SoC Design and Test Methodologies for Wireless Communications Baseband Processors

Henry Ye
Alcatel-Lucent
Hhye@alcatel-lucent.com

Overview

Evaluation of wireless communication standards.

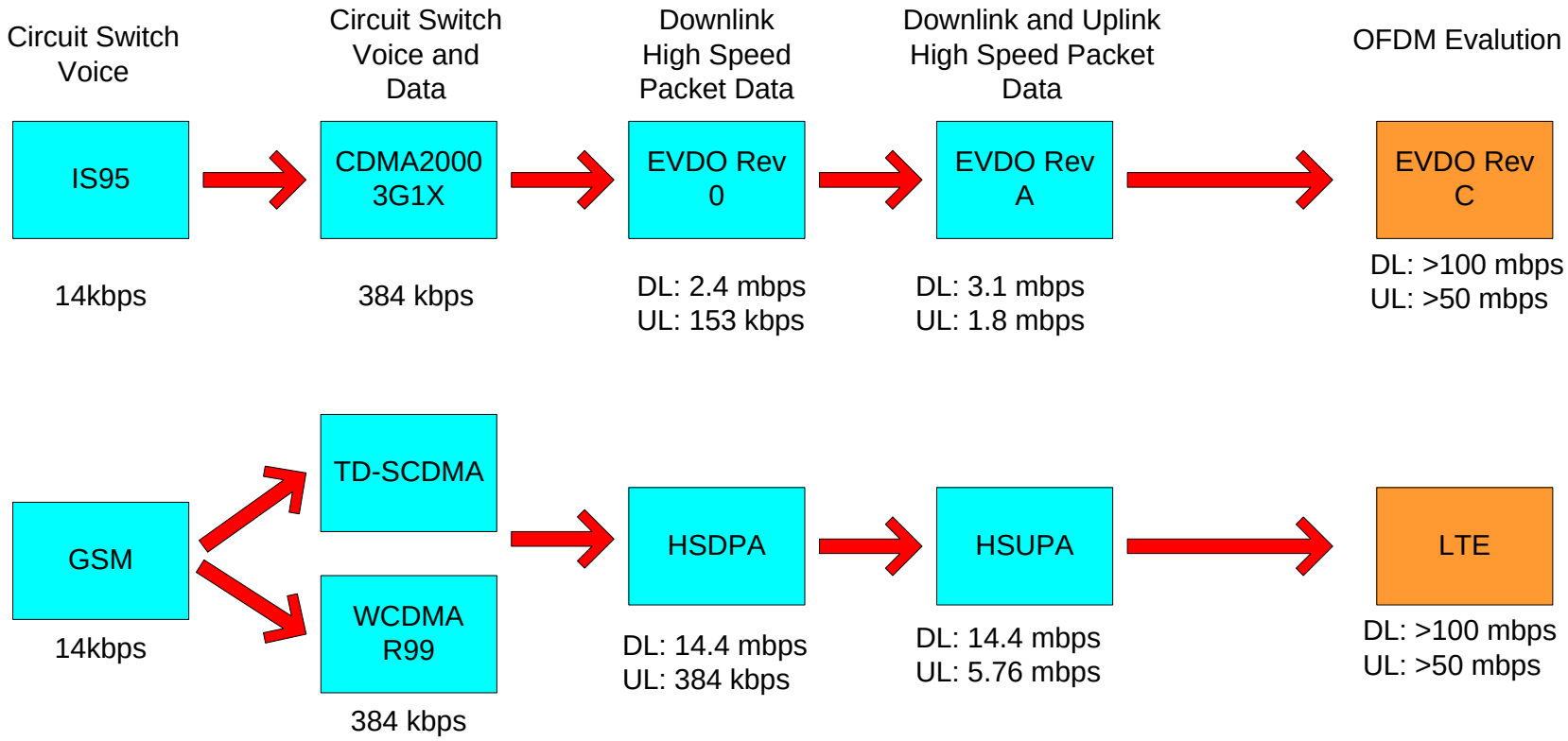
Examples of SoC device used in wireless communications.

Challenges in SoC design and verification.

Guidelines for SoC partition.

