Tiles: the Heterogeneous Processing Abstraction for MPSoC

Drew Wingard
CTO
Sonics, Inc.
wingard@sonicsinc.com

Overview

- ***MPSoC Requirements***
- Design Abstractions
- Tiles
- Tile-based Examples
- Summary

The Goal

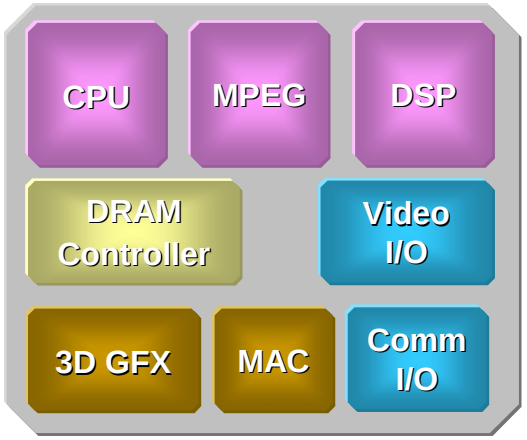
Create a ***system-on-a-chip***, comprising ***ten million gates***, that satisfies ***rapidly-evolving*** market requirements for:

- Speed
- Power
- Area
- Application Performance
- Time to Market

Using the minimum resources with maximum predictability

SoC Architecture Trends

- Massive feature integration
  - Driven largely by Moore's Law (supply) and convergence (demand)
- Continued movement of complexity to software
- Distributed architectures
  - Higher scalability (and independence?)
- Multiple processors
  - CPU
  - DSP
  - Special purpose (MPEG, packet, …)
- Distributed DMA
  - Removes centralized DMA bottleneck
  - Simplifies driver software integration



System On Chip

SONICS™ smart interconnect IP

MPSoC Architecture Options

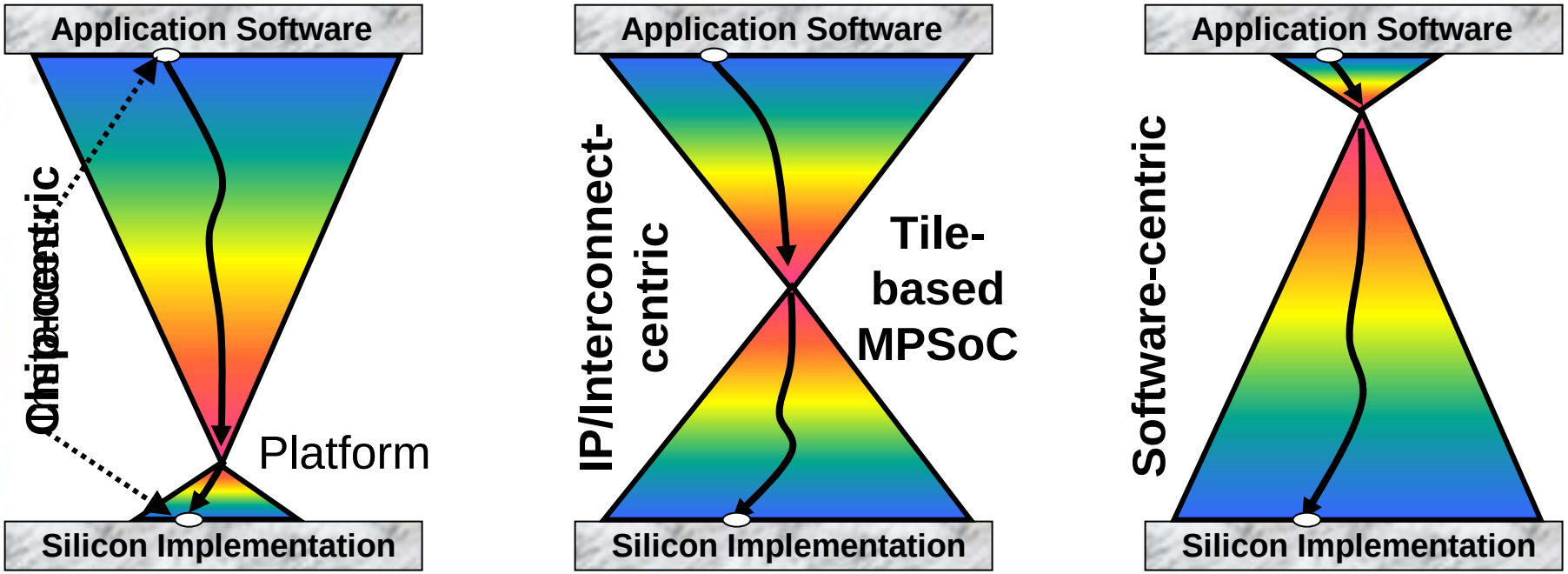
- Multi-processor cluster
  - + Many OS's know how to schedule, good for computing
  - – Hard to scale past about 4 CPU's, bad for real-time
- Uniform distributed computing fabric
  - + Highly scalable, locally efficient
  - – Hard to program/schedule, uniformity hurts QoR
- Distributed heterogeneous processing subsystems
  - + Per-subsystem mix of hardwired and programmable
  - + Highest scalability, high QoR
  - + Divide-and-conquer programming model
  - + Can deploy cluster and/or fabric approaches in subsystems

Tile-based Heterogeneous MPSoC Platforms

- *Tile* – distributed, largely independent subsystem for a SoC, normally composed of:
  - Processing
  - Memory
  - I/O
- Tile processing can be performed in fixed or programmable logic, or general-purpose or special-purpose programmable processors
- Key elements of tile-based platforms:
  - Socket-based design
  - Decoupled interconnect architectures
  - Communication constraint capture
  - Firmware packaging.

What is a Platform?

