Verification Strategy for Integration 3G Baseband SoC

Yves Mathys
Motorola Inc.
GENEVA
Switzerland
+41 22 799 11 11
yves.mathys@motorola.com

André Châtelain
Motorola Inc.
GENEVA
Switzerland
+41 22 799 11 11
andre.chatelain@motorola

ABSTRACT

The verification strategy of the second generation Motorola tri-processors baseband chip for the 3G wireless phone market is presented. The next generation of wireless phones supports multiple wireless protocols, high data bandwidth and a full range of multi-media applications running on an open OS. The baseband chip provides the integrated processing platform for such terminals. Baseband chips are among the most complex System on Chip (SoC) of this industry.

The verification strategy to integrate such highly complex SoC is multi-fold and adaptive: hierarchical across the different abstraction levels with special emphasis on verification corners related to the abstraction level.

The SoC architecture validation and optimization step should occur very early-on in the design cycle but require high level modeling methodology to capture the complete hardware and mission critical use cases and deliver quantitative data on the architecture. Adaptive verification scenarios are discussed across the abstraction levels and the SoC hierarchy, from stand-alone IP verification, through sub platforms and to the SoC level. A collection of metrics are presented and illustrate the efforts required to verify such complex SoC. We conclude with proposals and opportunities to enhance the SoC verification strategies based on the lessons learned.

Categories and Subject Descriptors

B.7 [Hardware]: Integrated Circuits;C.4 [Hardware]: Performance of Systems

General Terms

Design, verification.

Keywords

Verification, architecture, SoC, 3G baseband.

1. INTRODUCTION

3G baseband systems on chip are multi-processors architectures with generic processing units, typically a DSP and MCU, and a set of dedicated processors for handling mission critical tasks. The motivation for adding dedicated processors is to offload the generic processing units and also to lower the overall dynamic power of the SoC. On 3G, dedicated processors are used for the 2G and 3G modem functions, hard real time complex scheduling, encryption or multi-media processing.

Memory architecture is becoming critical to feed the concurrent computing units. Special care is required to architect the memory subsystem to ensure that the overall performance will be fulfilled early on.

As 3G terminals are moving to systems based on open OS, the embedded software is no longer bound and tends to grow while the final product is being developed. One can find multi-levels of memory hierarchy for the different computing units, making the performance verification much more challenging due to the non-deterministic behavior of cache systems.

In order to achieve the low power requirements of mobile terminals for standby, voice call or PDA and multi-media applications, the 3G SoC supports a full range of power modes. Consequently the chip verification effort is increased as the system behavior must be guaranteed across all static modes and during transient.

Another aspect of complexity is the proliferation of asynchronous clock domains across the baseband. Multiple clock domains are required for optimizing the different subsystems and to lock-on the radio frequencies - 3 to 4 PLLs are becoming common practice.

The complexity of a 3G baseband SoC can be well illustrated by the following figures:

- ❑ 50-100 Million transistors, including 10-20 Million for logic
- ❑ 30-50 digital and analog peripherals
- ❑ 400-500 pads with complex muxing schemes
- ❑ 3000-4000 pages of IC specifications

Building such large SoC leads naturally to a multi-site project, typically 4-7, to cover the comprehensive expertise in all engineering domains required. With 100-200 man-years of effort on multi-sites across the globe, these projects are clearly exciting human adventures where efficient distributed database and large scale communication are key ingredients. Such highly complex

DAC '03, June 2-6, 2003, Anaheim, California, USA
Copyright 2003 ACM 1-58113-688-9/03/0006…$5.00.

SoC projects are truly verification intensive and the SoC verification strategy is a key ingredient for successful products.

This paper is organized as follow: in the next section the hierarchical verification strategy is discussed with special emphasis on the architecture verification. We illustrate the SoC verification strategy with the development of Motorola second generation 3G processors in 130nm. Finally, we highlight future verification directions in the conclusion.

2. VERIFICATION STRATEGY

The verification strategy to integrate such highly complex SoC is multi-fold and adaptive: hierarchical across the different abstraction levels with special emphasis on verification corners related to the abstraction level. The verification strategy is both top-down and bottom-up.

We distinguished the following levels of verification:

- □ SoC architecture
- □ Stand alone IP
- □ Subsystem platforms
- □ SoC integration
- □ System/Chipset

An efficient verification is based on a clear verification focus at each level of the hierarchy, well defined interface across the levels and appropriate tooling to ensure the complete verification.

