The Wireless System Design Challenge

The Battery Limitation

- Projected energy per digital operation (2004): 50 pJ
- Lithium-Ion: 220 Watt-hours/kg == 800 Joules/gr
- At 50 pJ/operation: 10 teraOps/gr!
  - Equivalent to continuous operation at 100 MOPS for 30 hours (or average power dissipation of 6 mW)

The Changing Metrics

Flexibility
Power
Cost

Performance as a Functionality Constraint
("Just-in-Time Computing")
The Wireless Challenge

The Software Radio

- Idea: Digitize (wideband) signal at antenna and use signal processing to extract desired signal
- Leverages of advances in technology, circuit design, and signal processing
- Software solution enables flexibility and adaptivity, but at huge price in power and cost
- 16 bit A/D converter at 2.2 GHz dissipates 1 to 10 W
The Mostly Digital Radio

Analog
\[ \cos[2\pi(2\text{GHz})t] \]

Digital
\[ \sin[2\pi(2\text{GHz})t] \]

RF input \((f_c = 2\text{GHz})\)

LNA

RF filter

chip boundary

A/D

Digital Baseband Receiver

I (50MS/s)

Q (50MS/s)

Architectural Choices

Dedicated Logic

Direct Mapped Hardware

Hardware Reconfigurable Processor

Software Programmable DSP

Hardware

\(\mu\text{P} \)

Satellite Processor

MAC Unit

Addr Gen

Prod Mem

Prod Mem

General Purpose \(\mu\text{P}\)

1/Efficiency
(Re)configurable Computing: Merging Efficiency and Versatility

Spatially programmed connection of processing elements.

\[ y = Ax^2 + Bx + C \]

“Hardware” customized to specifics of problem.

Direct map of problem specific dataflow, control.

Circuits “adapted” as problem requirements change.
The Implementation Opportunity

The Radio-on-a-Chip Design Problem

- DSP and control intensive
- Mixed-mode
- Combines programmable, flexible, and application-specific modules
- Cost and energy are the key metrics

The Radio-on-a-Chip Design Problem

- Multiple levels of design optimization
  - The “fractal nature” of design
- Capturing the functionality
- Capturing the architectural choices
- Quantifying the exploration trade-off’s
System Optimization Hierarchy

Digital Intercom — A Design Exercise in Communication/Component Based Design

- Known and tested specification of limited complexity allows focus on architectural implementation methodology
- Two-chip implementation leverages separates between analog (RF) and digital design concerns

Basestation

Mobiles

Up to 20 users per cell @ 64 kbit/sec per link
TDMA selected as MAC protocol
**Two-Chip Intercom (TCI)**

- **Chip 1**
  - Custom analog circuitry
- **Chip 2**
  - Mixed analog/digital
  - Fixed logic
  - Programmable logic
  - Software running on processor

Direct down-conversion front-end
(Yee et al)

**System Level Design with Embedded Platforms**

**Separation of Digital Communications and Protocol Processing**

- **Digital Baseband Processing**
  - **Protocol**

- **MATLAB**
- **VCC**

- **TX**
- **TX_CLK**
- **TX_DATA**
- **RX**
- **RX_CLK**
- **RX_DATA**
- **ON/OFF**

- Physical to Protocol Interface Signals

- Different tool environments require up-front partitioning
- Interface design critical to ensuring final designs work together
- Small number of interface signals
  - Clearly specified behavior and constraints
- Verification relying on co-simulation

**DAC**

Tutorial
Digital Baseband (receiver)

Functional Description:
Simulink + Stateflow (The Mathworks)

Direct Mapping – an obvious choice for high-performance data-flow

Design Estimations (First order)
From Simulink Schematic:
RF + ADC/DAC
Transmit: 30 mW
Receive: 70 mW
Digital (conservative)
Transmit: 20 mW (100,000 transistors)
Receive: 80 mW (700,000 transistors)

UC Berkeley IMake Flow (Brodersen)
Performance Analysis of Baseband Processing
Produces Timing Constraints for Protocol Design

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Additional Calculations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>0.10s</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>1.0s</td>
<td></td>
</tr>
<tr>
<td>Bit</td>
<td>1.0s</td>
<td></td>
</tr>
</tbody>
</table>

Additional Calculations:

- Microsoft Excel

Exploring the Protocol Design Space

1. Verify Behavior
2. Algorithm Design and Exploration
3. Capture Behavior
4. Hardware-Software Mapping
5. Performance Estimation
6. Refine Communication
7. Architecture Libraries
8. Capture Architecture
9. Verify Architecture
10. Verify Behavior