- Set of choices made at one layer of design that
  - Constrains the design exploration space (above and below)
  - Offers abstraction upward to reduce complexity
  - Enables predictable refinement downward



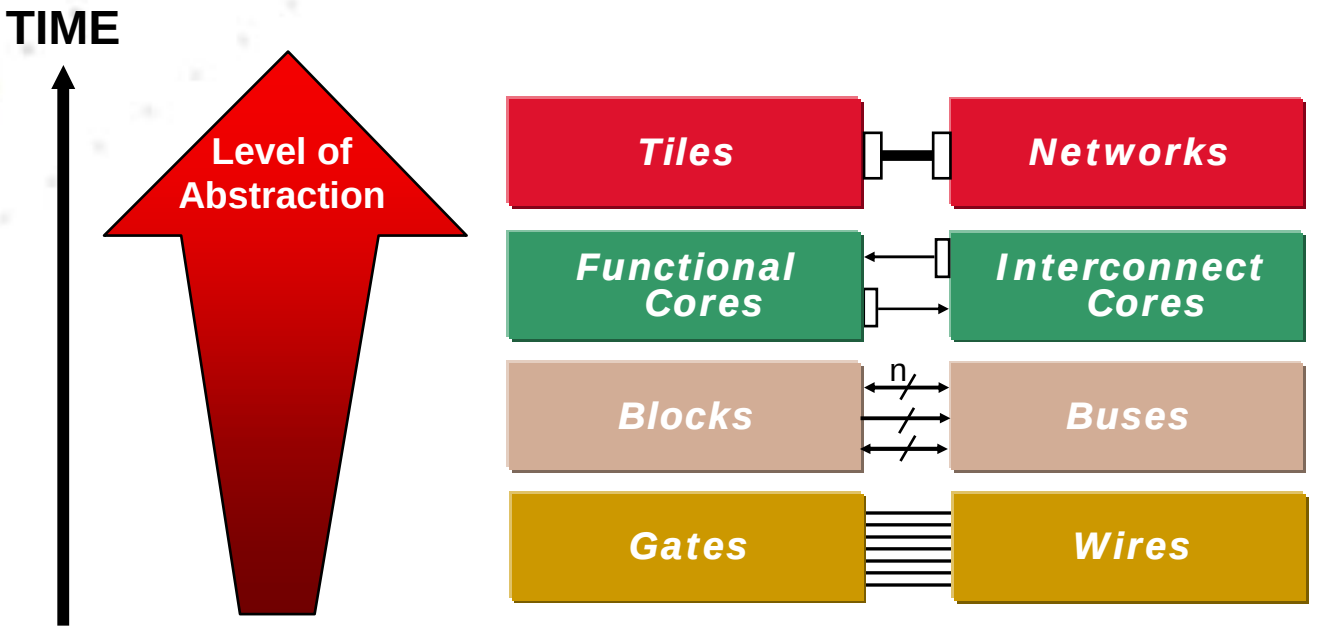
Source: Alberto Sangiovanni-Vincentelli (UC Berkeley)

SONICS™ smart interconnect IP

Overview

- MPSoC Requirements
- ***Design Abstractions***
- Tiles
- Tile-based Examples
- Summary

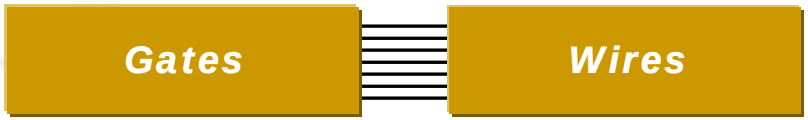
Evolution of Design Abstractions



Abstraction minimizes the number of objects that the system designer must manage

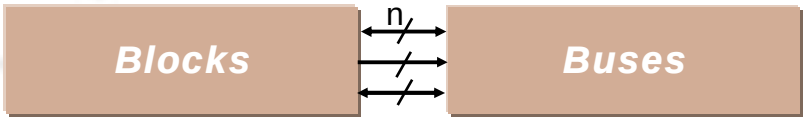
SONICS™ smart interconnect IP

Abstractions: Gates and Wires



- Hides complexity of
  - Transistors, routing layers, contacts, vias
- Functional unit: Gate
  - Simple combinational and sequential elements
  - Placement unit to backend
- Communication unit: Wire
  - Point-to-point and multi-point
  - Routing unit to backend
- Connectivity: Port

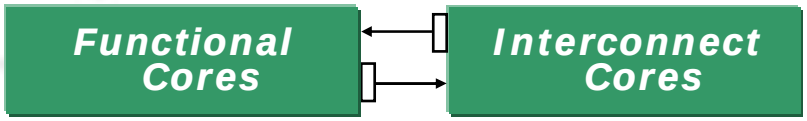
Abstractions: Blocks and Buses



- Hides complexity of
  - Gates and wires
- Functional unit: Block
  - Useful functions, grouped for convenience or replication
- Communication unit: Bus
  - Grouping of wires with selection and multiplexing logic
  - Most communications logic embedded in block, not bus
- Connectivity: Bus interface
  - Almost indistinguishable from bus itself!

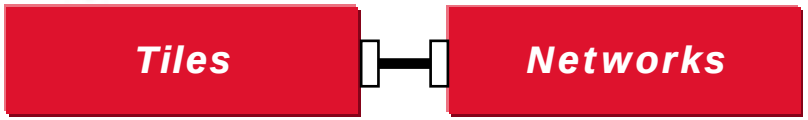
SONICS™ smart interconnect IP

Abstractions: Cores and Interconnects



- Hides complexity of
  - Blocks and buses
- Functional unit: Functional core
  - Implements complete function (e.g. CPU, MPEG2 MC)
- Communication unit: Interconnect core
  - Decouples functional cores
  - Manages communication
- Connectivity: Socket
  - Well-specified boundary of responsibility

