# Infrastructures for Modular Integration of MPSoCs

Pieter van der Wolf
NXP Semiconductors Research

MPSoC'07
June 25 - 29, 2007



## Abstract

- The current practice of IP-based SoC integration exposes an overwhelming complexity to the SoC integrator, who has to understand the intricacies of H.264 decoding, 3D graphics, WLAN, 2G/3G cellular communications, etc. in order to integrate these functions into a single SoC with an appropriate SoC infrastructure. We propose an hierarchical SoC integration style where functions are pre-integrated into coarse-grain pre-validated subsystems by domain-experts. Each subsystem is then delivered for SoC integration with a 'data sheet' containing an abstract characterization of the subsystem in terms of functions, interfaces, protocols, communication workload and associated latency constraints for its correct operation. Such data sheets enable the SoC integrator to integrate a set of subsystems in a modular and predictable way with an underlying SoC infrastructure.

## Outline

- Introduction
- Modular system architecture: key concepts
  - Physical view
  - Logical view
- Predictable system integration
  - Subsystem characterization
  - Integration of subsystems
- Conclusion



## Problem Definition

- SoC integrator must deal with:
  - Many small IPs
    - Rather than pre-integrated functions
    - Requires extensive know-how
  - SoC performance hard to predict
    - Interference through shared resources
    - Data dependences (caches)
    - Late software integration
  - Bottleneck to SDRAM
    - Latencies that IPs cannot survive
  - Constraints on cost, power, etc.
- Main theme
  - Modularity and ease of integration

## 5 Key Elements of Next Generation Platforms

1. Modular systems based on coarse-grain subsystems

inter

2. Scalable & efficient communication infrastructure

3. Open standards for communication interfaces

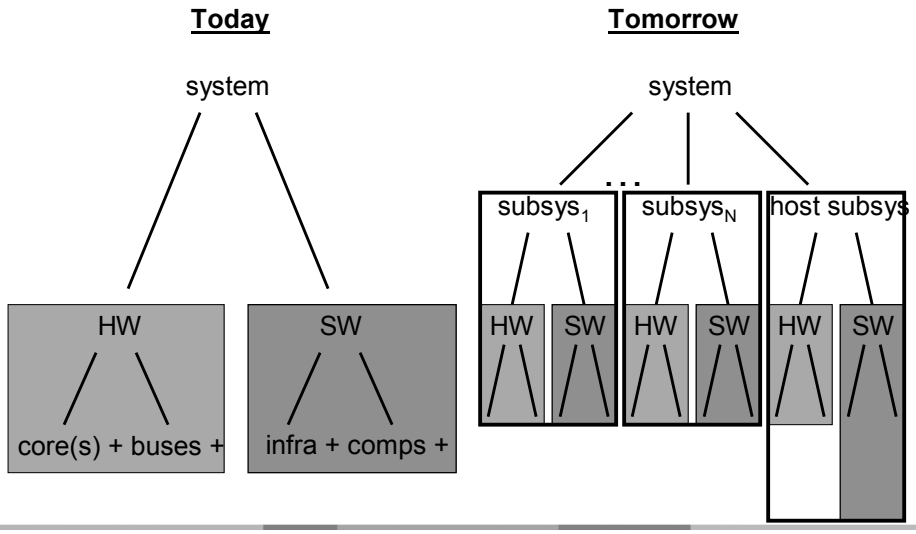
4. Controlled sharing of mem, interconnect, power, ..

intra

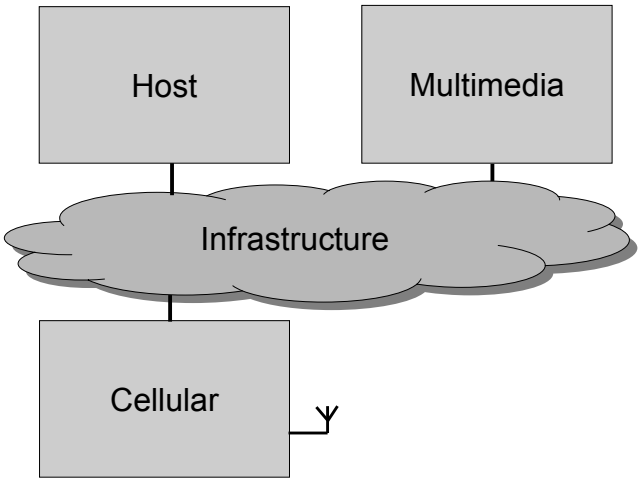
5. Programmable architectures for subsystems