Cadence VCC Environment

System Level Design with Embedded Platforms

Tool: Microsoft Excel
**The Intercom Protocol Stack**

- **User Interface Layer (UI)**: Service Requests, Voice samples
- **Transport Layer**: Service Requests, Voice samples
- **Mac Layer**: Voice samples, Service Requests
- **Data Link Layer**: Voice samples

**Describing the Behavior**

<table>
<thead>
<tr>
<th>Layer</th>
<th>C-code (lines)</th>
<th>State-transition diagrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Mulaw</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>270</td>
<td>10</td>
</tr>
<tr>
<td>Transmit</td>
<td>120</td>
<td>3</td>
</tr>
<tr>
<td>Receive</td>
<td>140</td>
<td>2</td>
</tr>
<tr>
<td>Synchronization</td>
<td>140</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>1030</td>
<td>18 (80)</td>
</tr>
</tbody>
</table>

Model of computation: Co-design Finite State Machines (CFSMs).
Formal Specification enables Verification

- **Does system satisfy certain properties?**
  - System described in some formal mathematical languages (e.g. Esterel, CFSM)
  - Properties written in some formal logic (e.g. Temporal Logic) or formal model (e.g. Esterel, CFSM)

- **Two approaches**
  - Property Verification
    - Invariant (only one remote can send voice data in any time slot)
    - Response (if a remote sends a request to the base station, then eventually there is an acknowledgement)
    - deadlock freedom
  - Refinement Checking
    - Does the (low-level) implementation conform with the (high-level) specification?
      (Do the mapped CFSMs function the same as the specification?)

- **Example: Mocha System (Henzinger, UCB)**

---

Example of Property Verification

Remote returns to the disconnect state if user presses the disconnect button.

\[ AG (\text{Disc} \rightarrow \text{AF (Not Conn)}) \]

× NOT OK
Why it Fails?

- Remote accepts Disc from the user even if it is not connected
- After the remote has sent DiscReq and waits for acknowledgement
- However, base station ignores DiscReq if remote is not registered
  ➔ Deadlock!

Targeted Implementation Platform

- Embedded Processor
- Memory Sub-system
- Interconnect Network
- Baseband Processing
- Fixed Protocol Stack
- Programmable Protocol Stack
The Architecture Description in VCC

ASIC/FPGA  SiliconBackplane  Tensilica Xtensa

Modeling the Architectural Components

The embedded processor

- Xtensa embedded CPU (Tensilica, Inc)
  - Configurability allows designer to keep "minimal" hardware overhead
  - ISA (compatible with 32 bit RISC) can be extended for software optimizations
  - Fully synthesizable
  - Complete HW/SW suite
- VCC modeling for exploration
  - Requires mapping of "fuzzy" instructions of VCC processor model to real ISA
  - Requires multiple models depending on memory configuration
  - ISS simulation to validate accuracy of model

Tensilica model in VCC

The “fuzzy” instruction set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Instruction</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>inst.LD,2</td>
<td>inst.MUL.c,9</td>
<td>inst.DIV.l,118</td>
</tr>
<tr>
<td>inst.LL,1</td>
<td>inst.MUL.s,10</td>
<td>inst.DIV.l,122</td>
</tr>
<tr>
<td>inst.ST,2</td>
<td>inst.MUL.i,18</td>
<td>inst.DIV.d,146</td>
</tr>
<tr>
<td>inst.OP.c,2</td>
<td>inst.MUL.l,22</td>
<td>inst.DIV.d,155</td>
</tr>
<tr>
<td>inst.OP.s,3</td>
<td>inst.MUL.i,45</td>
<td>inst.JF,5</td>
</tr>
<tr>
<td>inst.OP.l,1</td>
<td>inst.MUL.d,55</td>
<td>inst.GOTO,2</td>
</tr>
<tr>
<td>inst.OP.l,1</td>
<td>inst.DIV.c,19</td>
<td>inst.SUB,19</td>
</tr>
<tr>
<td>inst.OP.l,1</td>
<td>inst.DIV.s,110</td>
<td>inst.RET,21</td>
</tr>
<tr>
<td>inst.OP.d,6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"The Silicon Backplane" (Sonics, Inc)

Flexible bandwidth arbitration model
- TDMA slot map gives slot owner right of refusal
- Unowned/unused slots fall to round-robin arbitration
- Latency after slice granted is user-specified between 2-7 Bus Clock cycles
Exploring Architectural Mappings

Processor Utilization - Estimation

- Processor Utilization
- Clock Frequency
- Peak performance
- Latency insensitive
- RTOS overhead
System Optimization Hierarchy

Network Level

Radio Node Level

Module Level

Functional & Performance Requirements
Network Architecture
Performance analysis

Constraints

Implementation Fabrics for Protocols

A protocol = Extended FSM

A module

Slot_Set_Tbl 2x16

Intercom TDMA MAC

DAC  with Embedded Platforms
Intercom TDMA MAC
Implementation alternatives

<table>
<thead>
<tr>
<th></th>
<th>ASIC</th>
<th>FPGA</th>
<th>ARM8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>0.26mW</td>
<td>2.1mW</td>
<td>114mW</td>
</tr>
</tbody>
</table>

ASIC: 1V, 0.25 μm CMOS process
FPGA: 1.5 V 0.25 μm CMOS low-energy FPGA
ARM8: 1 V 25 MHz processor; n = 13,000
Ratio: 1 - 8 - >> 400

How much flexibility is really needed?