Guidelines for SoC verification/co-verification.

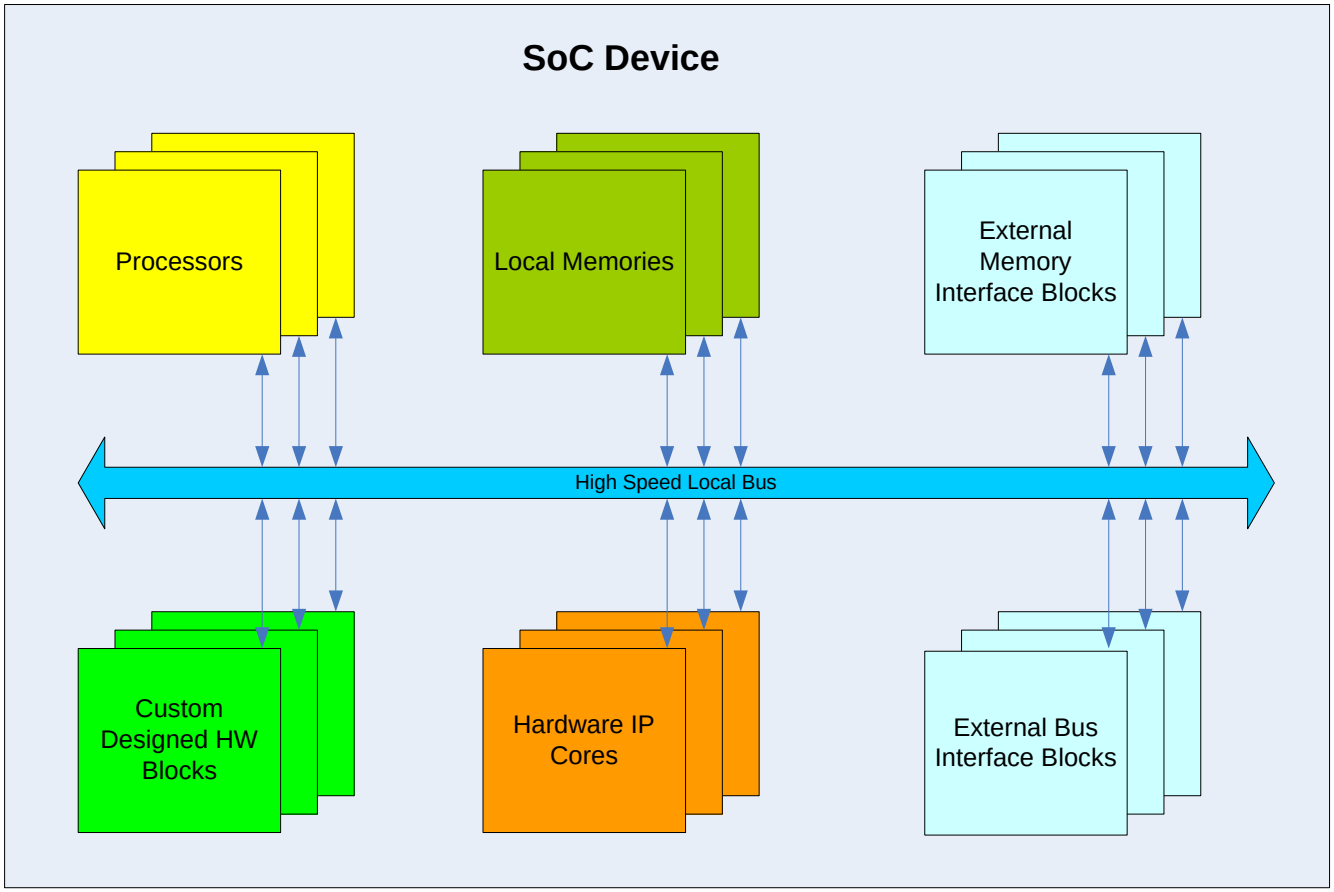
Evolution of Wireless Communication Standards



Higher level of integration is required for baseband processors in both Base Stations and terminals, in order to support the increased data rates in newer standards.

