The MARCO/DARPA Gigascale Silicon Research Center
Jan Rabaey, Gary Baldwin, and Kurt Keutzer

“Empowering designers to realize the potential of gigascale silicon by enabling scaleable, heterogeneous, component-based design”

UCB*, CMU, Michigan, Stanford, Princeton, MIT, Penn State, Purdue, UCLA, UCSD, UCSC, Austin, Wisconsin

* Contracting University
Many words of thanks to …

A. Richard Newton
Founding Director GSRC

“for his ambitions, visions, and unbridled efforts that led to the creation of GSRC and made it is what it is today — the flagship of the Marco Centers”

He made it fly, we will make it soar!
The Design Roadblocks

75% of Designs will be Mixed Signal by 2006

Mixed Signal Design

Heterogeneity

Performance

Power

Resolution (bit)

Signal Bandwidth

1kHz 10kHz 100kHz 1MHz 10MHz 100MHz 1GHz

1kHz audio

Resolution

Cost

$10,000,000

$1,000,000

$100,000

$10,000


Source: T. Vucurevich, Cadence
“It’s a Moonshot, Not Rocket Science”

Proposed GSRC 10-Year Goal, November 1997

Overall Program Goals

- > 1 Billion transistor chip
- In a technology < 0.1 micron
- Using IP from several sources (mixed-signal)
- Running at > 2 GHz on-chip
- With a team of < 30 designers
- In < 6 months
- With competitive cost and power-delay-area product

50nm
10GHz
Creation in Motion: Overarching GSRC Research Emphasis

A broadened focus on application-oriented embedded systems under tight cost, PDA, and time-to-market constraints

Founded on One Basic Principle

“From Ad-Hoc System-on-a-Chip Design to Disciplined, Platform-Based Design”
The Missions

- Chips of just the right complexity
- Designed using the necessary advanced technologies
- Meeting all constraints in the PDA space, at least one of which is very tight
- Making maximal component and architecture reuse
- In a competitive cost and time-to-market scenario
From Moonshot to Planetary Missions (and Beyond)

The Missions

◆ Chips of just the right complexity
◆ Designed using the necessary advanced technologies
◆ Meeting all constraints in the PDA space, at least one of which is very tight
◆ Making maximal component and architecture reuse
◆ In a competitive cost and time-to-market scenario
Design Drivers:
Setting the Context and Providing the Metrics for Success

Digital Radar on-a-chip
• Ultra High Performance
Getting There Rapidly, Safely and Often

The Methodology

- Mission-specific platforms
- Subject to tight constraints (power, cost, size)
- Reusable platforms — not every mission needs a new craft, only modification
- Flexible platforms — missions may need to be redefined during flight
Platform-based Design — A Paradigm Shift
“The Methodology to Get There Rapidly, Safely, and Often”

◆ A Platform
“A precisely-defined articulation point, where the successive refinements of the specification meet with abstractions of potential implementations.” (Alberto Sangiovanni-Vincentelli)

◆ Platform-based Design
“A formalized “meet-in-the-middle” design process that uses precise articulation points and stretches over several layers of abstraction.”

For more on platforms, please defer to Ted Vucurevich’s presentation this evening And our session tomorrow morning
Platform-based Design? A Paradigm Shift
“The methodology to get there rapidly, safely and often”

Removing otherwise seemingly insurmountable roadblocks

◆ Enabling a unique “application-level API”, simplifying the mapping of applications onto flexible and heterogeneous SOCs
◆ Enabling maximal reuse of hardware and software components
◆ Enabling verification at the right time and the right place
◆ Allowing for effective power- and energy management at the right levels of abstraction
◆ Enabling the use of constructive and predictable fabrics
◆ Enabling effective and economic test of heterogeneous SOCs

See: “Defining Platform-based Design,” ASV—GSRC Website
The Discipline of Platform-Based Design

Application

Programming Model: Models/Estimators  Kernels/Benchmarks

Architecture(s)

Architectural Platform

Microarchitecture(s)

Cycle-speed, power, area  Functional Blocks, Interconnect

Circuit Fabric(s)

Silicon Implementation Platform

Manufacturing Interface

Delay, variation, SPICE models  Basic device & interconnect structures

Silicon Implementation
The Discipline of Platform-Based Design

Silicon Implementation Platform

Architectural Platform

Application

Programming Model: Models/Estimators

Kernels/Benchmarks

Architecture(s)

Microarchitecture(s)

Cycle-speed, power, area

Functional Blocks, Interconnect

Circuit Fabric(s)

Silicon Implementation Platform

Manufacturing Interface

Delay, variation, SPICE models

Basic device & interconnect structures

Silicon Implementation

Programmable Systems

Constructive Fabrics

Comp and Comm Based Design

Test, Verification, Energy & Power

Calibrating Achievable Design
Extending the **Platform Concept Upwards:**

**Network Platforms**

- **Formalization of Network Platforms**
  - APIs: sets of Communication Services

