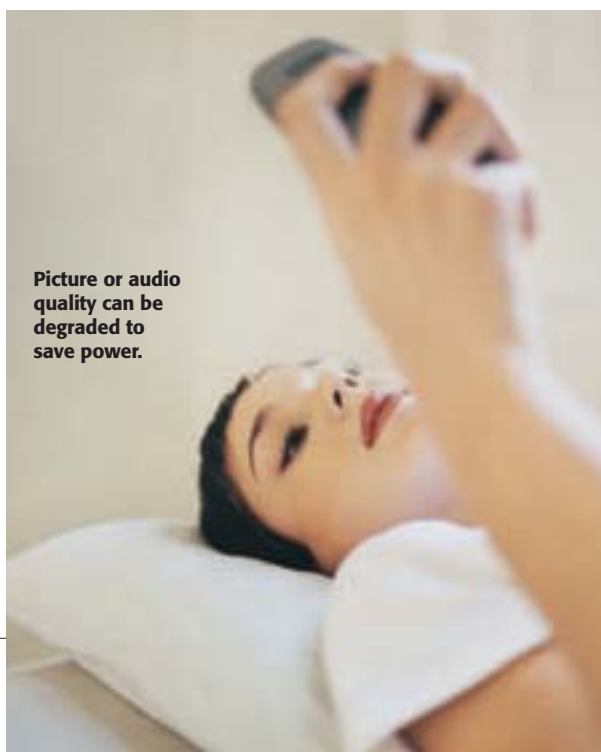
dynamic power management intervenes to cut power by turning off the clocks that feed the system, or even turning the local supply off, thus zeroing also the leakage power.

For active circuits, dynamic energy management goes further by scaling the supply voltage to reduce the energy consumed per task when full-speed performance is not needed. The reduction in supply voltage will slow the circuit down to the point where 'just in time' computation is achieved. In order to apply dynamic voltage scaling, designers need to analyse how the system is managed dynamically and then find good control policies, which could be predictive, adaptive or a combination of both. To allow the use of flexible control algorithms, the hardware should provide full visibility of its status and full control to the operating system software.

Two SoC designs, demonstrating the evolution of a multimedia processor for advanced mobile phones, provide examples of how these techniques can be combined. The initial 180nm, 1.2 million gate design for General Packet Radio System (GPRS) handsets employed an ARM processor with caches and local memories. The device had one main bus, using the Amba High-Performance Bus (AHB) protocol, with a large number of I/Os and DMA engines. A second, lower-speed, Amba Peripheral Bus (APB) protocol was used to connect configuration interfaces and commodity peripherals, such as timers.

In the second processor, for Universal Mobile Telecommunication System (UMTS) handsets, a complete subsystem dedicated to videoconferencing functions was added that included dedicated media accelerators. In this design, a bus structure based on multi-layer AHB buses was chosen. In order to limit the bus loading from peripheral ports on the main data bus, a few tricks were adopted.

Low-speed peripherals as well as those configuration ports that would not be used frequently were put on the secondary APB bus and isolated using a 'freezing' bridge. The forward and return datapaths of the AHB and APB buses were also split physically into several trunks.



Picture or audio quality can be degraded to save power.

