System Level Design Technology for Low Power Wireless Multimedia Systems

R. Lauwereins

IMEC, Kapeldreef 75, B-3001 Leuven, Belgium
Rudy.Lauwereins@imec.be

Abstract

The advent of the intelligent environment or "ambient intelligence" is a serious challenge for the systems designer. The systems of the future are small, complex, flexible and consume little energy. These conflicting requirements require new ways of designing that differ radically from conventional methods. A real software washing machine will solve the restrictions of today's methods in the near future.

I. OUR ENVIRONMENT BECOMES INTELLIGENT

We are heading towards a new technological timeframe that will become true in 2010. This time frame, the third after the mainframe and the personal computer, is named the "intelligent environment". It offer us access to all kinds of information at any time and any place. In addition, computer systems will move to the background of our daily lives while persons move to the foreground. We could say this environment is orthogonal to virtual reality that places persons in an artificially created world. Ambient intelligence aims at improving the quality of life. Technology however will remain invisible to us and systems will adapt to our behavior instead of the opposite. Terminals will evolve to wearable digital assistants, offering multiple multimedia services. They will become the "pen-and-paper" of the digital world. Besides, Intelligent sensors will be carried on and around the body. Adapted to our identity, the will inform us about our environment and our health: an electronic extension of our natural senses.

These intelligent tools only can become reality based on four technologies: (1) portable, highly miniaturized computer systems that are present everywhere in the environment (in cloths, in spectacles, shoes, etc.); (2) wireless communication between the various systems; (3) user friendly and more natural interfaces, preferably via speech processing or sensorial experience and (4) a distributed network of micro-electro-mechanical systems (MEMS), nano-electro-mechanical systems (NEMS) and a multitude of sensors.

II. INCREASING DESIGN COMPLEXITY

At the same time, innovative design methods are required to efficiently design the complex chips that power these systems. Extreme requirements are posed onto the systems of the future as well as onto there design methods. Today's design methodologies cannot cope with these requirements. We list the relevant problems and the limitations of today's methods.

A. Divergence between technology and design

The gap between design productivity and semiconductor technology is widening. The technological answer to the increasing demand for complexity, speed and power efficiency is "scaling". Scaling, the ever decreasing size of transistor dimensions, is the key for the success of micro-electronics till today. Every 18 months, we could double the number of transistors per square millimetre. The smallest dimensions of transistors in volume production are 90nm. Researchers already develop the necessary steps for (sub-)45nm. However, the astonishing speed with which the technology developed itself, is not found back in the design community. On the other hand, intelligent applications require the integration of heterogeneous functions on the same chip: System-on-a-Chip (SoC). They contain software as well as digital and analog hardware circuits combined with various components like MEMS. They should run with extremely low energy consumption, an aspect that is not well supported by today's design methods.

In the long run, researchers forecast that the technological scaling of transistor dimensions will reach its physical limits. Design and chip processing will then have to be considered together. Designers will have to give guidelines for the optimum technological configuration, already at the system level. From the start of the design, limitations and degrees of freedom of process technology will have to be taken into account.

Let us clarify this with an example.

One of the main roadblocks for the further downscaling of transistor dimensions is the interconnect, which link the transistors and the various functional units together. They consist of fine conducting metal tracks, separated from each other by isolating material. Due to scaling, more transistors are packed on ever smaller areas. Hence, more and smaller metal tracks have to be used, with increasing resistance and hence decreasing speed as a consequence. To avoid this increasing signal delay, aluminium has been replaced by copper, that has less resistivity. Signal speed can further be increased by reducing the dielectric constant of the isolating material, and hence by replacing silicon dioxide by low-k materials. Capacity decreases and hence also signal delay.

Isolating materials will however reach the limit in low-k value. Further scaling reduces the cross section of the copper wires, thereby increasing their resistance. Moreover, the length of the interconnection wires does not scale with transistor sizes. This is caused by the increasing complexity: the area made free due to transistor scaling is used to add

more functionality, thereby increasing the number of long distance interconnects that span the whole chip width.

It is hence advisable to take interconnect into account at a higher level of abstraction and hence not to neglect them anymore during system design. System designers will hence for the first time have to manage interconnect. A promising development in this respect is the division of large chips in tiles, already at the system design level. Depending on the requirements posed to these sub-systems in terms of speed and power, the interconnect within the tiles and between the tiles can then be further optimised. This approach will be key to the success of further scaling, but requires a close interaction between system designers and process technologists.

B. Co-design of hardware and software

In the past, the development of the hardware and software parts of a system were done independent of each other, resulting in many design errors that were only detected during integration and testing, when hardware and software were brought together. Since the hardware was literally baked into the chip, errors had to be corrected by the software engineers. Gradually, new IC design methods were introduced: hardware/software co-design methods. Hardware/software co-design was the first step towards Systems-on-Chip (SoC). With these design methods, it became possible to combine hardware and software step-by-step. For the intelligent environment, we will have to go one step further: the complete system will have to be described at a high level of abstraction, preferably using a single language. The design will then be gradually refined into an implementation consisting our of digital and analog hardware, as well as software, sensors and MEMS.

C. Ultra low power consumption

The intelligent sensors as well as the digital wearable assistants share one important property: they have to be extremely power efficient. Sensors will have to scavenge their own energy instead of having batteries and will hence have to use energy sources like solar energy, motion or vibration. The wearable systems will operate from batteries we do not want to continuously have to charge. Nevertheless, these systems require enormous processing power. They indeed have to run multiple dynamic applications and at the same time adapt themselves to human behavior. As a consequence, the wearable digital assistants need to have a power efficiency estimated to be between 10 and 50 million operations per mW. Intelligent sensors even 100 million operations per mW. In addition, wearable assistants have to be very flexible since they have to support multiple dynamic applications that can be downloaded over the internet at run time from application service providers. Hence, optimising functionality per power unit became the most important objective during system level design.