- **Application: Design of Picoradio networks**
  - Communication Refinement
  - Formalized in the Ulysses Toolset

---

**Application Layer**

- **CS:** Pull
- **CS:** Push

**Network Layer**

- **CS:** Multi-hop Request delivery, multi-hop Response delivery

---

Comm-based design

---

Max Power, BER
Formalizing the Architecture Platform Layer: Metropolis

Meta Model

- Communication Spec
- Architecture
- Functional Spec
- Constraints

Front end

Abstract syntax tree

Meta model compiler

Back end₁, Back end₂, Back end₃, ..., Back endₙ

Simulator tool
Synthesis tool
Verification Analysis tool

Metro Shell Command Interpreter
Platform-Based Design for Unmanned Aerial Vehicles (UAVs)

◆ Device Platform
  • **Isolates** details of sensor/actuators from embedded control programs
  • **Communicates** with each sensor/actuator according to its own data format, context, and timing requirements
  • **Presents** an API to embedded control programs for accessing sensors/actuators

◆ Language Platform
  • **Provides** an environment in which synchronous control programs can be scheduled and run
  • **Assumes** the use of generic data formats for sensors/actuators made possible by the Device Platform

**Sensors:** INS, GPS
**Actuators:** Servo Interface
**Vehicles:** Yamaha R-50/R-Max

Comm-based design Th
Mastering Application-Specific Programmable Architecture Platforms (Mescal)

◆ Unify the programming of hardware resources from FPGAs to conventional processors in a single unified framework.
  - Provides for a description of the processor model, as well as compilation of the application onto this model.
◆ The key to efficient application specific programmable systems is efficient matching of application concurrency to processor concurrency.
  - The flexibility of programmability adds an overhead. Need to minimize the overhead of programmability.
  - [Huang+ DAC02]

Fully Programmable Systems
Energy-Aware Modeling and Compilation

ProgrammablePlatforms$^T_h$ + PowerEnergyDesign$^T_h$
Peripherals in Programmable Platforms (Mescal)

- Peripherals add significant complexity in the design and deployment of programmable platforms.
- Provide complete generation of programmable interfaces (drivers) from formal peripheral specifications.
- Integrate multiple peripherals into a single programmable peripheral for significant platform simplification.

Fully Programmable Systems}{th}
Platform Verification

Verification Target

Verification Engine

- Microprocessors
  - Prove that processor implements ISA

- System Software
  - Identify typical patterns and detect exceptions
  - Work directly on code without any guidance from user

- Communication Protocols
  - Buses, networks

- Bit-Level Verification
  - Model checkers
  - Symbolic simulators
    - New tools that bridge between model checking and simulation

- Term-Level Verification
  - View data words as symbolic values
  - More abstract than RTL model
Bridging from Application to Engine — How and When to Apply an Engine?

◆ Model Extraction
  - Extracting patterns from code
    - C code: Generate finite-state abstraction
    - Verilog: Generate bit- or term-level models

◆ Integrated Design/Verification Environments
  - Support top-down design refinement

◆ Design for Verifiability
  - Building checkers into hardware
    - Then just need to verify the checker to ensure total system correctness
Embedded-Software-Based Self-Testing For Programmable Systems

TestTh

VCI : Virtual Component Interface

Test program
Responses
Signatures

Bus Interface Wrapper

CPU

DSP

VCI

Bus Arbiter

On-chip Bus

External Tester

Bus Interface Wrapper

System Memory

Target Wrapper

VCI

IP Core (with scan)

Test Support

Wrapper

IP core

Bus Interface Wrapper

Scan Interface
Data Buffer
Platform-Based Implementation

- Platforms eliminate *large loop iterations* for affordable design
- Restrict design space via new forms of regularity and structure that surrender *some* design potential for lower cost and first-pass success
- Platforms defined by *new* top-down and bottom-up regularity

Comm-based design\textsuperscript{Th} + Constructive Fabrics\textsuperscript{Th}
From Architecture to Silicon Implementation Platform

- Different targets employ different intermediate platforms, hence different layers of regularity and design-space constraints
- Design space may actually be smaller than with large steps!
  - Large-step predictions/abstractions may misguide the optimizations

Constructive Fabrics$^Th$
**Silicon Implementation Platforms (SIPs)**

- New SIPs are emerging that *roll-up* various layers of regularity to provide for more affordable ASIC implementation.
  - Without new SIPs, an increasing number of products will have no *affordable* application-specific customization options.