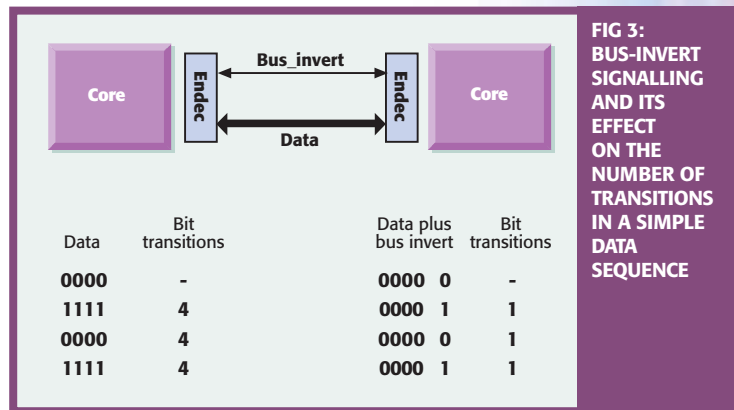


FIG 3: BUS-INVERT SIGNALLING AND ITS EFFECT ON THE NUMBER OF TRANSITIONS IN A SIMPLE DATA SEQUENCE

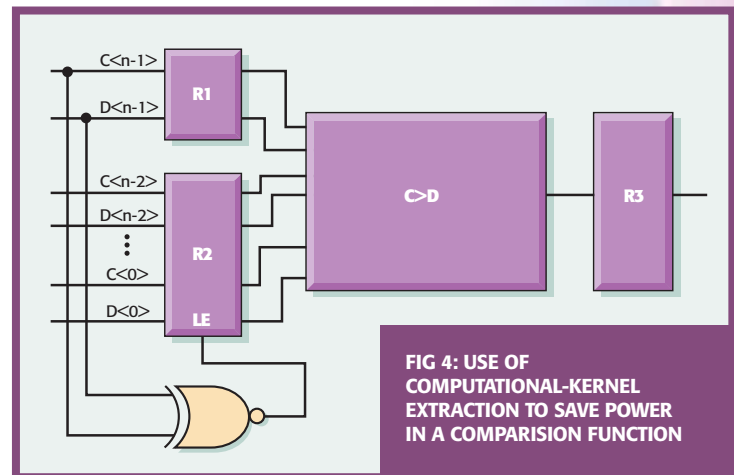


FIG 4: USE OF COMPUTATIONAL-KERNEL EXTRACTION TO SAVE POWER IN A COMPARISON FUNCTION

The decision as to which modules appeared on each trunk was made by analysing application behaviour. High-usage modules were grouped together on a primary trunk. Low-usage peripherals such as the Universal Serial Bus (USB) controller and the MultiMedia Card interface were put onto the secondary trunk. When data flow only needed the first path, the second would be kept 'frozen' and vice versa. So, on average just half of the total bus load needed to be driven. Globally, the load on each line of the 32bit data bus was reduced from about 10pF without any optimisations to less than 1pF

Dynamic power and energy management was supported by both SoCs through the use of up to 30 clock domains. The clock rate of each was put under software control so that the speed of each clock could be adjusted to the minimum value required by the actual workload. To complete support of dynamic voltage scaling, seven power domains were defined. The power to each domain could be disabled completely or scaled in combination with clock frequency. Level shifters accommodated inter-domain communication. A software interface to the power-control functions was made available by the hardware platform.

A globally asynchronous, locally synchronous (GALS) design style was used in the architecture. This style treats logic within each clock domain as synchronous. Outside each clock domain, asynchronous handshaking protocols were used to let blocks communicate with each other.

Beyond behavioural-level design, both software and hardware design offer opportunities for power reduction. In terms of software design, optimisations aimed at how the algorithm accesses memory make the biggest difference in determining power efficiency. Many compiler optimisations, for example, trade off processing speed against memory size. Similarly there are a number of techniques that can be used at the source-code level to reduce power. These include the choice of data structures, such as arrays, lists or trees, and the avoidance of memory-to-memory copies.

An effective memory hierarchy that includes caches and local memories helps reduce the energy needed for each memory. Code transformations can then help to reinforce both spatial and temporal locality and reduce the number of accesses that need to be made to the memory system in general and to main memory in particular.

As example, a programmer might choose, for the sake of simplicity, to nest loops in a way that eases algorithm development and readability. However, different transformation of those loops may yield better power efficiency by keeping data in cache memories for longer. Automated techniques have been developed to monitor the

memory usage of loops and convert them to more power-efficient forms.

Low-level optimisations at the assembly level help improve software energy efficiency. Instruction reordering can reduce the memory ‘spills’, that is, the traffic between processor registers and the memory subsystems. Specialised instructions supported by the target processor can improve code density and processing power. However, such optimisations come at the cost of software portability.

