Going Mobile:

The Next Horizon for Multi-million Gate Designs in the Semiconductor Industry

Christian BERTHET
STMicroelectronics
Grenoble, FRANCE
Outline

- Introduction
- SOC design in wireless applications
- HW and SW development platforms
- HW/SW prototype platform
- How did we do the SoC design?
- Challenges of SOC
- Conclusion
Convergence: Voice, Data, MultiMedia
With current CMOS Technologies, the whole logic of a next-generation mobile appliance could fit in a few square centimeters.

« Internet-in-your-Pocket » market will impose it one day or another.

Same thing occurred on a smaller scale a few years ago when the whole digital communications function was integrated on a single die.

**Systems contain more and more embedded processors: Making the real-time SW work with the HW is the real challenge!**

With Moore law, you can double a cache size every 18 months or so, improving its hit rate, but that will modify the real-time system behavior!

By chance, we are not capable to design twice more FTSS logic every 18 months (yet)!
Digital devices inside 3G phones

- 2.5G baseband function
- 3G WCDMA baseband
- Multimedia engine (MPEG4, MP3..)
- OS processor
- GPS engine
- Bluetooth / 802.11 engine
- DTV ?
HW/SW Co-design and Platforms

Three main phases:

- **Phase 1**: Architecture Definition with very strong HW and SW interactions
- **Phase 2**: Development phase with weak interactions
- **Phase 3**: Convergence on a fast-prototype platform with extremely strong interactions
Architecture from SW viewpoint

- Algorithmic and real-time models of the application

- Range of performance required: DSP mips, MCU mips, dedicated logic blocks, memory bandwidth, etc..

- Selection of processors and tools (compilers, debuggers, etc..)
Architecture from HW viewpoint

- Refinement of the range of performances on real cores architecture

- Selection of the appropriate micro-architecture

- Evaluation of Timing, Area, Power budgets
After a lot of hard work,

A few discussions and reconciliations,

real work can start!

The Red Function is for you HW guys or we’re dead!
STEP SW Development Platform

- PC-based platform
  - +1GHz PC processor is able to emulate 100 MHz DSP + 50 MHz MCU (« A 1GHz PC is damn fast!!! » a SW developer)

- Protocol Stack and Layer1 running on the PC under RT Linux
- No SW allocation on cores, no target OS
- Application Debuggers and Trace tools running on a second PC
- GPRS connection on Network
STEP HW Development platform

- RTL Development
- Integration of existing IPs
  - Processor cores
  - Interfaces (UART, USB, etc.)
- Customisation of IPs
  - Bus and other interfaces
  - Bug fixes
- Development of specific logic blocks
  - GPRS modem
  - Core Peripherals
  - Internal Memory Systems
  - External Memory Interface
Outline

- Introduction
- SOC design in wireless applications
- HW and SW development platforms
- **HW/SW prototype platform**
- How did we do the SoC design?
- Challenges of SOC
- Conclusion
Why a Fast Prototyping Platform?

- Faithful representation of the final design
- Available much sooner than the final silicon
- Guaranties the real-time behavior
- Validation of the fundamentals of the SoC HW/SW architecture
30+ MIPS (at low-voltage) is not sustainable by a traditional GSM MCU running from flash without a lot of costly embedded SRAM.
By using 1/ a different L1 SW partitioning, 2/ an efficient MCU cache, 3/ a DSP with control capabilities, we need very little embedded SRAM.
Fast Prototyping Platform

- Makes use of FPGAs, Memory modules and Bonded-out cores

- Advantages:
  - Perform SW integration on real cores (with OS, debuggers..)
  - Validate HW architecture (Bus bandwidth, External memory accesses, etc ..)
  - Find deep bugs in the HW
  - Same (even better) debug conditions (observability) as on the final target
First Prototype Platform: Aptix

ST120 Module

ARM7 Module

Virtex FPGA (GSM/GPRS modem)

Memory boards
STEP1 Prototyping Board

- ST120 + ARM720
- Virtex 1000E FPGA: GSM/GPRS modem
- Interface to JTAG
- Interface to RF
- Virtex 1000E FPGA: SIM, keyboard, display, 2UARTS
Prototyping Environment

- STEP1 BaseBand Board
- RF GSM/GPRS Board
- JTAG Connection to SW Debuggers
How did we do the SOC design?

- STEP-GSM chip
- 49 sqmm
- 0.18 CMOS
- +20 Millions transistors
- First GPRS call 4 days after tester-out
- No bug found before GPRS FTA
A lot of efforts in RTL Verification

- RTL Sign-off
  - Synthesizable RTL code and test-benches
- A lot of efforts in RTL verification
  - 500 hand-written C test-cases running on MCU and/or DSP
  - More than 100 K lines of C
  - 20 Millions simulated clocks
  - 100% Code coverage
  - Use of property checking
- Early use of HW Emulators
  - 1MHz simulation engines
  - Accelerated regression testing
  - Validation with JTAG debuggers
Bug Tracking – Project phase analysis

Problems by State - Severity 1 -> 5
-o Identifier ASC (Identifier notequal GNBan00034 && Project isequal...)

Fast Prototyping
Architecture
RTL Design
Functional Verification
Backend
Multi-site data management

- HW team in 2 different sites, SW team in 2 other sites
- HW and SW Database update performed every day
- Merge of branches under the responsibility of owner
Other mature technologies

- Physical design tools: Synthesis, floor-planning, clock-tree synthesis, static timing analysis
- RTL-to-gate Formal Proof
- DFT and APTG
Verification of deep sub-micron effects

- STEP-GSM contains no less than 27 clock domains: lot of efforts in clock verification
- Quantitative IR drop analysis and supply network verification
- Cross-talk avoidance rules and Post-routing verification/fixing
Challenges of SoC design

- RTL Verification related
  - IP verification suites generally not reusable at SoC level
  - Efficient SoC verification requires 100’s thousand lines of C
  - Need for automatic SoC test generation

- Design related
  - Need for single-pass flow accounting for submicron effects

- Debug related
  - Does SoC mean the end of Logic Analyzers and Scopes?
  - JTAG debugger access is bandwidth limited
  - Need for observability/controlability rules/tools at SoC-level

- Process related
  - Systems integration of different processes is reaching limits
  - Transistor leakage in CMOS technology prevents from embedding large amounts of RAM in mobile applications
  - Need to develop new methodologies for System-In-Package
Ways to address SoC challenges

A baseband/SRAM/Flash Triple-stack

**Advantages:**
- PCB space saving
- Increase External Memory bandwidth
- Decrease memory access time
- Decrease power consumption
- Keep the Best of the 3 technologies


**Conclusion**

- **Mobility is everywhere**
  - Non-predictible market
  - Products must be ready before the standards!
  - Design tools are changing
  - SoC Designers are changing (C + RTL + physics profile)
  - SoC databases around the globe

- **Complex SoC projects require many different skills and technologies under the same roof**

- **This broad collection of skills and technologies is the major advantage of semi-conductor companies**