## System Architecting

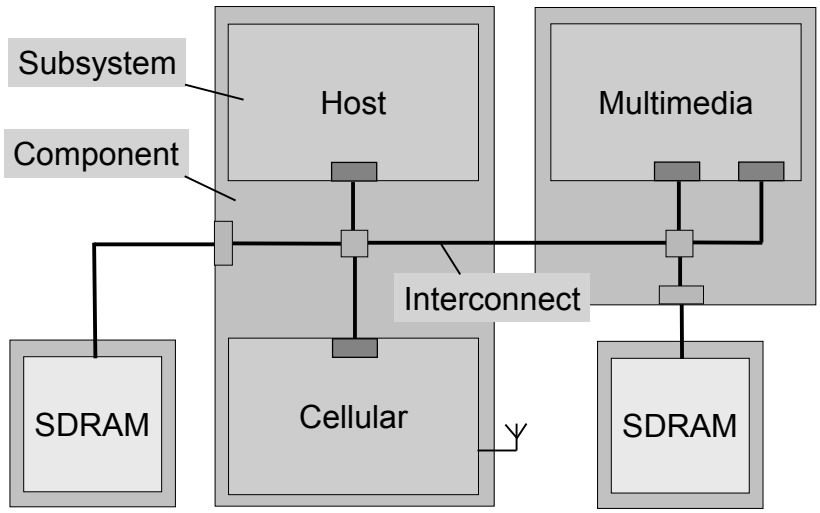


## Modular System Architecture

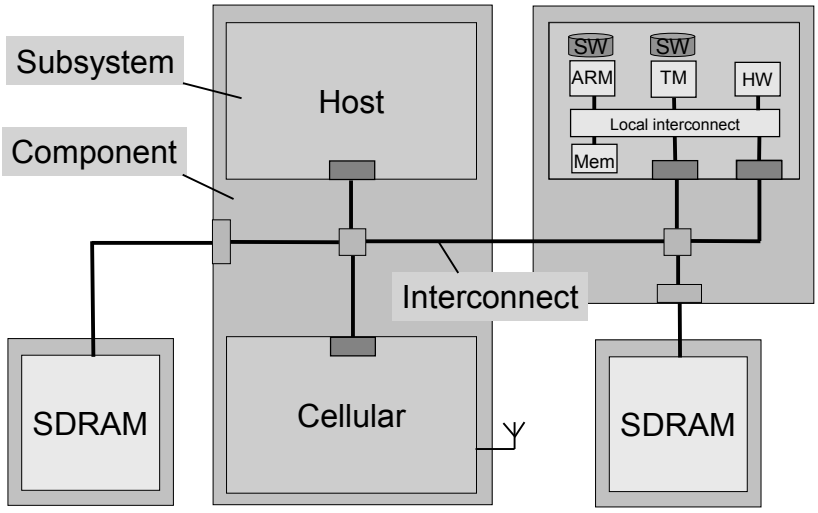





## Physical System View

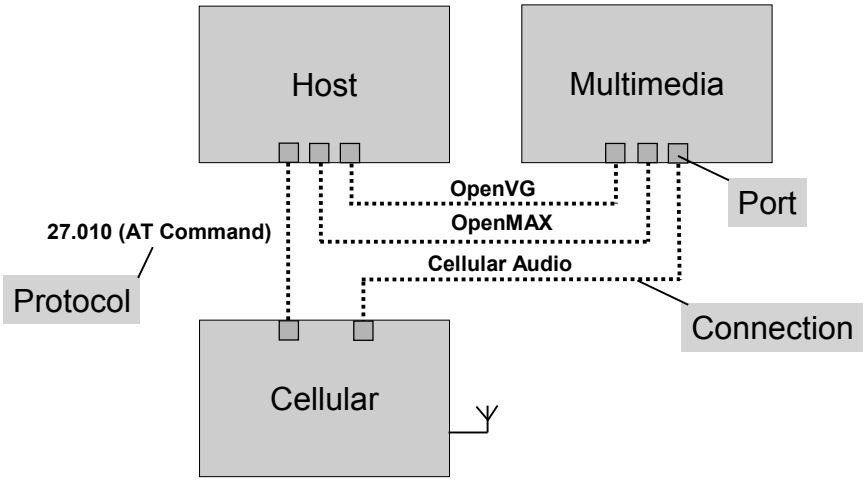


## Physical System View

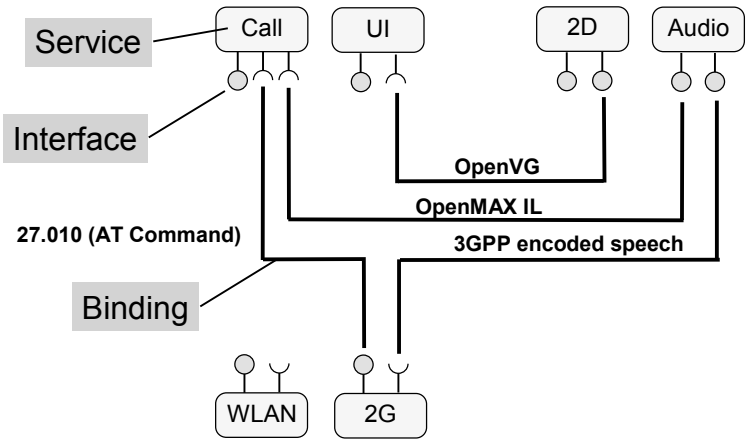





## Ports, Connections and Protocols

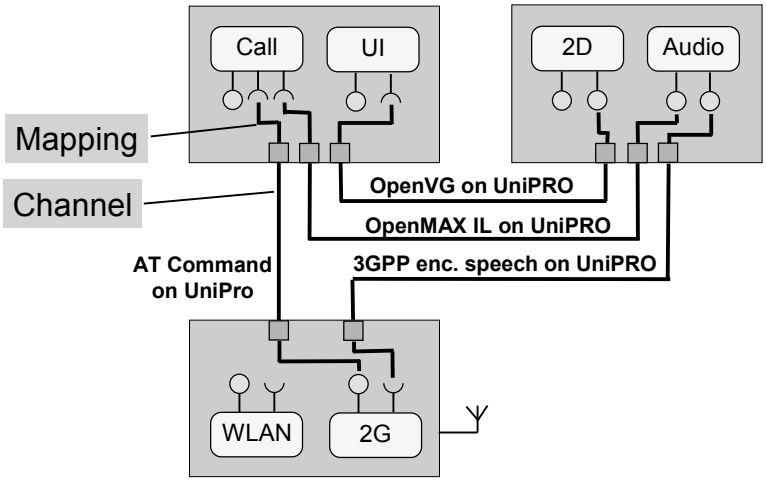


# Logical System View






# Combined System View



## Key Concepts

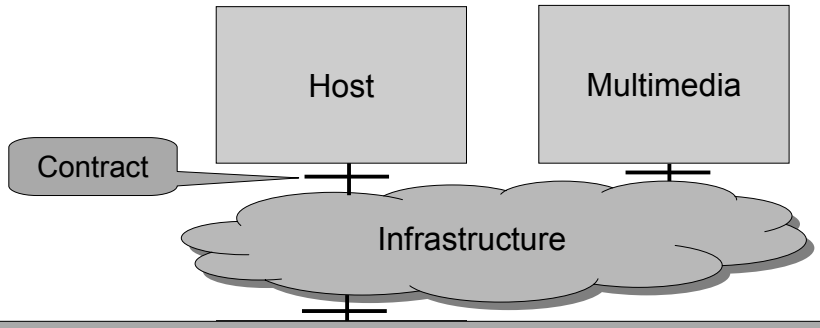
- Subsystem
  - Container of services
  - Unit of design, delivery and integration
  - Pre-integrated and pre-validated
- Service
  - Functional entity with well-defined interfaces
  - Unit of discovery, resource management and invocation
  - Implementation defines resource usage
- Interface
  - Access point for communication with a service