D. Analog and digital circuits on a single substrate

The last cause of complexity is the presence of analog as well as digital circuits on the same chip substrate. Indeed, the switching of the digital part will generate noise into the silicon substrate, which propagates to the analog parts. The operation of the analog parts will be disturbed. This interference becomes more pronounced when IC technologies further scale down. Again, designers are confronted with the limitations of today's design tools. They will have to come up with innovative solutions that take into account the substrate noise.

III. NEW SYSTEMATIC DESIGN METHODS

To get at a system implementation that does not consume much energy, an adapted design flow has to be created. A promising approach proposed by IMEC splits the design problem in two steps. In the first step, we start from a "dirty" (read: not-optimized) application specification that is cleaned by a software washing machine into an implementation oriented specification. This cleaned specification consists of multiple parallel task (multi-threaded), enabling an efficient execution on multiple processors. It has been re-written such that power consumption is drastically reduced. In a second phase, this cleaned specification is refined to software and/or hardware code. During both phases, power can be optimised using specific tools, that are described in the sequel. Imec has already developed several of those tools, with the ambient intelligent environment as driver.

A. Task concurrency management

Employing the dynamic nature of the applications results in more power efficient implementations. The multimedia applications running on the intelligent systems of the future are very dynamic in nature, indeed: tasks and complex data types are dynamically created and deleted at run-time as a response on non-deterministic events. Often tasks have to be executed in parallel. Hence scheduling algorithms are required to determine the execution order. For multiple processor systems, also assignment of tasks to processors has to be carried out. These techniques already exist, but they do not offer the modern system designer the required power efficient solutions. Most existing techniques decide on al options at design time. Designers choose a fixed implementation that fits worst case run-time behavior. However, this implementation is over-dimensioned during most of the execution time, leading to a waste of energy. Alternatively, when the designer opts for a system meeting average case requirements, problems often occur due to a lack of resources at certain instances of time. Current design tools that delay decisions to run-time, are not cost efficient enough and not enough adapted to real-time applications.

IMEC approaches this problem by dynamically minimizing power consumption of dynamic and parallel applications according to the actual load of the system. This concept is called Task Concurrency Management (TCM). It occurs in two phases, at design time and at run-time, which is its main difference with conventional methods. At design-time, various task assignments and schedules are explored and

the best ones are retained and communicated to the run-time system. At run-time, the optimum assignment and schedule are selected.

This approach is very promising for the intelligent environment. It combines large design flexibility and short design times with obtaining an optimum compromise between execution time and energy consumption.

B. Data transfer and storage exploration

Energy consumption can be drastically reduced by optimising the storage and transfer of data to memory. The multimedia applications that will run on the future systems are indeed data dominated. Due to the increasing gap in processor and memory performance, data dominated applications are slow and energy hungry. Storage and transfer of data are hence dominant factors in energy consumption, area and system performance. It is hence extremely important to reduce the size of the memories and the number of read and write operations. Since multimedia applications are often written to prove the functionality of the algorithms and not with energy efficiency in mind, it is important to optimise the code by reducing memory accesses and memory size.

With this goal, IMEC developed its data transfer and storage exploration (DTSE) methodology. DTSE starts from the source code specification of the application (e.g. written in C). It optimises the order of the data transfers and determines the optimum memory hierarchy for the data storage, The method consists of a number of sequentially executed steps. The first series of steps transform the source code independent of the target platform onto which they will be mapped. The second series of steps offers the designer insight in the trade-offs between energy and performance on the platform that has been selected. To support the DTSE methodology, IMEC developed the ATOMIUM tool suite, that consists of tools that can be applied separately or in conjunction with each other. Next to the flexibility offered, these tools provide ample feedback to the designer and hence allow interactive utilization. The designer stays in control of his/her design.

C. Refinement to hardware and software

The cleaned application specification that comes out of the software washing machine, has to be further refined to software or hardware code. This requires an adapted design environment that knows the different splits between hardware and software and that allows to describe the hardware as well as the software implementation preferably using a single language. The design methodology should allow a fast performance analysis and a fast hardware/software trade-off exploration. This way, processes can be migrated rapidly from hardware to software and vice versa, without having to re-write the code. The design environment will then refine the specification to synthesizable VHDL or Verilog code for the hardware section and for example C code for the software section.

Roadblock for the implementation in hardware is often the presence of digital as well as analog circuits on the same chip. One of the problems is the digital switching noise that is injected into the substrate and aversely impacts the performance of the analog part. This effect consists of three aspects: (1) the generation of noise by the digital circuit, (2) the propagation of noise from the digital to the analog circuit and (3) the impact of the substrate noise on the analog circuit.

IMEC's contribution to the modelling of substrate noise is SWAN (Substrate Noise Waveform Analysis). SWAN efficiently predicts the noise voltage that is injected by digital circuits, both in low-ohmic epi-type substrates as well as in high-ohmic substrates.

IV. CONCLUSION

The advent of the intelligent environment requires in the near future innovative design methods that solve the shortcomings of current methods. To obtain an energy efficient system implementation, is two phase approach is required. It transforms the high level specification of the application into an energy efficient description with parallel tasks and refines this in a second step to software and/or hardware code. During both phases, energy consumption can be optimised with specific tools, that are currently in development.

In the long term, system designers will have to take into account semiconductor technology aspects. It is clear that this requires a multi-disciplinary approach with enthusiastic teams of system designers, process and packaging experts. Power efficiency is hereby the driving force, but more importantly factors like flexibility, design time, performance and power consumption will have to be weighted against each other.