Communication Synthesis and System-Level Design Exploration

Luca Carloni
Department of Computer Science
Columbia University in the City of New York
Outline

1. Trends in High-Performance Embedded Computing
   • Common challenges in the design of distributed embedded systems and multi-core systems-on-chip

2. Correct-by-Construction Component-Based Design
   • Revisiting LID: The Protocol & Shell Paradigm
   • Communication Synthesis
   • Synthesis-Enabled Design Exploration

3. Some Open Research Challenges
Trends in Consumer Electronics: Broader, Richer Connectivity

- **2011**
  - 100M households watching internet video on TV

- **2012**
  - 1.2B Mobile Internet Users
  - 1.4B subscribers for 3G/4G/WiMAX

Source: www.intel.com

*Source: Intel, ABI Research, 3GPP RAN1, In-Stat*
Evolution of Cellular Phones

- Source: Y. Neuvo, "Cellular phones as Embedded Systems", ISSCC 2004
Inside a State-of-the-Art Mobile Phone: the Apple I-Phone 3G
Multi-Core Systems-on-Chip for High-Performance Embedded Applications

- **New Chips**
  - feature more parallel and distributed architectures
- **New Challenges**
  - steady growth in design complexity, component integration
  - fixed power budget
  - uncertainties of nanometer device behaviors
  - bus-based communication does not scale
  - I/O communication bottleneck
  - tighter hardware-software interaction

Luca Carloni – Columbia University

Source: www.intel.com
From a Concurrent Specification to a Distributed Implementation on a Parallel Computational Platform

Multi-Core Systems-on-Chip
- distributed
- heterogeneous
- concurrent
- component reuse, assembly
- design methodology based on the synchronous assumption

Networked Embedded Systems
- distributed nature of applications challenges traditional design methodologies
  - drive-by-wire, fly-by-wire
  - building automation
- large variations in computation/communication times
- hard to maintain a global notion of time
Component-Based Design and Synchronous Model of Computation

RTL Design separates functional specification from performance analysis.
Once all modules are composed, the overall system works correctly as far as it is running with a clock period $T_{clk} = \max \{T_1, T_2, T_3, T_4\}$.
Impact of Inter-Module Path Delays

Nanometer technologies force us to reconsider the delay of global interconnect:

\[ T_{\text{clk}} = \max \{ T_1, T_2, T_3, T_4, D_a, D_b, D_c, D_d, D_e, D_f, D_g \} \]
The Theory of Latency Insensitive Protocols

Latency equivalence:
same data stream, different timing

\[ S = \ldots a \ldots b \ldots c \ldots d \ldots \]

\[ S' = \ldots a \cdot b \ldots c \ldots d \ldots \]

"Strict System"

"Patient System"

Luca Carloni - Columbia University
Compositionality

- For patient processes the notion of latency equivalence is compositional
  
  - Th.1: $P_1$ and $P_2$ patient $\Rightarrow P_1 \cap P_2$ patient
  
  - Th.2: for all patient $P_1, Q_1, P_2, Q_2$
    $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$ $\Rightarrow (P_1 \cap P_2) \equiv (Q_1 \cap Q_2)$
  
  - Th.3: for all strict $P_1, P_2$ and patient $Q_1, Q_2$
    $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$ $\Rightarrow (P_1 \cap P_2) \equiv (Q_1 \cap Q_2)$

- Major Theoretical Result
  
  - if all processes in a strict system are replaced by corresponding patient processes then the resulting system is latency equivalent to the original one
The Protocol & Shell Paradigm

- Pearls (synchronous IP cores)
- Shells (interface logic blocks)
- Channels (short wires)
- Channels (long wires)
Channel Segmentation (Wire Pipelining)

Pearls (synchronous IP cores)
Shells (interface logic blocks)
Channels (short wires)
Channels (long wires)
The Protocol & Shells Paradigm and The Role of Synthesis

- preserves modularity of synchronous assumption in distributed environment
- guarantees scalability of global property by construction and through synthesis
- simplifies integrated design & validation by decoupling communication and computation, thus enabling reusability
- adds design flexibility up to late stages of the design process
The Role of Communication in a System-Level Design Automation Flow

- Computation synthesis includes HW design with soft/hard IP.