  - Interface standards are key to interoperability



## Integration of Subsystems

- Standardized protocols
  - At different levels
    - Transport level and below
    - Above transport level
  - Seamless integration of subsystems and SoC infrastructure
  - Interoperability of subsystems
- Performance methodology
  - For predictable integration of subsystems from different sources
  - Guarantee performance while subsystems share resources

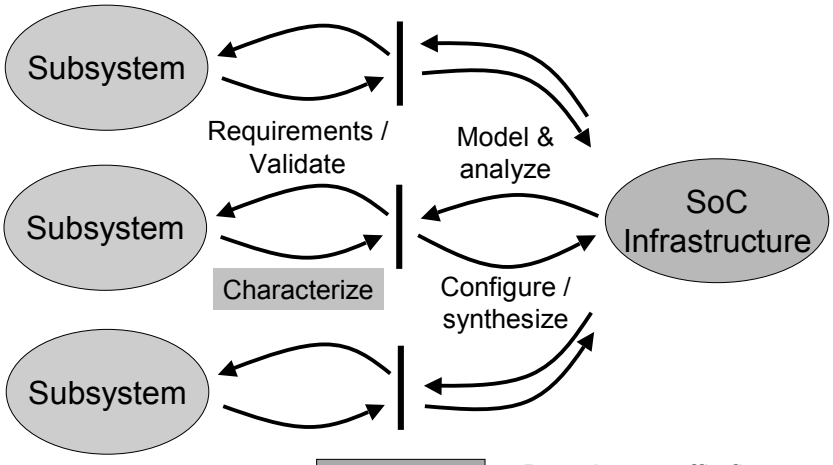
## Performance Methodology




- Characterize communication services required by subsystems
  - Contracts between subsystems and SoC Infrastructure
- Performance of subsystem can be pre-validated based on contract
- SoC infrastructure supports correct integration based on contracts
  - Compositional integration with predictable performance