Abstractions: Tiles and Networks



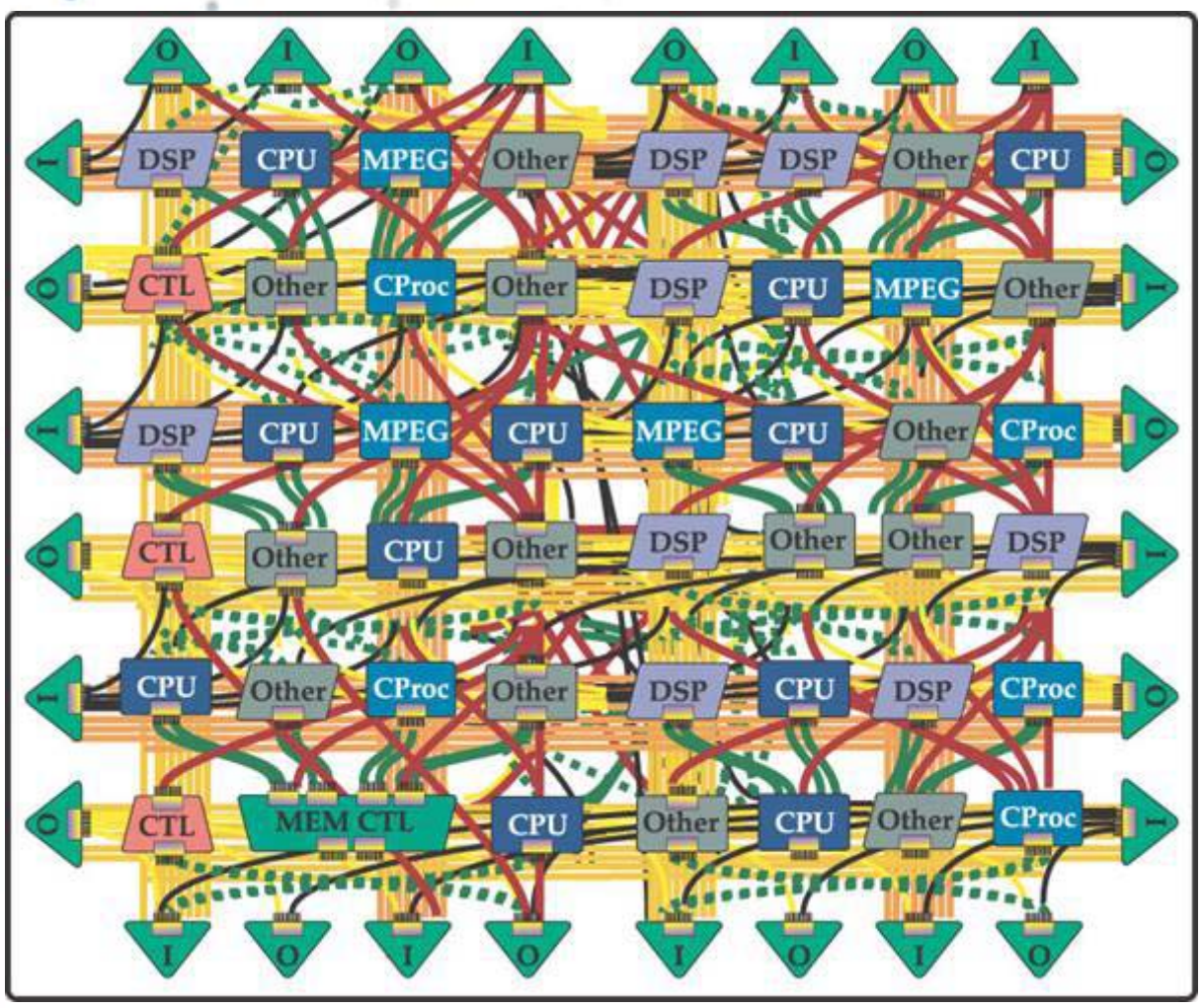
- Hides complexity of
  - Functional and interconnect cores
- Functional unit: Tile
  - Complete subsystem (many with embedded CPU)
- Communication unit: Network
  - Manages messages, not words
- Connectivity: Socket
  - Similar to previous, but more peer-to-peer

SONICS™ smart interconnect IP

Cores and Buses: Mismatched Abstraction

- Industry focused on matching cores and buses
  - E.g. VSIA, ARM AMBA, etc.
- Embedded comm. logic prevents true abstraction
  - Functional cores forced to be “bus aware”
    - Not a “virtual component,” but a very complex block
    - Forces tightly-coupled design
- Result: failure to successfully abstract
  - SOC designers forced to work at too low level
  - Significant challenges scaling to higher integration

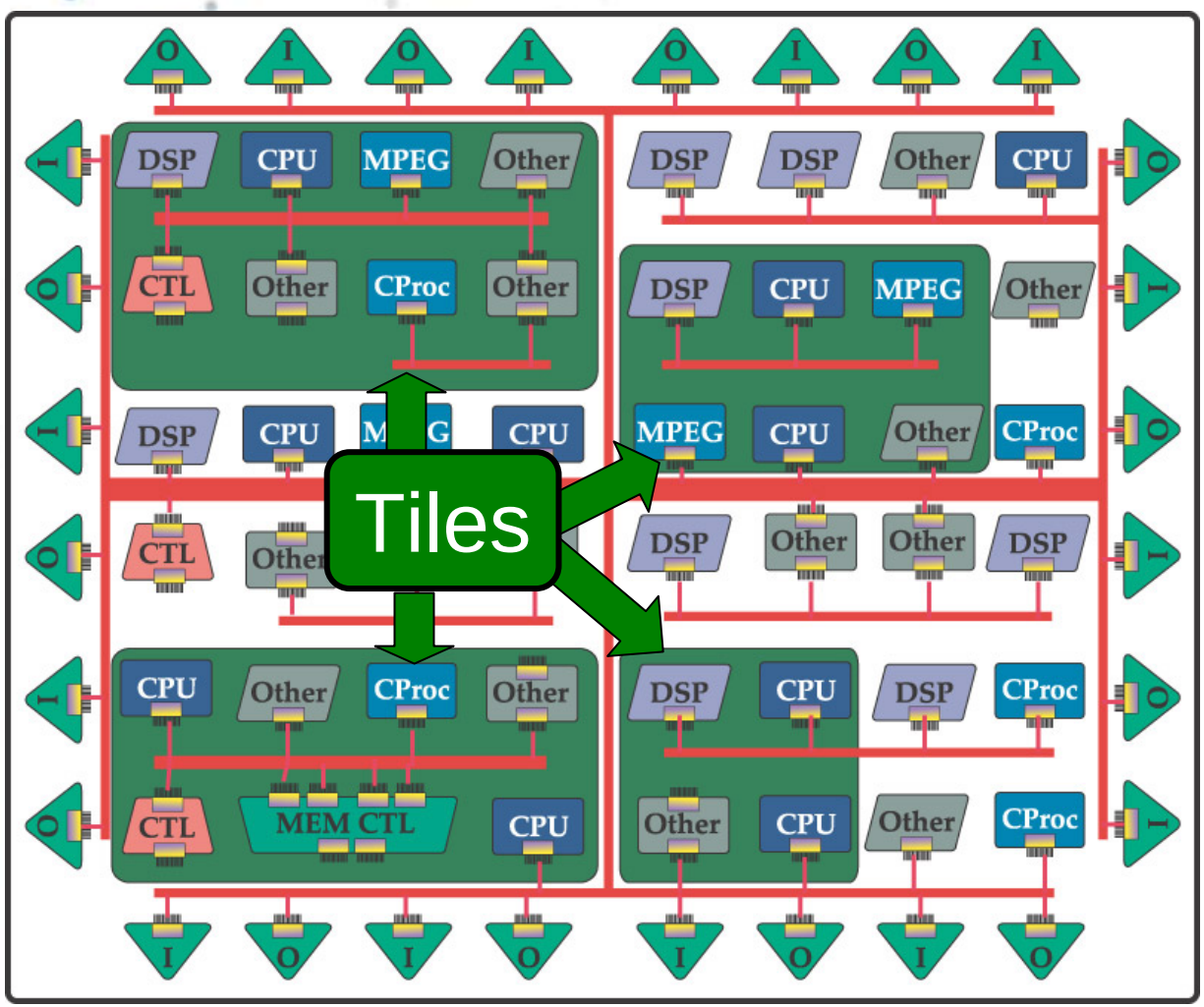
Tightly-coupled MPSoC Example



- Possible today
- Unstructured
- Over constrained
- Nearly impossible to complete

