Going Mobile: The Next Horizon for Multi-million Gate Designs in the Semi-Conductor Industry

Christian BERTHET
STMicroelectronics
BP217, 38019 Grenoble Cedex,France

ABSTRACT

The complexity of a System-on-Chip design is not only in the million transistors packed in a square millimeter. The major challenge for technical success of a SoC is to make sure that millions lines of software fit in with millions gates.

In this paper, the problematic of multi-million gate design is illustrated from the viewpoint of a practical development of a complex digital system done at STMicroelectronics for a GSM/GPRS cellular application.

Categories and Subject Descriptors

C.3 [**Computer Systems Organisation**]: Special-purpose and application-based systems – *real-time and embedded systems.*

General Terms

Design.

Keywords

SoC Design, HW/SW co-design.

1. INTRODUCTION

For many people, the main complexity of a SoC is in the millions of transistors packed in a few square millimeters. This is partially true. The large majority of SoCs are made of both HW and SW: behind a multi-million gate chip, there are multi-million lines of C or assembly code…The former without the latter has less and less meaning! This general trend is confirmed by recent studies showing that out of a million engineers world wide working on embedded systems, there are eight times more engineers in SW design compared to HW design [1].

The content of the paper is organized as follows : In Section 2, we detail the general context of a multi-million gate SoC design. In Section 3, we present the problematic and raise some issues of SoC design based on our experience. In Section 4, we analyze some peculiarities of SoC design for mobile applications. Finally, in conclusion, we address the role of vertically integrated semi-conductor companies in this challenging context.

DAC2002, June 10-14, 2002, New Orleans, Louisiana, USA.

2. MULTI-MILLION GATE SOC

2.1 HW Components of a SoC

SoCs are generally made of several embedded processors, each one running control code or DSP code or both. They run either frozen code in ROM (sometimes called "firmware") or execute code located for example in external flash memories. It may happen that they form a true multi-processor system (i.e. with interactions, shared memories and/or shared peripherals). Each processor is usually associated with a set of peripherals such as DMA, interrupt controller, power and clock controller, etc... On-chip memories are ROM and RAM arrays such as caches and tightly coupled memories or slower memories. Systems-On-Chip also implement blocks of dedicated logic such as hard-wired accelerators and co-processors (e.g. Java), decoders (e.g. MPEG), dedicated functions (e.g. modem blocks), standard/proprietary IO functions (e.g. USB, UART, etc…). In general, these blocks are memory-mapped in a processor space.

2.2 SW Components of a SoC

The main categories of embedded software are OS (or kernels), drivers of HW blocks and IO peripherals, and applications such as communications (e.g. protocol stack and physical layers) or multimedia (e.g. MP3, MPEG4) software. The key factors of the software implementation are essentially the real-time behavior (in particular for OS, drivers, interrupt handlers, etc ..) and the performance of the processor and memory systems.

2.3 Complexity of SoC

For a SoC design using state-of-art .13 micron CMOS technology, gate density is in the range of 200Kg/mm^2 and SRAM density in the range of 500Kbit/mm^2. A complete SOC with say, 5Mgates and 8Mbits of SRAM, or about 100 millions transistors, will be in the range of 50 mm^2 (with pads, PLL, routing margin, etc ..).

In order to manage this complexity, a large part of the HW design work is based on reuse, up to 90% according to recent surveys [2] which confirm the effort of IP packaging and delivery methods in the semiconductor industry. In the SW area, reuse is still in its infancy, specially for low-level SW (physical layers).

2.4 Example of SOC Design: STEP-GSM

The STEP-GSM chip is a 20 Million transistor SoC design implemented in .18 micron technology. It is a digital base band

device for 2.5G GSM/GPRS cellular applications, integrating two processor cores, a high-performance ST120 DSP/MCU core and an ARM720 MCU with cache. It also implements dedicated blocks of logic and high-speed SRAM banks for the cores (caches and fast memories). The complete HW and physical layer SW were developed by ST and the GSM and GPRS class 12 functionalities were validated using a commercial protocol stack.

As the saying goes, "today's SOC is tomorrow's IP", this base band will serve as a foundation when integrated in a more complex 3G system, together with a 3G modem, a multimedia engine, and other digital functions currently developed in isolation such as Bluetooth or GPS processors. Adding all these functionalities on a single die will give the kind of complexity mentioned above.

3. PRACTICAL ISSUES OF SOC DESIGN

Different characteristics of SoC design are detailed in the following paragraphs. Although they pertain to a given project, it is the feeling of the author that they can be of interest to others.