## SoC Performance Methodology

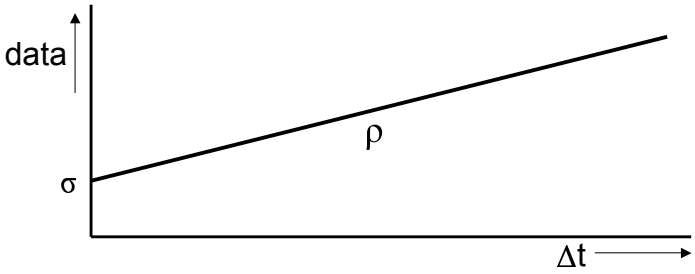



Contracts

- Bounds on traffic flows
- Latency constraints

# Modeling Traffic Flows

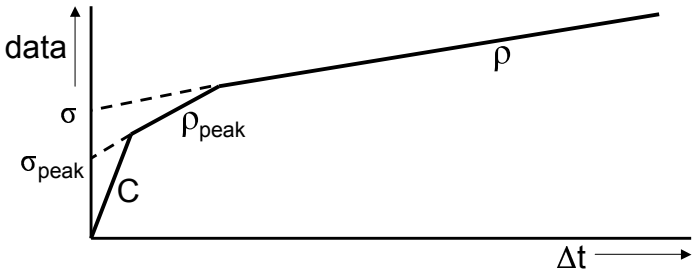
- Express amount of transfered data for sliding window of arbitrary size
- ρ is long term average bandwidth (bytes/s)
- σ is burstiness constraint (bytes)
- For every window of size Δt: data ≤ σ + ρ * Δt





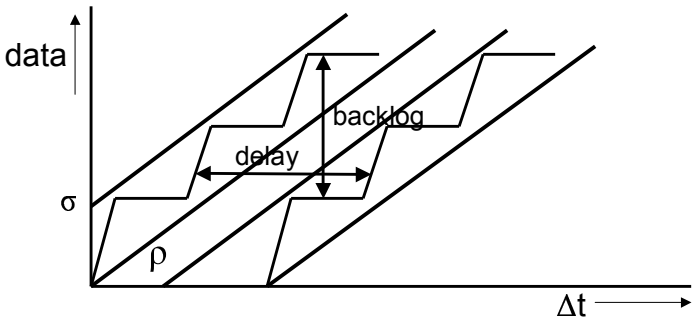

# Modeling Traffic Flows

- C is capacity of link (bytes/s)
- For every Δt: data <= Min (C * Δt, $\sigma_{peak}$ + $\rho_{peak}$ * Δt, σ + ρ * Δt)
- If data is transfered with packets of size L: σ >= L (1 – ρ/C)



# Arrival and Service Curves

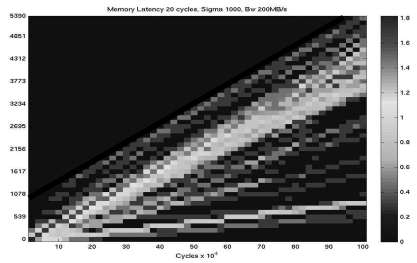
- Processing of flow can be characterized by arrival and service curves
  - An arrival curve describes incoming traffic
  - A service curve describes the outgoing traffic
- Processing will incur delay and backlog
- Backlog translates into buffer requirements






# Characterization of Subsystems

- Aim: black-box characterization
- Derive contracts for which subsystem meets performance requirements




- Sigma-rho pairs can be derived for H.264 decoding
- Trade-offs are possible between sigma and rho
- Low-cost traffic regulation with credit counter
- Automated characterization

# Conclusion

- Key concepts for modular integration of MPSoCs
  - Hierarchical SoC integration style to break complexity barriers
  - Networked infrastructure with standard protocols
  - Decouple subsystem design and subsystem integration
- Industry standards are key enablers
  - MIPI, OpenMAX, …
- Performance methodology
  - For predictable integration of subsystems from different sources
  - Characterization of subsystem requirements
    - Automated black-box characterization to derive required contracts
  - Modeling and analysis of SoC infrastructures
    - Challenge: derive tight bounds on worst-case performance
    - To satisfy performance requirements of subsystems (contracts) without costly over-design of infrastructure