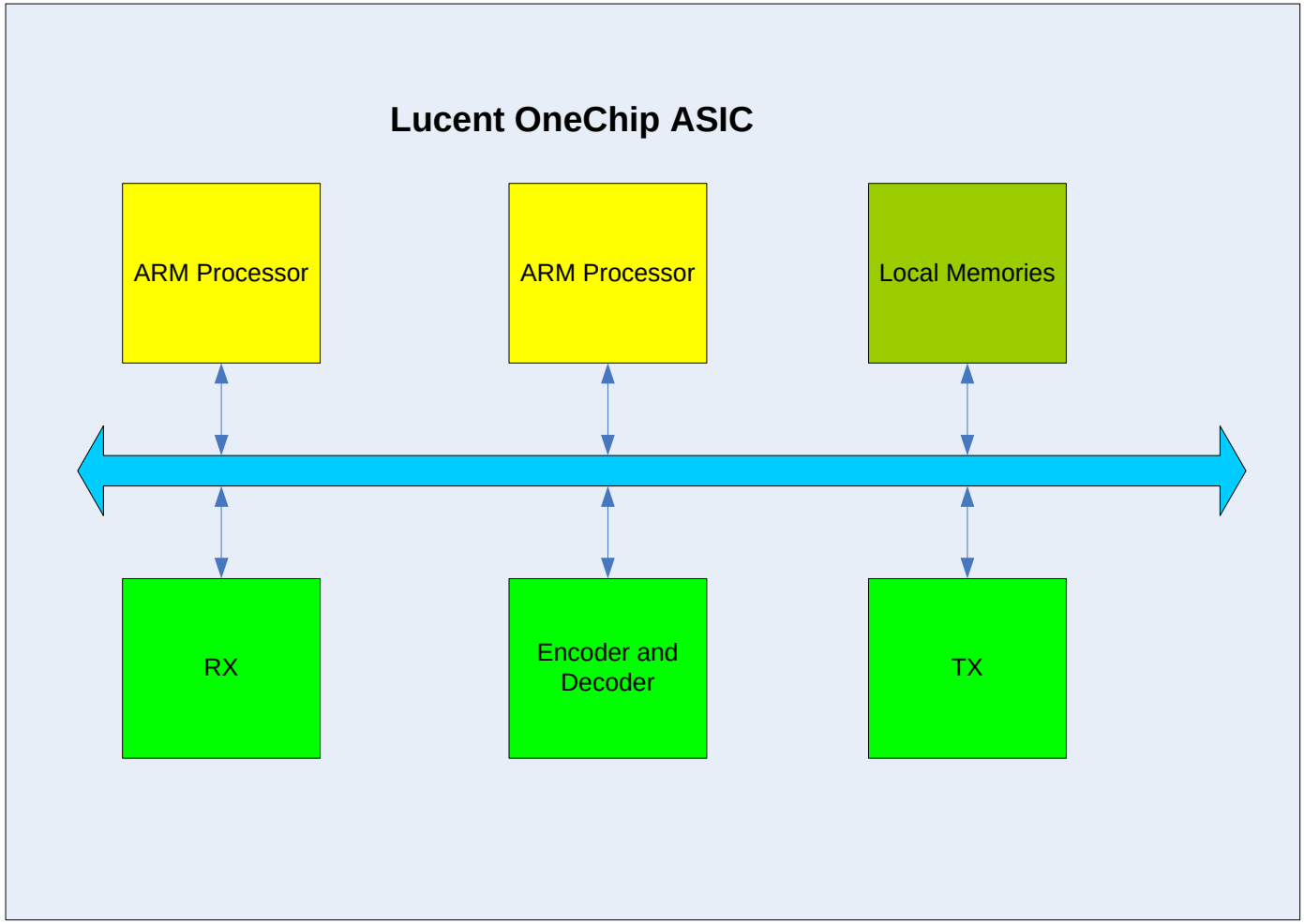
- The advancement in VLSI technologies and the appearance of SoC devices have made this possible.