SONICS™ smart interconnect IP

Decoupled Restructuring of Example Leveraging Tiles



- Simpler
- Well structured
- Enables concurrent design
- More predictable
- More scalable
- High re-use

SONICS™ smart interconnect IP

Overview

- MPSoC Requirements
- Design Abstractions
- ***Tiles***
- Tile-based Examples
- Summary

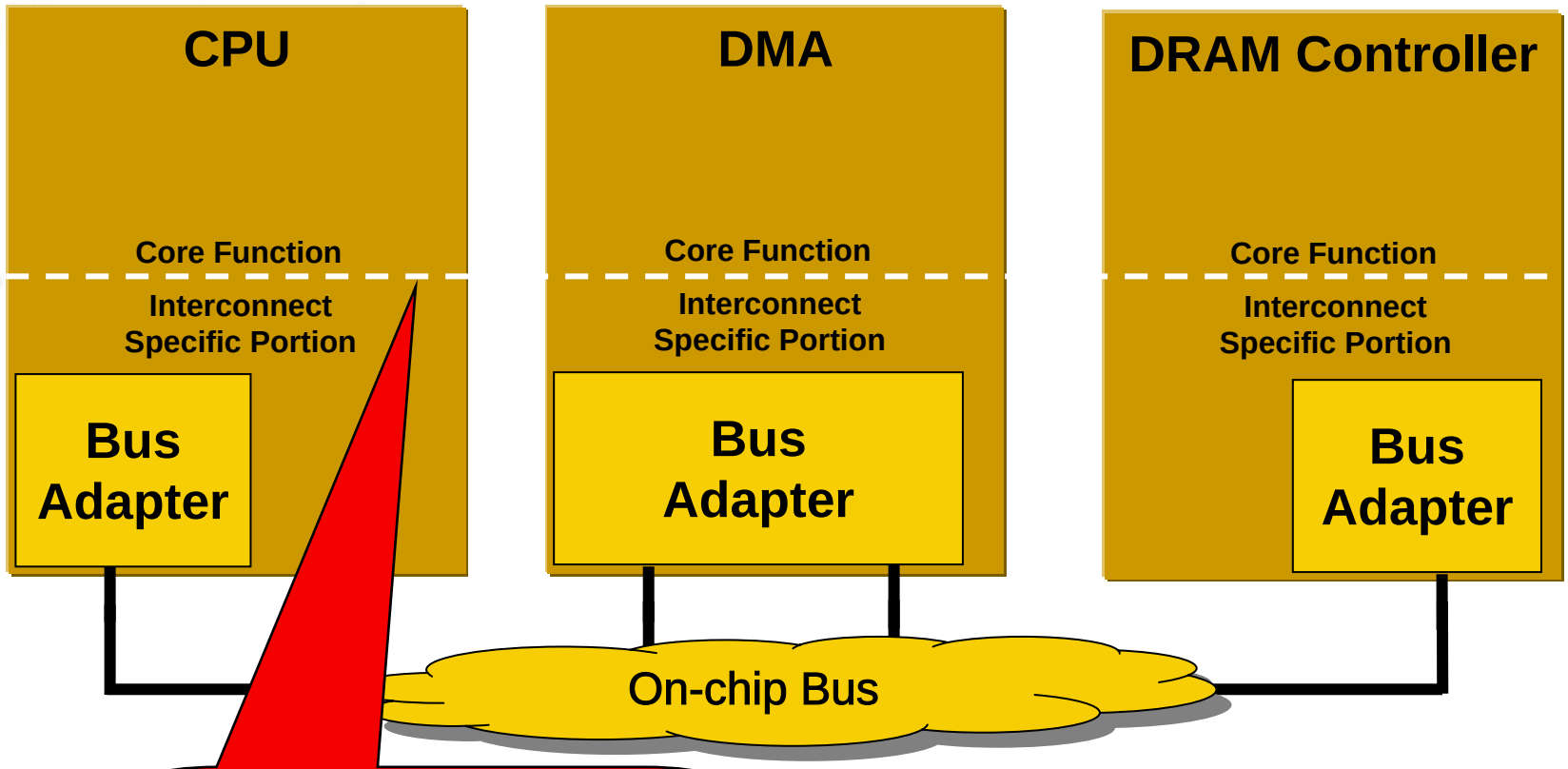
Anatomy of a Tile

- A tile is composed of …
  - Processing elements, memories, and I/O
- … interfaced using standard sockets …
  - Such as OCP
- … connected using decoupled local and global interconnects …
  - Such as Sonics *SMART* Interconnect IP
- … and packaged with the required firmware to make the processors largely self-sufficient

Socket-based Design

- A method of using a standard interface to provide electrical, logical, and functional connections between cores and tiles
- Equally applicable to functional and interconnect cores
- Defines boundary of responsibility
- Enables independent design & verification
- Ensures interoperability