3.1 HW/SW Co-Design and Platforms

As for many SoCs, HW and SW are designed in parallel, no one can afford to wait for the HW before starting the SW development. However, they do not use the same vehicle all along the design process due to the different levels of description/validation of the SW and HW architectures, and their availability in time.

During the STEP-GSM design, HW and SW used two distinct flows, sometimes converging and sometimes diverging. Looking back at the chronology, three main phases took place: Phase 1 is the architecture definition with very strong HW/SW interactions, Phase 2 is the development phase using separate platforms with weak interaction, and Phase 3 is the convergence on a fast-prototyping platform with extremely strong interactions.

Combined with a very secure and safe silicon design methodology, this process has shown a very fast final integration time: the porting of the entire SW on the first STEP-GSM silicon took only 4 days to perform an over-the-air call, after assembly and testing of the silicon and application board.

3.1.1 HW/SW Architecture Definition

The initial specification phase requires many interactions between the SW and the HW teams. However the focus is not the same.

SW architects are mainly interested in the algorithmic aspects of the applications (communications, multimedia) using algorithmic tools such as Matlab, and consequently, in defining the range of performances required from the HW components.

In parallel, for HW architects, the emphasis is put on refining the required performances of the processor systems (core mips, bus bandwidth, latency of on-chip and off-chip memories, etc…) and also on estimating the resulting silicon in terms of speed, area, and power consumption. Typical tools are abstract models of the SoC architecture (e.g., C threads, which can be or not mapped on the targeted cores), and spreadsheets for the silicon estimates.

The decision process to allocate specific algorithms to HW blocks rather than SW is a kind of "meet-in-the-middle" compromise, based on application and system global performances, availability of existing IP's, availability of competent resources, silicon area and power consumption. Convergence occurs when an agreement is found on the HW/SW decomposition of the SoC translated into an architecture manual which is a kind of "wedding" contract. Then, after signature, "divorce" can really start… before reconciliation.

3.1.2 SW Development Platform

The SW development of STEP-GSM was basically done using a PC-based platform, thanks to the performances of PC processors: A 1GHz Pentium under Real Time Linux is easily able to emulate the execution of C code running on an embedded 52MHz CPU and a 104 MHz DSP (the upper bound of the application requirement, in our case GPRS Class 12).

The main advantages of this platform are that it is available sooner than a prototype, it is hardware independent (i.e., the hardware constraints are postponed to later steps: no need to care about memory footprint, power consumption, memory access times, external protocols, etc ..). Also it can be used to easily migrate pieces of code from one core to the other one, since SW allocation on cores is not yet done.

The Pentium processor also communicates with a DSP/FPGA off-the-shelf board on a PCI interface to handle real-time connections to the external world, e.g. radio interface, etc…. It also emulates the actual system tracing and application SW debuggers thru UART ports, with the exact same tools as on the final board.

At this stage the Pentium execution is faking the SW partitioning on the different cores, however the SW architecture is still an abstract view of the SW, it does not even use the target OS and the drivers are very different from the final ones. The platform is powerful enough to run the whole SW architecture including the protocol stack, and it was validated up to a GPRS call.

3.1.3 HW Development Platform

In parallel, the HW development of the STEP-GSM is an RTL model with a complete testbench, developed using VHDL and Verilog simulators on Unix workstations. It covers all the specification levels, from SoC architecture definition down to the RTL sign-off. This model targets first FPGAs for fast-prototyping, and second a CMOS library for the final silicon

Interaction with the SW was limited to the definition of tests scenarii and occasional (sometimes tough) discussions in front of the "solicitor" about possible changes in the "wedding" contract.

3.1.4 Fast-Prototyping Platform

The convergence of HW and SW occurred on the fast-prototype FPGA platform as soon as both HW and SW have reached an acceptable level of maturity. Two generations of platform were used: first, an Aptix machine and then, a dedicated board (more easily duplicated and portable). Each one basically uses bonded-

out chips for the processor cores (when they are not synthesizable) and two Virtex-1000E FPGA from Xilinx for the rest of the logic, possibly with dedicated memory extensions.

This platform is the basic tool to find deep bugs in the HW. The turn-around time is very fast at this stage of development: once the bug is identified, it's a matter of few minutes for a SW fix and few hours for a HW fix, nothing compared to the cost of a mask set and the duration of a re-spin in foundry. More complex micro-architecture changes can be handled without major impact.