2.1 Architecture Verification

The complexity of the SoC requires the formal verification of the micro-architecture to ensure that the performance and power requirements are met while the architecture specification is developed. The performance and power requirements are driven out of applications use cases. Examples are: voice call with data, PDA modes, standby modes etc.

The verification of the micro-architecture of the SoC is becoming increasingly important with the SoC complexity. In order to ensure the performance and power requirement, designers should be able to capture the entire system at a high level of abstraction early on in the design cycle.

Many Hw/Sw Co-design tools (CoWare [1], SPW [2], CoCentric [3], Seamless [4]) are not adequate for such verification as they are based on a model where hardware and software are co-developed in parallel during the development process and expect a detailed description of the complete system. For many SoC developments such as wireless cellular phones, software tends to follow hardware development since the software evolved rapidly with the market demand and the baseband hardware project is launched 1 1/2 to 2 years before final end products. Therefore, the SoC architecture model needs to cope with such complexity and be able to capture system components at a high level of abstraction.

The modeling of the SoC architecture and its validation at a high level of abstraction ensure that the system will have the required performance within the power budget. At the earliest stage of the design phase, critical architecture decisions are made during the trade-off analysis and results in architecture optimization such as: Hw/Sw partitioning, accelerators choices, bus topologies, arbitration schemes, cache configurations, memory selection, MIPS requirements, etc.

The impact of new low power design techniques, such as DVFS (Dynamic Voltage Frequency Scaling) and power gating, can be investigated with ease on system scenarios.

The SoC designers validate the architecture for different use cases. Today's applications running on complex SoC, such as 3G baseband processors, have different levels of complexity and as it is difficult to foresee which one will be the worst case, several critical use cases need to be considered to stress each subsystem of the SoC. Although the capture of each type of use case is done at a high level of abstraction due to the limited software application information available at this stage of the design, continuous refinement is needed until the use case reflects reality in terms of data transfer, computation power, deadline and tasks scheduling.

At Motorola, we developed a tool and methodology (ArchAn [5,6,7], Panama [8,9]) to model the entire system, hardware architecture and software, to perform architectural trade-off analysis though simulations. This approach allows SoC architects to model the timing and power behavior of the hardware components (CPU, cache, bus, peripheral, DMA), and capture the software applications in *Time Modeling Language* (TML). The peripherals act mainly as bus and interrupt traffic generator. The TML models the execution profile of the software functions interacting with the hardware architecture using simplified instructions. TML instructions are divided in five + one classes: READ, WRITE, EXECUTE, NOTIFY and REQUEST + CACHE available only when the CPU includes a cache.

The Figure 1 shows a small example of a task modeled in TML. In this DMA driver model example, "EX 250" represents the cycle count executed at the beginning of the routine, RD and WR represent the read and write delay when accessing system resources, the NOTIFY instruction models the communication between CPU and peripherals, the RQ instruction allows to capture task to task communication, finally some control instructions (REPEAT, IF..THEN..ELSE) model the control flow for conditional execution and loops.

```
VAR count := 10

ROUT dma_dr 3 7 50 {
EX 250
REPEAT count TIMES
   RD 2 PERIPH1
   EX 400
   WR 4 PERIPH
END REPEAT
IF USE_DMA=1 THEN
   WR 2 DMA
   NOTIFY START DMA
ELSE
   RQ CPU sw_dma
END IF
}
```

Figure 1. TML example

Timing and scheduling schemes are accurately modeled for peak loading analysis. In the early phase of the development, execution timing model is estimated or where software does not yet exist or is not accessible (customer IP), estimation on the execution time can be done by hand or automatically [10], [11].

The ArchAn/Panama generates numerous timing and power reports, i.e. CPU average & peak load, routine laxity & hold time, bus load, static & dynamic power consumption, to assist the SoC designer to optimize the architecture.

Typical simulation speed is in the range of 500 to 700kcycles/sec[1] for a complete multi-processors baseband model, allowing simulation of 120ms of W-CDMA application in less than 20 seconds.

Once the SoC architecture is frozen, the validation effort continues. The use cases captured in Panama are used to develop integration tests at RTL top level. System tests are derived from each case such as data flow, interrupts mapping, operating modes and wakeup transitions. Therefore, the use cases are further refined as applications information becomes available.