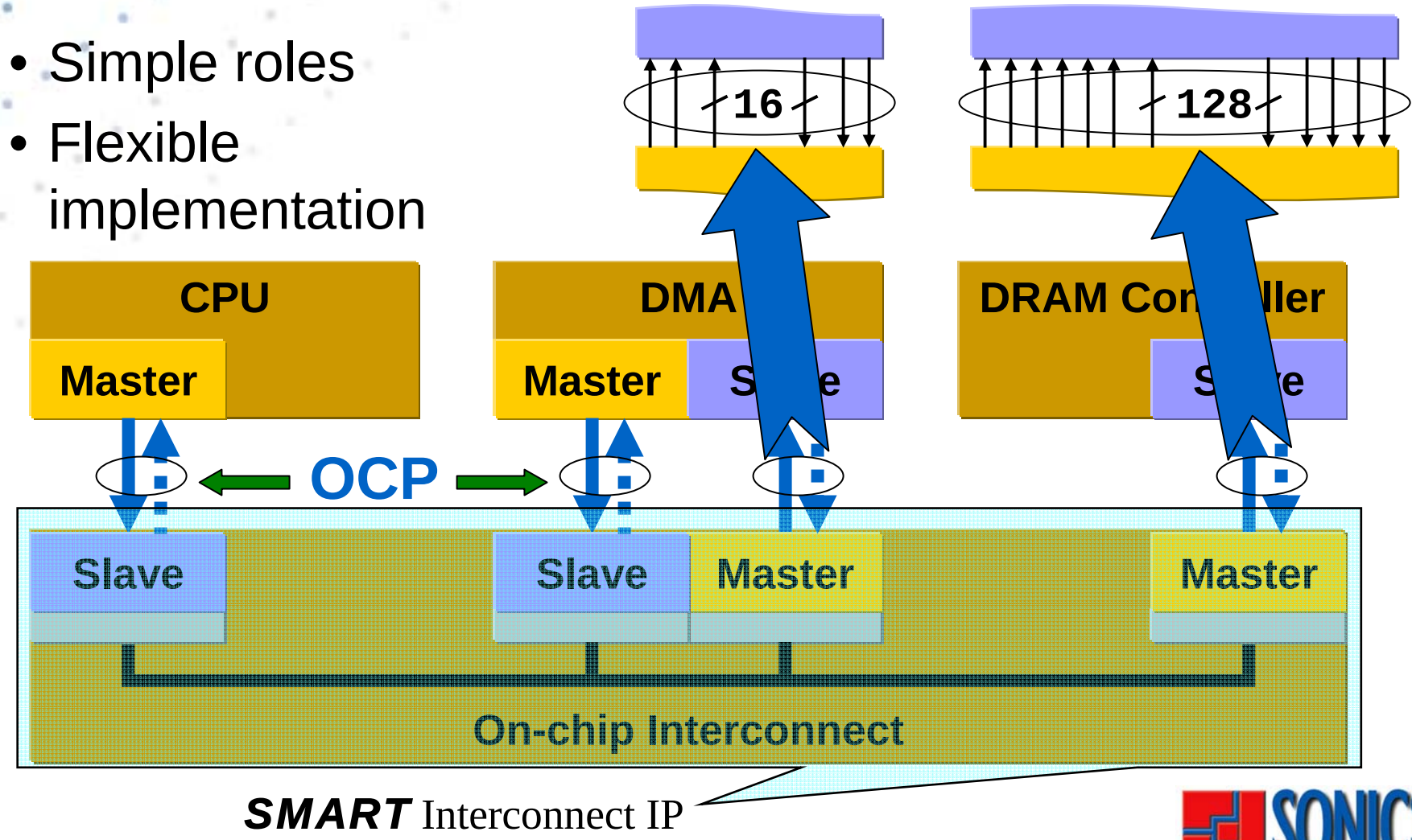
Traditional Partitioning



SONICS™ smart interconnect IP

Socket-based Design Partitioning

- Simple roles
- Flexible implementation



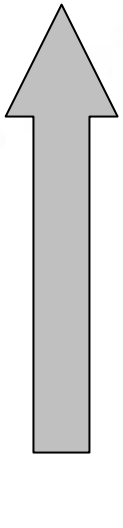
The OCP Socket

- Owned/advanced by OCP-IP (www.ocpip.org)
- Interconnect-neutral
- Defines data flow, control flow, test signaling
- Highly configurable to core needs
- Simple Master/Slave request/response protocols
  - With many options
- Fully synchronous, point-to-point
- Optional pipelining, bursting, threading
- Flexible handshaking and other flow control

Limits of Tightly Coupled Design

- Tightly coupled design has been dominant
  - Assumes largely synchronous, instantaneous and free communication
  - Widely practiced, and supported in design flows
- BUT
  - Delivering clocks is problematic
  - Wire delay is dominant
  - Routing area can cost more than gates
  - Too many constraints, from too many blocks
    - Cannot afford lowest common denominator design