In an MPEG4 decoder design, source level-code transformations used by researchers at IMEC in Leuven, Belgium, gave energy savings of 30 to 40%. Compiler optimisations, in contrast, account for less than 1%. This confirms once again that also in the software domain the higher the abstraction level at which the problem is addressed, the larger the benefits.

At the structural hardware-design level, various techniques can be used to reduce power consumption. Common techniques include the use of resource sharing, glitch minimisation – to prevent the propagation of transitions through combinatorial gates – the extraction of computational kernels and pre-computation of values that will be used multiple times. For example, if downstream logic is driven by a magnitude comparison between two input data values coded on N bits, it is often not necessary to compare the full values. Most of the time, it may be sufficient to compare just the first one or two bits of each word to understand immediately which value is the larger, as shown in Figure 4. Designing the comparison logic to perform this operation first will yield a lower energy cost for the comparison.

EDA TOOLS CAN BE USED TO EXPLOIT POWER-DRIVEN SYNTHESIS WHEN HANDLING ‘LEGACY IP’.

POWER CONSTRAINTS

EDA tools can be used to exploit power-driven synthesis when a designer needs to handle ‘legacy IPs’, such as those coming from IP providers. In this approach, the behaviour and power consumption of each circuit type is extracted from functional simulations, and passed to the synthesis tool. Power is then set as a constraint in addition to area and timing, so that the tool tries to build a circuit with a low power topology, and map functions over standard cells having minimum power consumption.

EDA tools try to automate other power-saving techniques. For example, Synopsys Power Compiler and PKS from Cadence Design Systems support the automatic insertion of clock gating as well as operand-isolation techniques and the use of sleep modes. However, designers need to monitor the transformations made by the tools, because our experience has demonstrated that sometimes either they are not applied – when the tools are not able to identify structures that will benefit in the source code – or they are applied with negligible benefits with respect to the control logic overhead.

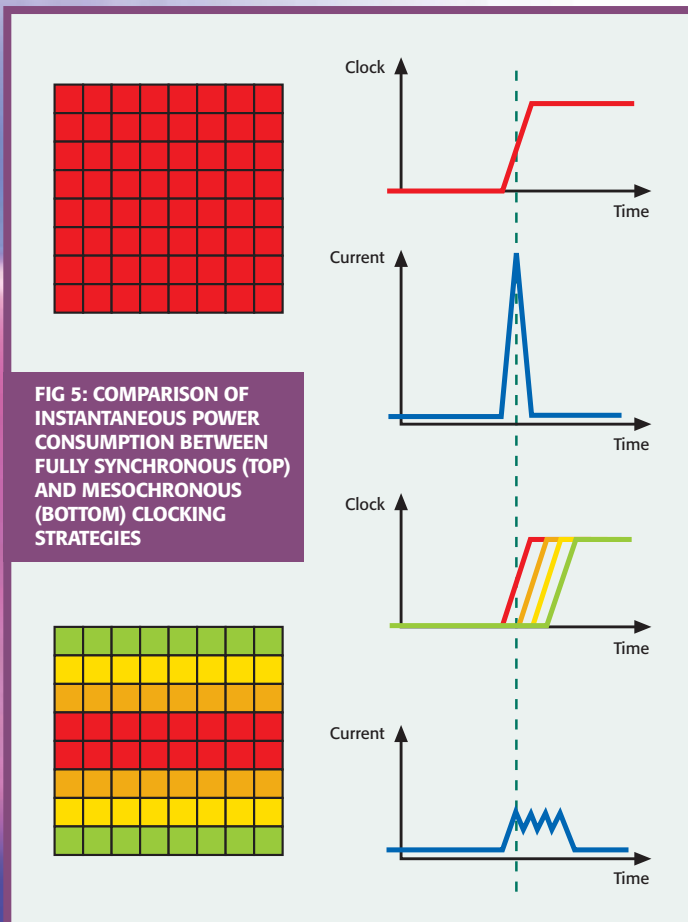


FIG 5: COMPARISON OF INSTANTANEOUS POWER CONSUMPTION BETWEEN FULLY SYNCHRONOUS (TOP) AND MESOCHRONOUS (BOTTOM) CLOCKING STRATEGIES

Use of specialised instructions and software manipulation can increase talk time.