Today the tests derived from the high-level use cases are modeled in the ArchAn/Panama tool & methodology, and are manually coded in the SoC HDL environment. Investigations are ongoing to allow the SoC designers to generate integration tests automatically from the use cases modeled in TML language.

2.2 IP Stand Alone Verification

Verification is also done bottom-up, as many IP blocks are developed and verified in stand alone while the SoC is defined. Many IP blocks are re-used from previous designs, typically up 80 to 90% of the SoC content come from existing blocks requiring minor adaptations and some fine tuning modifications based on experience from existing systems. At the IP level, the verification focuses on stand-alone IP functionality, i.e. building test benches based on application stimulus to ensure that the IP is fully compliant with the specifications.

Most of the verification work is performed at the RTL level, while the C++ based description, SystemC or C++ with custom library, are used to develop reference models.

The design automation technology offers a comprehensive and efficient set of tools to cover the IP verification needs. Good examples are OpenVera [12], *e*-language [13], and others.

At this level, the tooling and the simulation speed are adequate to perform efficient stand-alone verifications.

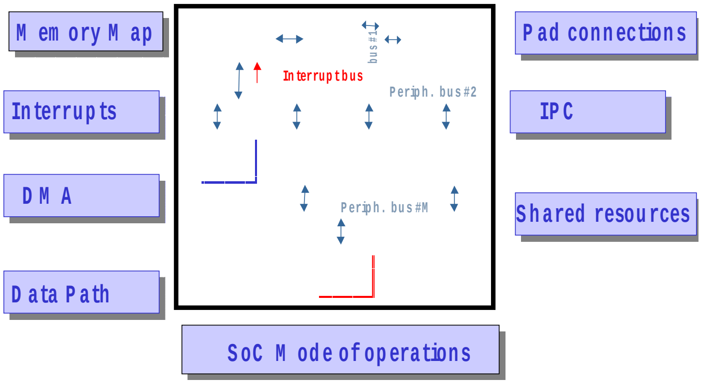

Figure 2 SoC Verification

2.3 Platform Functionality

The next level of the hierarchy is the platform level. As 3G baseband processors cover a large spectrum of functionality, from pure modem functions to full-featured videophone, the SoC is divided into a set of functional platforms.

A platform consists of a rather large set of IP blocks, computing core(s), memory banks and software functions performing a sub function of the overall system. Good examples of platforms are: the radio subsystem, RF front end to the first layer of the modem software or the complete video subsystem.

The complexity of some of these platforms requires hardware emulation to perform a complete verification of the subsystem. Hardware emulation allows verification of platforms in a real radio environment.

2.4 SoC Verification

At the SoC level, verification focuses on the integration of platforms and IP blocks by verifying signals connectivity, memory map location, connections to the PADs, data path, DMAs, interrupts and inter-process communication either on or off chip (see Figure 2). System level tests will also be developed to verify sub-system behaviors but this is quite limited due to low simulation speed at the chip top level.

3. i300 3G BASEBAND PROCESSOR

For the integration of the second generation 3G baseband in 130nm technology, we assembled a total of 1500 test patterns to validate the RTL description of the SoC. Out of these 1500 patterns, 514 were retained to validate the gate level (GL) model.

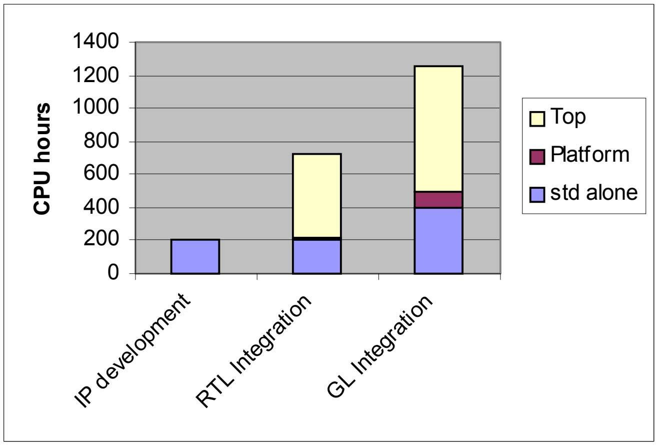

Figure 3 Regression CPU time

As illustrated by Figure 3, the processing requirement for stand alone IP verification is quite low but increases dramatically for GL verification.