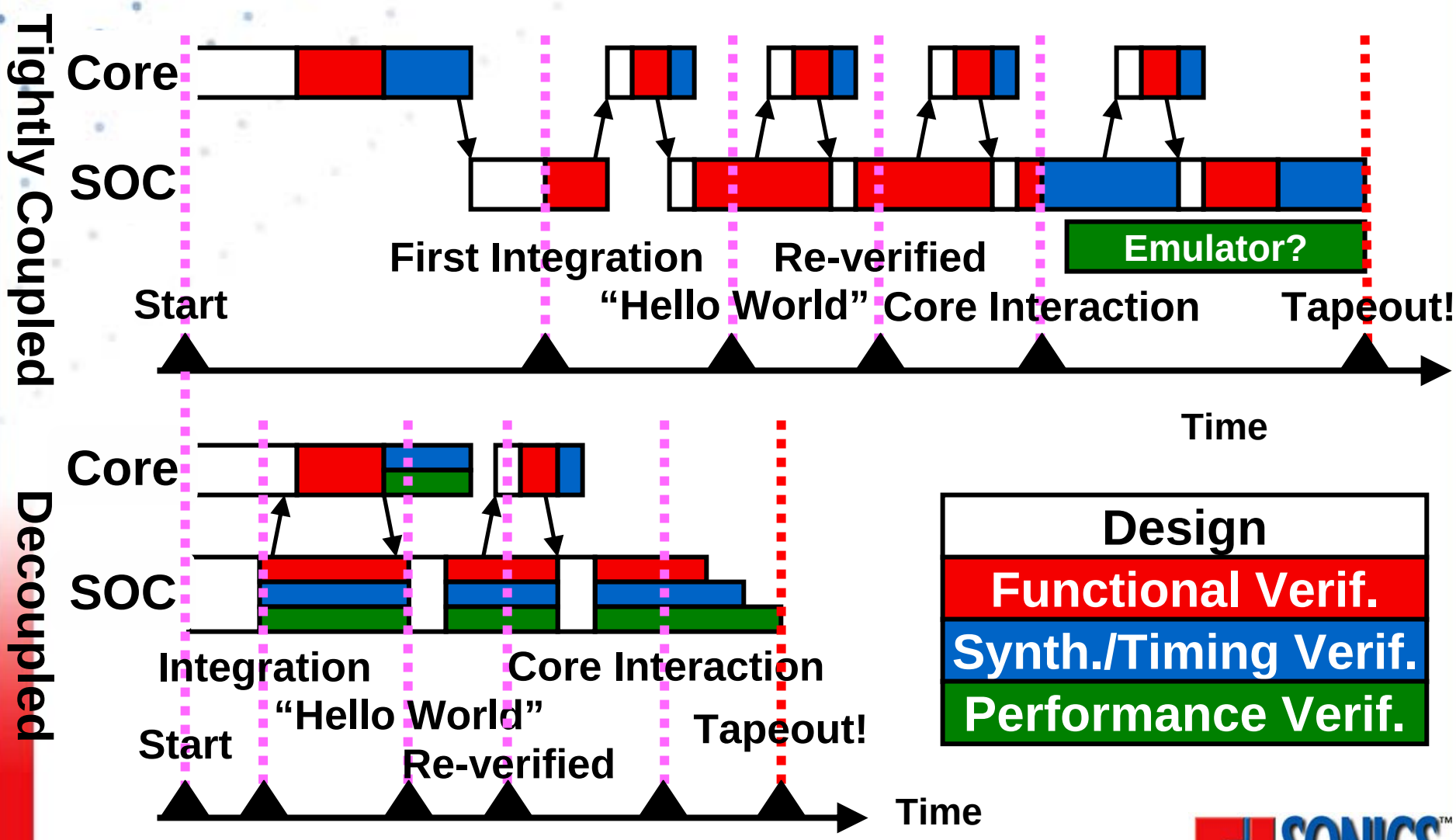
Layers of Decoupling



Performance	• Degree of blocking • QoS
Identification	• Addressing • Source identification
Transfer Protocol	• Bursting, pipelining • Threading
Signaling	• Data, address, event widths • Handshaking / flow control
Electrical	• Signal timing and capacitance • Clock frequency

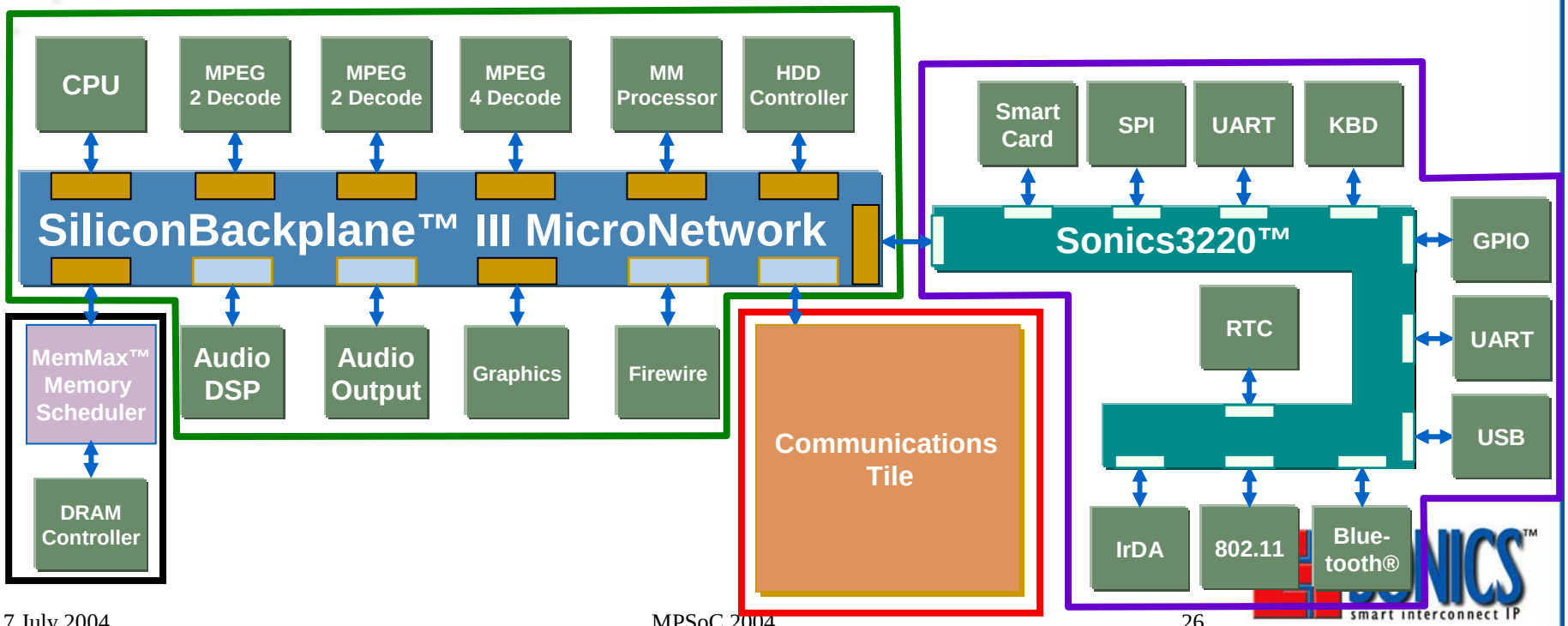
SONICS™ smart interconnect IP

Design Timeline Benefits of Decoupling



Decoupled Active Interconnect IP

- System communications in interconnect, NOT built into cores
- Isolates and protects IP cores
- Offers scalable architectures to optimize power vs. performance
- Delivers highly predictable implementation
- Enables targeting SOC to specific market segments