Simplified View of a Generic SoC Device



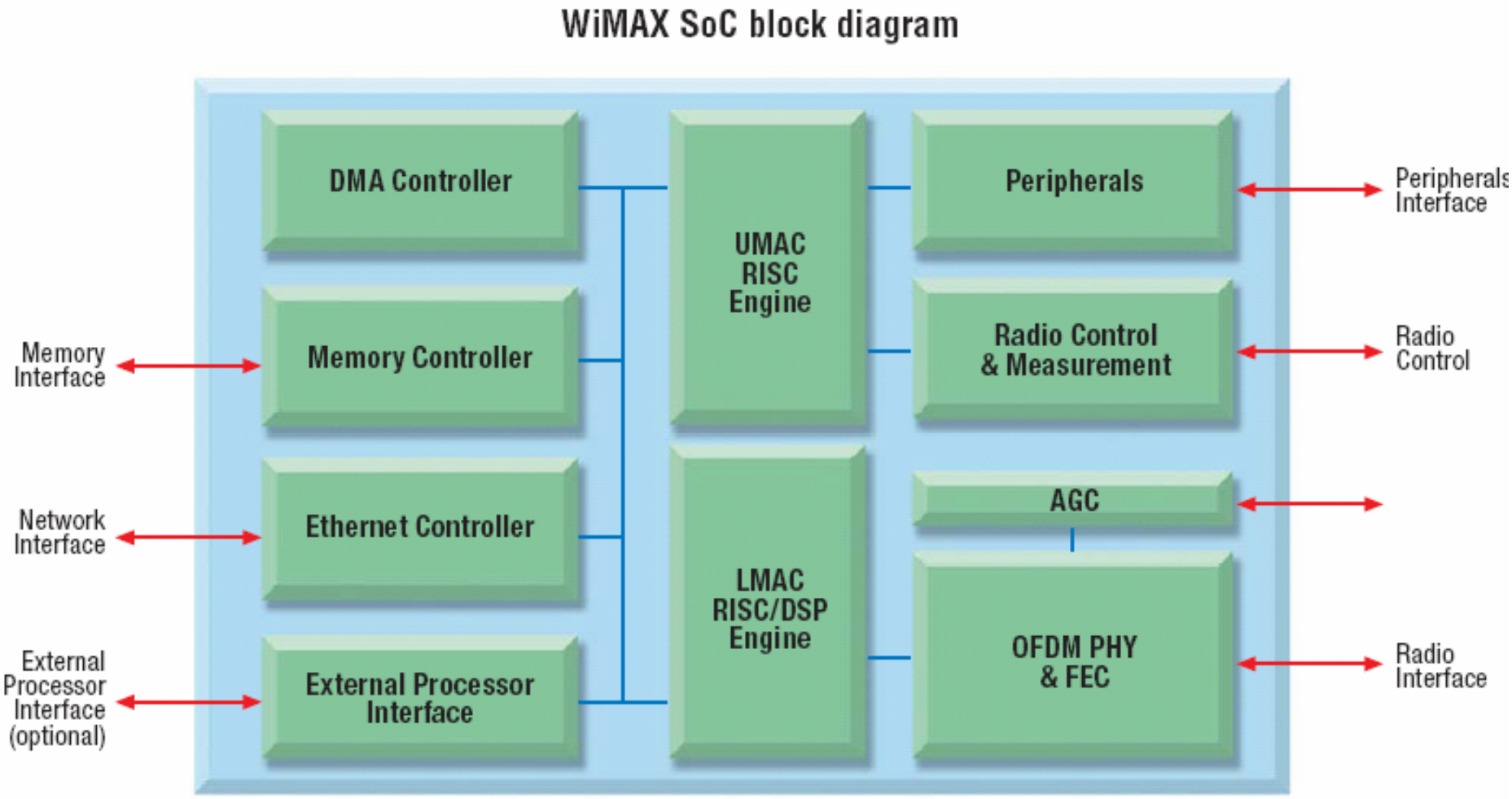
A SoC device can have some or all of the above blocks

SoC Example 1: Lucent WCDMA OneChip ASIC



High density multi-channel infrastructure baseband SoC device.