**ESL Specification**

**ESL Design**
- Computation Synthesis
- Interface Synthesis
- Communication Synthesis

**Flexible, Synthesizable RTL Representation**

**Logic and Physical Design**
- Logic Synthesis
- Placement
- Routing

**Physical Implementation**

**Full-Chip Analysis**
- Floorplanning and Wireplanning
- Estimations for Area, Power, Performance...
- Interconnect Timing/Power Analysis

Diagram elements:
- soft IPs
- comm. lib
- hard IPs
- program. blocks
- module views
- block views
- activity inputs
- parasitics

Luca Carloni – Columbia University
Using Local Intra-Core Information to Increase Global Performance in Block-Based SoC Design

- **New Adaptive-Shell design**
  - can exploit dynamically the internal characteristics of a white-box core
  - fully compatible with “classic latency-insensitive protocol”

- **New Synthesis Algorithm**
  - new shells automatically synthesized only around critical cores in the SoC

- **Application to SoC design**
  - COFDM-based ultrawide band transmitter
  - processing throughput improvement up to 30%
  - area overhead between 1 and 3%
  - no impact on critical path delay

C. Li and L.P. Carloni, “Leveraging Local Intra-Core Information to Increase Global Performance in Block-Based Design of Systems-on-Chip”, IEEE TCAD (to appear)
Constraint-Driven Communication Synthesis

[Joint work with Alessandro Pinto and Alberto Sangiovanni-Vincentelli]
**Constraint-Driven Communication Synthesis (1)**

- **Input Constraints**
  - set of IP cores
    - area, interface, possibly position constraints
  - end-to-end communication requirements
    - latency, throughput

[ Courtesy: A. Pinto ]
Constraint-Driven Communication Synthesis (2)

- **Library of Components**
  - interface types, max number of ports
  - max capacities: bandwidth, latency, max distance
  - performance and cost model

[Diagram showing components and performance metrics]

Dynamic Energy per Flit (J)

Energy per flit: 35.2pJ
Leakage @ 1GHz: 5.1mW
Area: 3148μm²

Energy per flit: 8.2pJ
Leakage @ 1GHz: 0.85mW
Area: 598μm²

[Courtesy: A. Pinto]
Constraint-Driven Communication Synthesis (3)

- **Instantiation and Composition**
  - renaming of components
  - quantity and type constraint satisfaction

---

Luca Carloni – Columbia University

[ Courtesy: A. Pinto ]
Heterogeneous Parallel Composition of Network Specifications

Luca Carloni – Columbia University

[ Courtesy: A. Pinto ]
Constraint-Driven Communication Synthesis (4)

- Synthesis
  - return valid composition that meets constraints and
  - minimizes objective function (power dissipation)
**COSI: The Communication Synthesis Infrastructure**

On-Chip Communication Design Exploration and Synthesis with COSI

Luca Carloni – Columbia University

[Courtesy: A. Pinto]
Application:
Real-Time Estimation of Building Occupancy

• Improves operations of HVAC systems
  - reduced energy costs
  - raise comfort level for building occupants

• Improves emergency evacuation procedures
  - interviews with fire departments confirm importance of having real-time information on people location in a building during emergencies

3.3 Trillion KWh

Source: C. Jacobson (UTRC)
Application: Real-Time Estimation of Building Occupancy

- **State**
  - people occupancy in various areas of the building

- **Input**
  - building layout, partitioned in the areas of interest
  - location of fixed-position sensors
  - sensor capabilities
    - range of detection, accuracy
  - operation modes
    - normal vs. emergency egress

- **Output**
  - probability distribution of the number of people located in each area of the building