Sonics' *SMART* Interconnect IP Characteristics

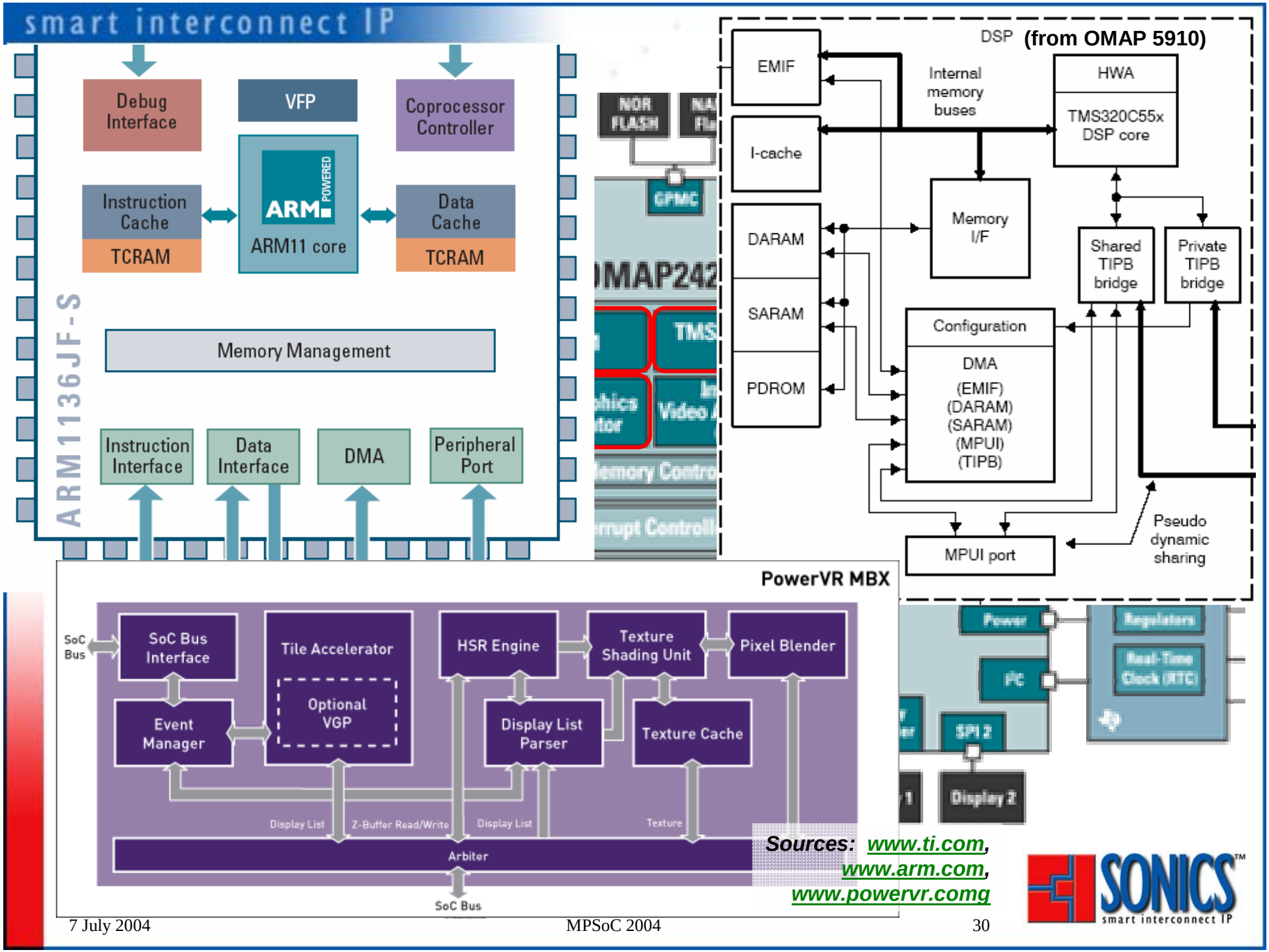
- Scalable pipelining
  - Adapt internal pipeline to frequency, span, loading
- Multi-threaded, non-blocking
  - Prevent slow transactions from impacting fast
  - Enable time interleaving
    - Increase total available bandwidth
    - Increase effective utilization
  - Enable guaranteed QoS
- Distributed multiplexor topology
  - Minimize routing area

Overview

- MPSoC Requirements
- Design Abstractions
- Tiles
- ***Tile-based Examples***
- Summary

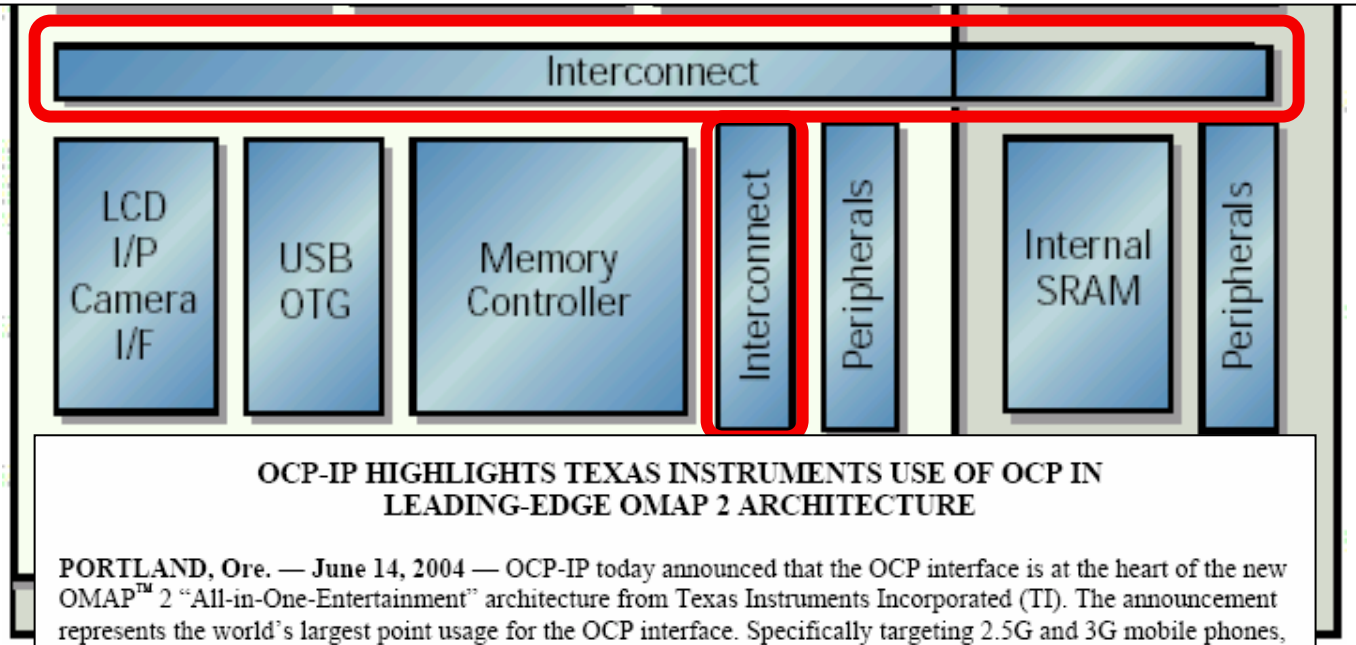
Inside a Wireless Handset SOC Using a Tile-based MPSoC Platform