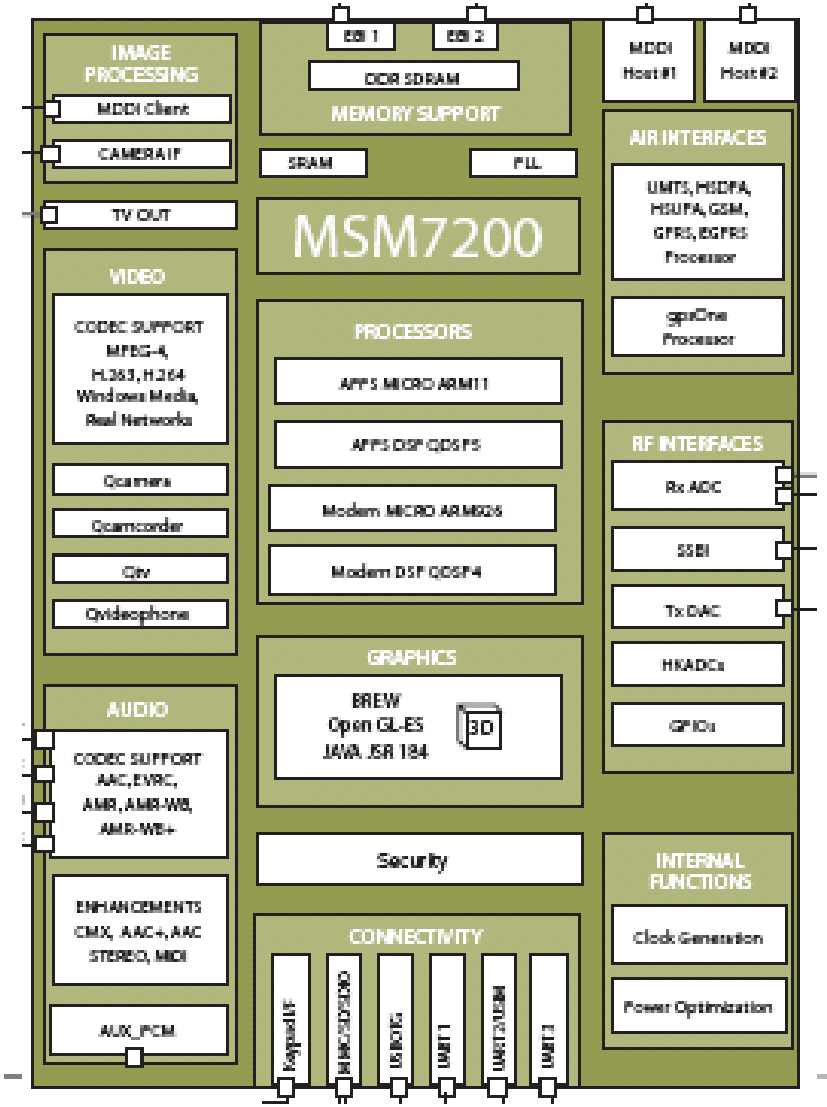
SoC Example 2: Fujitsu WiMAX MB87M3400 PHY/MAC Processor