<table>
<thead>
<tr>
<th>Component Regularity and Reuse</th>
<th>Regular Fabrics</th>
<th>Geometrical Regularity</th>
<th>Silicon Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstractions</strong></td>
<td><strong>SIP</strong></td>
<td><strong>Constructive Fabrics</strong></td>
<td><strong>See session tomorrow morning</strong></td>
</tr>
</tbody>
</table>

- **Architecture**
  - **Logic Regularity**
Energy-Efficient Fabrics

- Combining multiple VDDs, VTs, and sizing to minimize active static power dissipation
- Managing leakage in logic and memory
- Quantification of bounds and overhead of energy-reduction techniques

Managing standby power
In memories

Supply voltage optimization

<table>
<thead>
<tr>
<th>delay increment [%]</th>
<th>energy reduction [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

UB: cVdd, inv. chain
LB: gVdd, any design

PowerEnergyDesign

Negligible

Managing standby power
In memories
Effective Test Techniques for Silicon Platforms

◆ Embedded SW-Based Self-Testing
  • SW-based testing/-diagnosis for new DSM defects (delay and X-talk induced faults, resistive-short, resistive open, etc)

◆ Testing for DSM Delay Defects
  • Fault modeling of new DSM defects for efficient Test Generation, fault grading and self-testing
  • Investigate and validate Idd wavelet-based testing under various process variations and its application for fault location in analog circuits

◆ Analog/Mixed-Signal Self-Testing
  • Self-testing of high-resolution (≥16 bits) converters
  • On-chip measurement for high-speed serial communication links
Calibrating Achievable Design (C.A.D.) Theme

This Theme addresses Design Technology Productivity gaps

◆ Specification Gap
  ♦ What will be the critical design problem?
  ♦ GSRC Technology Extrapolation (GTX) System and “Living ITRS” [Website]

Source: SEMATECH

Source: SEMATECH
Calibrating Achievable Design (C.A.D.) Theme

This Theme addresses Design Technology Productivity gaps

- Development and Delivery Gap
  - How to deploy DT better and faster? = TimeToMarket, QOR
  - GSRC Bookshelf for CAD-IP Reuse
    http://vlsicad.eecs.umich.edu/BK
Calibrating Achievable Design (C.A.D.) Theme

This Theme addresses Design Technology Productivity gaps

- **Measurement and Improvement Gap**
  - Did envelope of achievable design grow? = METRICS
  - GSRC METRICS System for Design Process Optimization  [http://vlsicad.ucsd.edu/METRICS](http://vlsicad.ucsd.edu/METRICS)
GSRC – TOP Accomplishments in Last Year

◆ Establishment of disciplined “Platform-Based Design” concept as a major paradigm shift for Gigascale design
  • See white paper at http://www.gigascale.org/pubs/141.html

◆ The Living ITRS
  • Use of GSRC Technology Extrapolation (GTX) through in 2001 ITRS Roadmap, providing first time ever consistency checks for power, die size, density, performance
  • Models and studies included along with GTX distribution of ITRS 2001

◆ Established Embedded Software-based Self-testing as a new test paradigm for:
  • Programmable Systems
  • Delay and Crosstalk-induced faults
  • Software-based diagnosis of processor cores

◆ Developed method of Symbolic Simulation with Approximate Values
  • Applied to block in network traffic management chip
  • Verified correctness of > $2^{36}$ combinations of control signal values

Many exciting things on near horizon …
GSRC at DAC

- 31 (out of 165) papers involving GSRC faculty, post-doc’s and students
- Faculty involved in 6 panels and 3 tutorials
- Tu June 11, 1:30-2pm, Dac Pavilion, “Focus Centers: Your Tax Dollars at Work,” featuring Rob Rutenbar (C2S2) and Jan Rabaey (GSRC)
Summary – Not Just Research as Usual

◆ Platform-Based Design the Unifying Theme Inspiring Many Innovative and Exciting Research Projects
  • A major paradigm shift that impacts every aspect of Design Technology
  • Provokes cross-boundary thinking yielding novel and “unusual” solutions

◆ Bringing Design Drivers to the Forefront
  • Essential as integrator within and between themes
  • Multiple drivers needed to span the extreme corners of the design space
  • Metrics for success

◆ Bringing Design Technology into A New Era
  • Making the Impossible Possible
Revisiting the Quarterly Meetings

Broader exposure of efforts within themes to community-at-large

◆ “Highlight a Theme” sessions (about 2 per quarter)
  ◦ Tomorrow Morning: Communication-based design and Constructive Fabrics

◆ The GSRC Symposium
  ◦ Two-day long GSRC conference, combining presentations and posters
  ◦ Inauguration: September 03
  ◦ Presentations focus on major accomplishments and breakthroughs
    ♦ Given by faculty and students; selected by small committee
  ◦ Today’s symposium is first trial run