- Example: TI OMAP 2420
- Heterogeneous processors
  - General-purpose CPU (e.g. ARM 11)
  - DSP (e.g. TI C5x)
  - 2D/3D graphics (e.g. Imagination PowerVR MBX)
  - Video accelerator (e.g. MPEG4 codec)
- Multiple levels of shared memory
  - External DRAM and Flash
  - Internal SRAM and ROM
- Very complex peripheral subsystems
- Security management (content, service, stability)


smart interconnect IP
Debug Interface
VFP
Coprocessor Controller
Instruction Cache
TCRAM
ARM POWERED
ARM11 core
Data Cache
TCRAM
ARM1136JF-S
Memory Management
Instruction Interface
Data Interface
DMA
Peripheral Port
DSP (from OMAP 5910)
EMIF
I-cache
DARAM
SARAM
PDROM
Internal memory buses
HWA
TMS320C55x DSP core
Memory I/F
Shared TIPB bridge
Private TIPB bridge
Configuration
DMA
(EMIF)
(DARAM)
(SARAM)
(MPUI)
(TIPB)
MPUI port
Pseudo dynamic sharing
PowerVR MBX
SoC Bus
SoC Bus Interface
Tile Accelerator
Optional VGP
HSR Engine
Texture Shading Unit
Pixel Blender
Event Manager
Display List Parser
Texture Cache
Display List
Z-Buffer Read/Write
Display List
Texture
Arbiter
SoC Bus
Power
Regulators
I²C
Real-Time Clock (RTC)
SPI 2
Display 2
Sources: www.ti.com, www.arm.com, www.powervr.comg
SONICS
smart interconnect IP

Interconnect-centric Wireless MPSoC

According to Avner Goren, … **the processors have a modular architecture** that reflects changing needs. … in response to the advanced processing and data-shuffling needs of video and graphics Goren said **the company devised a proprietary, low-latency crossbar switch to interconnect the key components**, such as the DSP and memory.



OCP-IP HIGHLIGHTS TEXAS INSTRUMENTS USE OF OCP IN LEADING-EDGE OMAP 2 ARCHITECTURE

PORTLAND, Ore. — June 14, 2004 — OCP-IP today announced that the OCP interface is at the heart of the new OMAP™ 2 "All-in-One-Entertainment" architecture from Texas Instruments Incorporated (TI). The announcement represents the world's largest point usage for the OCP interface. Specifically targeting 2.5G and 3G mobile phones, the OMAP 2 architecture is the second generation of the processors to utilize the OCP interface; previous generations include the OMAP16xx and OMAP17xx product series.

Sources: *www.ti.com, www.commsdesign.com, www.ocpip.org*

SONICS™ smart interconnect IP

Inside a Digital Multimedia SOC Using a Tile-based MPSoC Platform

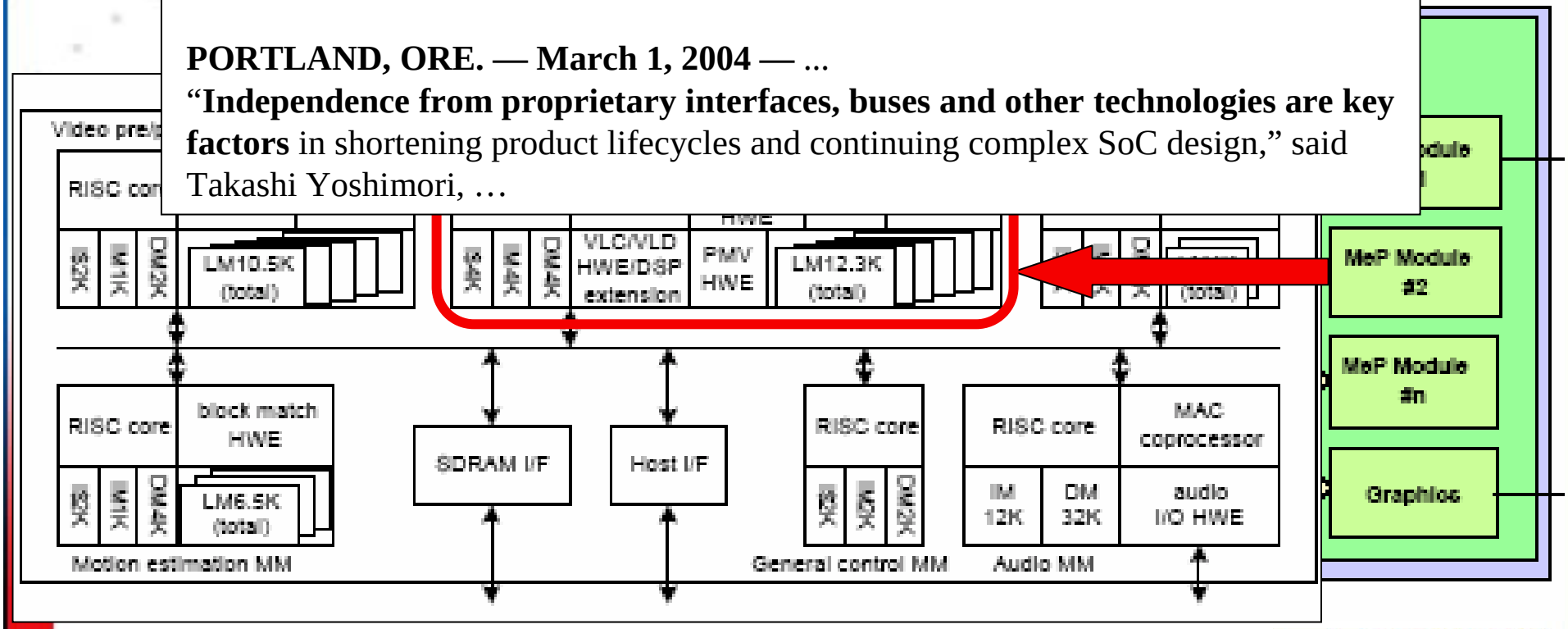
- Example: Toshiba MeP
- Applications: DVD player/recorder, PVR, digital TV
- Heterogeneous processors
  - General-purpose CPU (e.g. TX MIPS)
  - Configurable CPU (e.g. MeP)
    - With many tile-specific enhancements
  - 2D/3D graphics accelerator
- High efficiency requirements from shared memory
  - External SDRAM
- Large changes between sub-applications

MeP Tiles

Toshiba Joins OCP-IP

New Governing Steering Committee Member Further Enhances OCP-IP Working Groups and Increases Organization's Prominence in Asia

PORTLAND, ORE. — March 1, 2004 — ...
"**Independence from proprietary interfaces, buses and other technologies are key factors** in shortening product lifecycles and continuing complex SoC design," said Takashi Yoshimori, …



Sources: *www.mepcore.com, www.ocpip.org*

SONICS™ smart interconnect IP

Summary

- New abstractions needed to manage MPSoC complexity
- Should adopt Tile-based design model
- Socket-based design enables concurrent design
- Active interconnect cores offer
  - Decoupling at multiple layers
  - Independent optimization of SOC from cores
  - Rapid and predictable design
- Tile-based design is proven today

Acknowledgements

- Entire ***SMART*** Interconnect IP development team at Sonics
- Fellow members of OCP-IP