Source: http://www.fujitsu.com/downloads/MICRO/fme/wimax/Wimax_3400.pdf

Contains 1 ARM and 1 DSP

SoC Example 3: Qualcomm MSM7200



Source:
http://www.cdmatech.com/download_library/pdf/msm7200_chipset.pdf

Supports GSM/GPRS/WCDMA/HSDPA/HSUPA

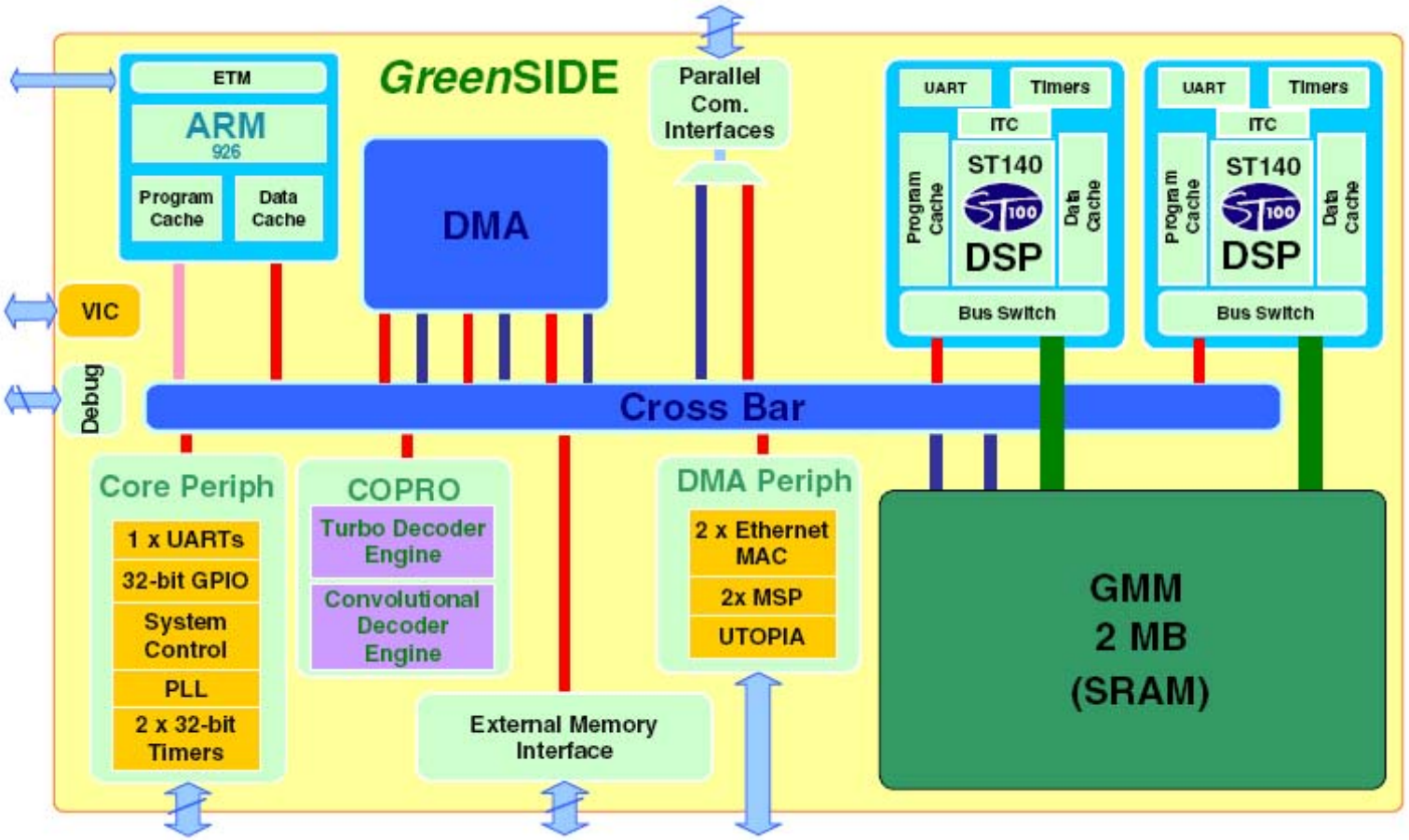
Support 7.2 mbps on the downlink (HSDPA) and 5.76 mbps on the uplink (HSUPA)

Contains 2 ARM processors and 2 DSP processors.

Also contains many hardware IP cores for specific functionalities such as

- Imaging processing, video processing, audio processing, and graphic processing.

SoC Example 4: STMicro GreenSide STW5100



Source: http://www.st.com/stonline/products/literature/ta/11145.pdf

Contains 2 DSPs, 1 ARM, 2MB SRAM, and hardware IP cores for decoders.

Served as a SoC platform for infrastructure baseband processor market.

Customers are responsible for most of the sw development.

Challenges in SoC Developement

Tow major challenges in SoC Development:

- The partition of functionalities between custom-designed hardware, hardware IP cores, and embedded processors.
- The functionality verification/co-verification.

While there is no single methodology for all SoC design and verification, this presentation is intended to provide some guidelines of the methodologies that may be applicable to the current generation wireless communication baseband processors.

Guidelines for SoC Partition (1/3)

Embedded processors are ideal for functionalities that require significant amount of flexibility and programmability.

- The tradeoff is higher cost and power consumption

Functionalities with heavy computational complexity are usually put in custom-designed hardware.