Main purpose of this platform is to perform the SW integration. At this stage, the SW is allocated on the different cores, using the target OS for each core, and this covers the partitioning assembly/C as well. Embedded processors' debuggers and trace tools for processors and system debug are extensively used (tracing dedicated events, bus access, interrupts) using JTAG and dedicated trace ports, in the same way as on the final silicon.

But it is also used to validate the HW architecture. The STEP-GSM implements an innovative SW partitioning that benefits from the MCU capabilities of the ST120 DSP. This partitioning lightens the performance burden on the other MCU and a direct consequence is that the overall system needs a limited on-chip SRAM memory to perform a GPRS class 12 connection. This major hypothesis of the HW architecture was validated on the fast-prototype platform, much before the availability of the first silicon and no other platform could have done it.

The fast-prototype guaranties the real-time execution (the dedicated board can even be used in a mobile environment !). Hardware constraints on the SW can be handled such as memory footprint, power optimization, memory access times, verification of external IO protocols.

As a matter of fact, the final silicon does not bring much more in terms of functionality for the HW/SW integration than the fast prototype, except that it is the final product in the adequate package which can be plugged on a target platform, and later on a real form-factor phone.

3.2 Trends in SoC Architecture

During the HW architecture phase, most of the main decisions are performed: processor cores, buses, peripherals, dedicated blocks of logic, etc .. based on different criteria such as their availability, their performances using actual memory system.

As systems get bigger and bigger, this phase becomes more and more critical. There is a trend to use standard cores, standard IP's and more recently, standard platforms. IP reuse already plays a fundamental role in the industry, obviously for time-to-market reasons, although the maturity of IP's and their verification level is not yet at the state-of-the-art. STEP-GSM makes an extensive use of pre-existing IP's and cores.

Platform-based approach is a consequence of the increasing cost of design NRE and mask set, and a way to make sure that the investments done in the industry for IP packaging will pay off. It is advocated as the only way to keep up with Moore's law.

In order to avoid the risk of slowing innovation down, as can do any standard, the right trade-off has to be found and new tools for SoC design must be invented. Bus architecture is a typical example: custom bus design and verification are error-prone tasks, and automatic bus synthesis is not yet a fully mature technology. A standard of bus interfaces will surely improve SoC design cycle, but standardization of a given bus will not suit all SoCs.

Another current trend is the possibility to use a common HW/SW platform all along the design cycle. During the STEP-GSM project, it has not been possible to do so, due to many reasons: First, SW designers are used to develop on PC and use C tools on PC whereas HW designers are used to workstations. Second, the cost of FPGA boxes and HW emulators remains prohibitive for extended usage for SW development. Dedicated boards are more affordable but, as explained above, they come later in the design cycle. Finally, the real-time aspect is a must quite soon in the SW development process justifying the need for platforms such as the PC-based one described above.

HW/SW co-design models (e.g. SystemC-based) with models of processor cores connected using BFM (Bus Functional Models) find their justification more as HW architecture exploration tools rather than true HW/SW co-design tools. Chip-level specification based on transaction-level is still to be widely accepted, as can be the RTL languages. A non-negligible issue with high-level models is to keep them updated with the architectural/micro-architectural changes which are obviously reflected in the RTL code. Even if it is the case, the use of Fast Prototyping will remain as the only reliable proof of the chip architecture concept before the real silicon is available.

3.3 ASIC Design Flow

ASIC design from RTL to layout relies on quite mature methods and flows for most of the design steps. We illustrate some characteristics of the flow and possible improvements.

3.3.1 Main Characteristics of the Flow

The STEP-GSM was designed using a flow based on RTL sign-off. This means that a lot of effort was done in RTL verification, including the early use of HW emulators based on a fast-track synthesis (at a million dollars the mask-set in the forth-coming technologies, nobody can afford for many silicon runs). RTL code is synthesizable, including the testbenches, therefore largely facilitating the design mapping onto fast-prototyping platforms. The tasks (and responsibilities) for RTL verification are separated from the RTL design. Similarly, tasks (and responsibilities) for functional RTL verification on one hand and timing/physical verification on the other hand are also kept apart.

3.3.2 Mature Technologies

Among the mature (or very mature) technologies let's mention: RTL code coverage techniques, RTL verification using HW emulators, physical design tools (budgeting, floor-planning, clock tree synthesis, gate-level synthesis and static timing analysis), combinational RTL-to-gate formal proof, DFT and ATPG for

scanned designs, multi-site data management techniques and finally bug tracking analysis. The use of formal property checking is spreading and it will surely benefit a lot from a unified expression technique of temporal properties.