It is worth bearing in mind that techniques such as clock gating affect also scan-based testing and automatic test-program generation. The gating makes it more difficult to control and observe circuit structures. As a result, further logic has to be added to raise coverage percentage back to an acceptable level.

Leakage power can be reduced during synthesis by exploiting the silicon technology and the cell libraries with different transistor threshold voltages that processes at the 130nm node and beyond have introduced. High-threshold transistors are slower but leak less than low-threshold structures. So, a power-driven methodology will try to use low-leakage cells almost everywhere and isolate the use of high-speed, and high-leakage, cells to timing-critical paths.

The last phase of the design process is physical design. Although power savings at this level are minimal, if designers are not careful during implementation, they could destroy all the benefits expected from the choices taken in the phases above. Power-aware floorplanning and place-and-route techniques are essential, as is power-aware clock-tree synthesis.

The standard approach to clock-tree synthesis tries to reduce the skew between flip-flops across the entire chip. This leads to large current spikes near to each clock edge and to heavily buffered, power-hungry clock nets. Capacitance of 430pF has been measured in the lab for a 250,000-gate system implemented in a 180nm process. Power-aware clock-tree synthesis uses the GALS approach to reduce the need to synchronise all of the chip's logic to one clock. The chip can be partitioned into several smaller clock domains, where limited skew can be imposed more easily.

Power-aware place-and-route uses the placement of blocks such as phase-locked loops, memories and I/O pads to reduce wiring congestion and load capacitance. Similarly, logic within the same clock domains needs to be grouped together to stay close to the clock gating and sleep-mode control logic that has been introduced during synthesis. A bottom-up flow, based on 'hard' macros, can be beneficial to keep under control both timing and power.

In one design produced for a customer, we faced the challenge of clocking a very complex and regular system – a matrix of identical blocks – containing 48 million transistors and aimed at a 130nm process. The initial design

employed a fully synchronous design. This led to large power spikes, high electromagnetic interference levels and an on-chip power grid that was expensive in terms of routing resources. A power-rail analysis revealed that, the logic in the centre of the chip would suffer an 8% voltage drop as a result of the large current peak and parasitics in the power routing. As the cell library was characterised to $\pm 10\%$, it meant that the external nominal voltage needed to be raised by 4% for the chip to meet timing constraints in the centre of the chip.

The solution proposed was to adopt a mesochronous clock strategy. This splits the clock into a 'comb' of equally spaced clock phases. The chip was separated into tiles, each fed by a clock with a different and controlled delay. The clock was distributed from inner groups to outer groups, so that switching activity radiated from the core of the chip to the outside, as shown in Figure 5. Other distribution

strategies can exploit the radial, vertical or horizontal symmetry of a chip design. The result in this design was that the voltage at the centre of the chip was restored almost to its nominal value and the peak current was reduced threefold.

In summary, although power-aware design means paying attention to power and energy optimisation at each level, there are three basic recommendations to the designer. First, make your architectural choice for low power through design exploration at the highest level possible.

Second, prepare your team, by investing in education and specific training to acquire the necessary expertise in power sensitive design flows and power-driven synthesis. Third, make sure your silicon vendor offers a good mix of technologies, ASIC libraries and low-power embedded memories. These approaches will help solve the biggest single issue now facing chip designers. ■

THE CHIP CAN BE PARTITIONED INTO SEVERAL SMALLER CLOCK DOMAINS AND LIMITED SKEW IMPOSED.

Luca Mazzoni is senior consulting engineer for Accent. - <http://www.accent.it>



Heat removal presents power-related problems even for devices connected to a fixed supply.