HW Mapping Experiment: Flexible Implementation

Area Comparison - FPGA x PLD (Manual)
**HW Mapping Experiment: Power Consumption**

**FPGA versus PLD**

![Power Consumption Chart]

**HW Mapping Experiment: Flexible versus Fixed**

**Area Comparison – FPGA x Std.Cell (Manual)**

![Area Comparison Chart]
**TCI Exploration Summary**

- Exploration at chip micro-architecture level enables dramatic improvements in various metrics – results often non-intuitive
- Obtaining this insight using other techniques is either extremely time-consuming, or naïve.
- Exploration tools such as VCC require a re-schooling of the designer – new abstractions at function and architecture level, new perspective on “accuracy” of predictions
  - Yet learning curve is quite flat
  - Took graduate students approximately 1 month to get fully experienced with tools
  - Once descriptions and models are available, multiple options can be explored in a single day

**The Fractal Nature of Design (again)**

![Diagram showing the fractal nature of design with levels and constraints](image-url)
Picoradio’s: Wireless Networks of Ubiquitous Sensors and Monitors

Example - The Smart Home and Network Appliances

Other applications:
Office buildings, toys, Interactive museums

Security
Environment monitoring and control
Object tagging
Identification

System Requirements and Constraints

• Large numbers of nodes — between 0.05 and 1 nodes/m²
• Cheap (<0.5$) and small (< 1 cm³) and ultra low-power (< 100 µW) enabling energy scavenging
• Limited operation range of network — maximum 50-100 m
• Low data rates per node — 1-10 bits/sec average
  – up to 10 kbit/sec in rare local connections to potentially support non-latency critical voice channel
• Crucial Design Parameter:
  Spatial capacity (or density) — 100-200 bits/sec/m²
Wireless Communication Design Space Exploration

Implementation in hard- and software

Source
\((x_s, y_s)\)

Communication Request
(Data type, BW, latency, BER)
(T-C-F-DMA) to Point, multi-hop, star

Physical Layer
(Band, Modulation)

Dest
\((x_d, y_d)\)

- Step-wise refinement (partitioning, resource mapping and sharing) enables optimization, cost analysis, and correctness verification
- Based on well-defined abstraction layers

DAC System Level Design with Embedded Platforms Tutorial

Communicating over Long Distances
Multi-hop Networks

Example:
- 1 hop over 50 m
  1.25 nJ/bit
- 5 hops of 10 m each
  5 x 2 pJ/bit = 10 pJ/bit
- Multi-hop reduces transmission energy by 125!
  (assuming path loss exponent of 4)

But … network discovery and maintenance overhead

Optimal number of hops needed for free space path loss.

DAC System Level Design with Embedded Platforms Tutorial
Network simulators combine functional models with cost model for computation and communication.

**Comparing Energy Cost of Networking Approaches**

- Energy = Eb * Packet Size
- Reactive Routing good for rarely used routes
- Proactive Routing good for frequently used routes

![Energy Comparison Chart](chart.png)

Routing Overhead (bytes)

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>DSDV</th>
<th>AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>33</td>
<td>1500</td>
<td>3000</td>
</tr>
<tr>
<td>56</td>
<td>2500</td>
<td>5000</td>
</tr>
</tbody>
</table>

Routing Overhead (bytes) Normalized

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>DSDV</th>
<th>AODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2000</td>
<td>4000</td>
</tr>
<tr>
<td>33</td>
<td>6000</td>
<td>12000</td>
</tr>
<tr>
<td>56</td>
<td>10000</td>
<td>20000</td>
</tr>
</tbody>
</table>
Summary

- Low-energy design ascends to prime time forced mainly by the last-meter problem
- System-on-a-Chip approach enables and demands heterogeneous implementation strategies, sometimes involving non-intuitive and innovative design platforms
- Design exploration over various fabrics and partitions has dramatic impact on dominant metrics, such as energy and cost
- It requires orthogonalization of function and architecture, supplemented with performance models (cost, time, energy)
  - Architectural models for exploration in high demand
- This methodology holds at all levels of the system hierarchy
  The Fractal Nature of Design