3.3.3 Wanted Technologies

Looking backwards at the STEP-GSM flow, it is possible to identify at least two areas where new tools could reduce the design time.

At the RTL phase, a large part of the verification effort is devoted to chip-level test creation: for STEP-GSM, 100K lnes of C (MCU and DSP) were needed to simulate 20 millions clock cycles to cover 100% of the 100K lines of VHDL. Therefore, there is a lot of room for innovation in automatic test creation. It will not substitute for the intelligent task of hand-writing a test plan, but it will automate the tedious work of manually writing C programs to exercise RTL simulation. This will require that the very nature of the components behavior be expressed in a formal way, as was done for test generators for processors, such as Genesis from IBM. Such a tool could be combined with a coverage analysis tool and may give rise to new metrics for RTL coverage analysis.

Similarly, to improve the physical design time, effects such as interconnect cross-talk and voltage drop have to be addressed from the beginning [3]. Even if post-analysis tools exist today, these effects should be taken into account by synthesis and timing analysis tools, in order to avoid iteration loops.

4. SOC DESIGN FOR MOBILE SYSTEMS

Everything's getting worse when it comes to mobile systems. These products are defined to be small, portable, low-power, high volume, low cost and more and more complex! Compared to a simple voice-centric 2G phone, a 3G smart phone ("all-singing and all-dancing") means first, new 3G communication standards such as WCDMA (with increased band-width up to 2Mbits/s) and second, a merge with a lot of "computing-multimedia" technologies, with devices such as color LCD display and camera.

Furthermore, time-to-market pressure is constant and results from the consumer nature of the market, more and more shrinking the period of time between the system definition and the availability of a product (HW with SW validated on the field). This has a huge impact on the design of SoCs for theses applications. Methodologies allowing a rapid change in the design from a non-predictable request from the market, will definitely be favored. This pleads for the use of fast-prototyping.

4.1 Power-saving Design

With 1 Ah average, the integration degree of batteries for mobile devices is close to explosive materials and a technology breakthrough will be needed to go further. This dictates the generalization of power-saving design methods.

Dynamic power consumption of the SoC is minimized at all levels of design. At architecture level, power dissipation is taken as a major factor for the trade-off at HW/SW partitioning. A dedicated block for clock and power control is implemented at RTL level, so that voltage and frequency change mechanisms are dynamically controlled by SW. Power consumption is analyzed in detail at RTL. At design-level, low-power and low-leakage libraries are used and during synthesis, massive clock-gating is inserted

Power-down techniques are used to minimize static consumption. Leakage is a huge problem as device geometry scales down. Stand-by or sleep mechanisms are carefully tuned and power network are split in order to switch-off the current on parts of the SoC.

4.2 Limits of SoC Integration

Mobile appliances are a digest of today's most advanced semi-conductors technologies such as mixed (energy management), analog (RF), digital (base band and multimedia), and memories such as Flash and SRAM.

Integration on a single die is simply not feasible today. SoC design is already facing limits when integrating large amounts of SRAM on a CMOS device for different reasons among which yield and leakage. Doubling the size of the die to embed large SRAM banks has a major impact on the yield. Leakage is hundred times higher in a CMOS process compared to an SRAM process for the same memory size.

An alternative is to extend the SoC concept to a System-In-Package using dies from different technologies stacked in a single package. During the STEP-GSM project, a prototype has been developed with a triple stack of a CMOS SoC with a 32Mbits Flash die and a 4Mbits SRAM die [4]. If it is economically viable, this new concept can have a lot of impacts on the design, in particular at the system-level definition.

5. CONCLUSION

Vertically-integrated semi-conductors companies cover the whole range of technologies and processes from high-level system specification to foundry. We have given several examples of up-to-date technologies: fast-prototyping, power-saving and stacked packages which are complementary to address the challenges of complex mobile systems.

6. ACKNOWLEDGMENTS

I thank the HW, SW and integration teams of the STEP-GSM at STMicroelectronics, and The Who for their 1971 title.

7. REFERENCES

[1] Smith G. "The Embedded Software Design Crisis", Dataquest Presentation, VSI Alliance, Oct. 2001.

[2] Shandle, J., and Martin, G. "Making embedded software reusable for SoCs", EEdesign, March 1, 2002.

[3] Magarshack P. "Improving SoC Design Quality through a Reproducible Design Flow" IEEE Design and Test of Computers, Jan 2002.

[4] Matheron C., Damalix P. and Berthet C. "An ASIC/Flash/SRAM triple stack package", submitted to Hot chips 2002.