Source: Tomastik et al. '08
In a centralized implementation each digital video camera sends its stream of data to a central server running the video analysis application.
System-Level Task Decomposition

- System-level task decomposition
  - allows identification of key computation and communication requirements
  - drives selection of COTS components to build implementation and design exploration process

- For a video analysis application
  - the more tasks are executed locally the lower the bandwidth requirements that a videonode imposes on the network

Luca Carloni – Columbia University
Symmetry of Application Data Flow
Alternative Video-Node Embedded Platforms: a) RTCC

- **Raw Transmission Central Computation (RTCC)**
  - Embedded Computation
    - microcontroller implements the communication stack and oversees the video sensor
    - device choice depends on network technology
  - Communication
    - highest possible BW requirement
    - 2MB/s for our application
    - this rules out the utilization of ARCnet with the EIA-485 physical layer when the system consists of more than one... video node
  - Memory
    - no frame buffer required (w/ assumption that channel is always available)
Alternative Video-Node Embedded Platforms: 
b) DMCC

- Distributed Motion Detection and Central Computation (DMCC)
  
  - **Embedded Computation**
    - microcontroller implements the communication stack and oversees the video sensor, and process the signal from a Pyroelectric Infrared Sensor (PIR)
  
  - **Communication**
    - transmission rate is no more constant, instead average payload depends on the probability of detecting a motion event
      - ARCnet is now a possibility
  
  - **Memory**
    - frame buffer is necessary,
      - e.g. 64Mb of SDRAM
Alternative Video-Node Embedded Platforms: c) PDC

- Embedded Computation
  - videonode detects blobs in each frame and forwards some of their features to the next processing unit
  - Atmel microcontroller with 200Mhz ARM9 and support for Ethernet (MAC) and CAN protocols

- Communication
  - only few bits are transmitted for each frame for an overall BW requirement of a few Kb/s

- Memory
  - FLASH memory necessary to store embedded SW
    - 64Mb of SDRAM + 1Gb of FLASH
Alternative Video-Node Embedded Platforms: d) CDC

- **Completely Distributed Computation (CDC)**
  - Embedded Computation
    - videonode does full video processing task
      - e.g., 400Mhz Analog Device Blackfin DSP
      - device choice depends on network technology
  - Communication
    - small BW requirements, no a priori restriction on communication standards
  - Memory
    - More memory necessary for more complex algorithm
      - 128Mb of SDRAM + 1Gb of FLASH
Candidate Network Technologies

- **ARCnet**
  - widely adopted in building automation systems
  - low installation costs, high flexibility and predictability
  - **ARCnet@78kbps or ARCnet@2.5Mbps**
    - they have the same wiring costs, but there can be at most 32 stations per chain in the former, and 8 in the latter
    - multiple chains can be connected via routers

- **Two-Tiered Switched Ethernet**
  - more expensive but higher bandwidth
  - potentially better suited for video streaming applications

- **Two-Tiered Hybrid Network**
  - **ARCnet@78kbps or ARCnet@2.5Mbps** combined with Ethernet that implements the second tier of the network
Four Alternative Application Mappings for the Case of 15 Video-Nodes and 1 Front Panel

(a) 12 RTCC and 3 CH nodes interconnected with an Ethernet bus

(b) 13 PDC1 and 2 CH nodes interconnected with an Ethernet bus

(c) 14 PDC2 and 1 CH nodes interconnected with an Ethernet bus

(d) 15 CDC nodes interconnected with an ARChnet bus

Luca Carloni – Columbia University
Experimental Results: Implementation Costs after Network Synthesis
Some Open Research Challenges

- to model complex heterogeneous components
  • such as multi-core processors, standard bus/networks, wireless protocols
- to cope with large data-dependent delay variations in communication/computation times
  • typical of implementations with programmable cores
- to guarantee robustness with respect to faults and errors in the execution platforms
  • both in processing cores and communication network
- to cope with dynamic system reconfiguration
  • “load rebalancing” of comm/comp resources
  • fine-grain temporary power-down of components