- The tradeoff is the loss of flexibility.

Example:

- The CDMA chip level processing can be done in custom-designed hardware while symbol level processing is handled in DSP.
  - Flexibility of DSP can allow longer time interval for algorithm development and provide option for algorithm improvements.
- However, partial or all of the symbol level processing may have to be implemented in custom-designed hardware due to various reasons, e.g.,
  - Sensitivity to cost and high power consumption such as for terminal devices.
  - The whole symbol level processing can be enormous for infrastructure devices due to high channel density and/or increased data rates.
  - More experienced teams tend to put more into custom-designed hardware to save cost and power consumptions.

Guidelines for SoC Partition (2/3)

Unlike physical layer processing, most higher layer processing fits into embedded processors well when implemented in the same SoC device.

- For example, MAC layer functionality may be implemented in embedded processors such as ARM or DSP.
- SoC devices used in terminals tend to use more embedded processors due to more higher layer functionality integrated into the same SoC.

SoC architects and designers have to select the embedded processors carefully based on many factors:

- What embedded processors supported by the chosen ASIC foundries,
- the intended tasks,
- the design/verification methodologies,
- the available tools,
- and the cost/power consumption budget.

Guidelines for SoC Partition (3/3)

Hardware IP cores are usually designed by third parties and can be licensed with costs. They are used when providing business advantages such as when

- the SoC design team does not have expertise to develop it by internally, or
- using external supplied hardware IP cores can simply save development cost.

For reliability reasons, hardware IP cores targeting well-defined functional blocks are likely to be used for easier integration and verification.

- Some commonly used baseband IP cores:
  - Turbo decoder and encoder, IFFT and FFT (in OFDM standards).
- Interface specific hardware IP cores:
  - PCI/PCI-express cores, external memory controllers, ..

SoC Verification -- Overview

Two major steps in SoC verification:

- Block level
  - Focus on the validation of each individual block
- Device level
  - Focus on the verification of the integrity of the whole device and the interactions between all blocks (such as between custom-designed hardware, embedded processors, and hardware IP cores).

Note that there is no unique SoC verification method. Each team develop its own method based on its experience, the available tools, the functionality of the SoC device, and the chosen IP cores and embedded processors.

SoC Verification – Block Level

Block level verification for custom-designed hardware

- Bit exact model is commonly used for the verification of custom-designed hardware blocks.
  - The integrity of the bit exact model (usually in C or C++) can be verified through simulation.
  - The purpose of the verification is to ensure the hardware model (e.g., in VHDL or Verilog) matches with the bit exact model

Block level verification for embedded processors

- Step 1: Simple test to make sure the processors can access the peripherals and memories through its internal buses.
- Step 2: Verify the integrity of the software code that will run in the processors through simulation.
- Step 3: Verify the loading of the embedded processor is within the practical range and the time critical processing can be finished in time.
  - Cycle accurate simulation tool is usually used.
  - SoC team needs to define "stressful" test vectors for processor's loading and critical time analysis.

The verification of hardware IP cores may be similar to that of custom-designed hardware blocks, if simulation models are provided by the IP vendors.

SoC Verification – Device Level

The focus is on the interactions of different blocks.

Co-simulation is commonly used so that hardware model and the models for embedded processors can be simulated together.

- Bit exact model may still be used to compare against the results of the co-simulation.
- Popular co-simulation tools such as Seamless from Mentor Graphics are used.

However, co-simulation may be too slow for complicated SoC. As a result, emulation tools are usually used.

- The whole SoC device may be mapped into FPGA based emulation tools.
- In addition to gaining simulation speed, emulation provides a platform for HW/SW integration.

It should be noted that special test images rather than real software loads may be used in the embedded processors at this stage to focus on the interactions between processors and the rest of the device.

- The core algorithm implemented in embedded processors can be easily changed and will likely keep changing.

Summary

The increased data rates in later wireless standards require higher level of integration in baseband processors, which can be realized in SoC devices.

The two major challenges in SoC design are how to partition SoC device and how to verify SoC device.

While there is no single methodology for SoC partition and verification, some guidelines are provided in this presentation.