[1] Simulation done on a 300MHz SUN workstation

In our project, the final regression on GL took over 1200 hours of CPU time by regression per run - 50 days on a single machine. Most of the test patterns are rather small and independent allowing a massive use of concurrent processing runs across Motorola's worldwide computing network. As teams are distributed across the globe, problems are sometimes detected in Asia, nailed down hours later by the European team and guaranteed resolved by the US team as Asia is waking up for the next day.

Test patterns on the chip gate level netlist are very expensive in computing power and hard to debug, however these are the ones which check the closest model to Silicon.

This verification strategy leads to very successful first Silicon of the baseband modem. In a matter of weeks, the baseband was capable of running voice and data calls, first in GSM then in W-CDMA.

After six months of system validation, less than 30 defects have been reported and very few functional bugs. 50% of the issues came from I/O muxing in relation to pin out or chipset interface. Around 20% of the defects were due to custom circuitry, such as PLL, latch timing, tri-state bus which do not benefit from the formal RTL to GDSII verification methodology.

We used a rather large set of patterns with post layout back annotation causing long simulation times with difficulty to debug patterns. Overall quality of the top level patterns is very key to ensure first pass silicon. Unfortunately, we lacked the tools and method to assess the quality and the coverage of the top level patterns.

4. CONCLUSION

The verification strategy of large SoCs will be hierarchical with a streamlined top-down and bottom-up verification approach. The verification must be adaptive to use the most appropriate tooling to meet specific verification objectives. We presented a pragmatic approach for SoC verification which emphasizes SoC architecture verification, which led to very successful first Silicon of a 3G baseband processor.

Future work should address the following limitations:

The link between the architecture verification and the lower levels of validation is still weak and mainly manual, automation is highly desirable.

Model proliferation is clearly an issue as there is no formal equivalence checking between models. A lot of test debug time is spent due to the non-equivalence of models and the debug of the models themselves.

As the mask set cost is getting prohibitive, in the 1-2 million USD range for sub 100nm technology, the capability of repair with metal fix only is of paramount importance. The current ECO technique based on spare cells is too limited, redundant hardware should be explored as the cost of the transistors is diminishing, especially for new IP on first Silicon.

5. REFERENCES

[1] *A design environment for heterogeneous hardware/software systems*, K.V. Rompaey D. Verkest I. Bolsens and H.D. Man CoWare, In Proc. European Design Automation Conf., Sept. 1996

[2] *SPW*, Home page, http://www.cadence.com/products/spwlibraries.html

[3] *Synopsis CoCentric System Studio*, Home page, http://www.synopsis.com/products/cocentric_studio/cocentric_studio.html

[4] *Mentor Graphics Seamless CVE*, Home page, http://www.mentorg.com/seamless/.

[5] *Architecture Analyzer, ArchAn*, User manual version 3.0, Motorola internal tool, Feb 2002

[6] *High-level Architectural Co-Simulation Using Esterel and C*, A. Chatelain, G. Placido, A. LaRosa, Y. Mathys, L. Lavagno, CODES01, Copenhagen, 2001

[7] *Level 1 Modeling of UMTS Services Using High-level Architectural Co-Simulation using Esterel and C*, Dr. P. Renard, G. Placido, A. Chatelain, Y. Mathys, SMS (Motorola internal conference), Chicago, 2002

[8] *Panama*, User manual version 1.6, Motorola internal tool, Feb 2003

[9] *Panama: System-Level Performance & Power Architecture Analysis Tool*, M. Silbermintz, A. Sahar, S. ShemTov, L. Sverdlov, I. Algor, H. Miller, E. Weisberger, S3S (Motorola internal conference), Chicago, 2003

[10] *The use of a virtual instruction set for software synthesis of hw/sw embedded systems*, D. Sciuto M. Vincenzi A. Balboni W. Fornaciari. International Symposium on System Synthesis.

[11] *Efficient Software performance estimation methods for Hardware/Software codesign*, K. Suzuki and A. Sangiovanni-Vincentelli. In Proc. Design Automation Conf., pages 605-610, Jun. 1996

[12] *Synopsis Vera*, Home page, http://www.synopsis.com/products/vera/vera.html

[13] *Verisity Specman Elite*, Home page, http://www.verisity.com/products/specman.html