What is the System Design Riddle?

How do I complete my design on-time?

With no errors?

On-budget?

Then marketing says they need it...

- Sooner
- With higher performance
- In less area
- Sooner? Sorry, I meant early...
- ...maybe you can drop a feature?
Sonics’ on-chip networks help leading SoC designers solve some of the most difficult challenges in SoC design.

- **IP Integration**
- **High Frequency**
- **Memory Throughput**
- **Physical Design**
- **Power Management**
- **Security**
- **Time-to-market**
- **Development costs**

**Sonics System IP:**
On-chip Networks, Memory Subsystem, Power Partitioning & Management, Performance Monitor & Debug, Security Firewalls
Sonics – The NoC Leader for 18 Years

Sonics enables designers to integrate any IP from anywhere, anytime
- Easy IP re-use
- Connecting third party IP/subsystems

Total System IP approach:
- Intelligent memory scheduling
- Optimal power-aware designs
- Data flow services: QoS, Security firewalls
- Software drivers

Commanding presence in digital entertainment, mobile and wireless
- 200+ SoC tape-outs

Results: 2.5B+ units shipped
- 138+ patent properties
- Improved TTM and quality
No, not entirely

- *Using Sonics and other high-quality IP will aid greatly*
- *But there are limitations that good IP alone doesn’t solve*
  - Your design methodology is probably wrong, though it may be the best you can do today
What’s wrong? Waterfall Methodologies

- Sequential operations
  - *Little parallelism*
- Changes in one phase may result in a reset of all downstream steps
- While design reuse (and use of purchased IP) is allowed, it cannot be fully exploited
- If one feature is causing a delay, it may be impossible to move forward with the rest of the design until it is resolved
- In general, the focus is on the process, rather than the desired outcome

*It’s easy to see the problems with the waterfall method. It assumes that every requirement can be identified before any design or coding occurs.*
Some drivers for change

- **Consumer** products
  - Short release cycle
  - Low power, but still always on
  - Security
  - Design costs

- **IoT**
  - Starting design without a complete spec

- **Makimoto’s wave** shifts emphasis from standardization to customization

- **System companies** are in best position to assume product risk

Desired new attributes

- Ability to make reasonable progress with an incomplete specification

- To better solve modern design challenges:
  - Time-to-market
  - Power
  - Security

- Ability to easily ship “on-time”/early with a reduced feature set

- Improved relationship between architectural, logical, physical, and software design
Why is Sonics leading this?

SO MANY BLOCKS!?

Source: Semico Research Corp.
Why is Sonics leading this?

SO EXPENSIVE!?!

These problems all involve the chip architect!

Source: Semico Research Corp.
SoC Architects Drive Both SW & HW

- Large (and growing) teams dependent upon architecture
- SoC architect responsible for many views of architecture
  - Normally disparate
- Increasing complexity of SW and HW results in increased costs and delays
The Agile Manifesto was written in February of 2001, at a summit of seventeen independent-minded practitioners of several programming methodologies. The participants didn't agree about much, but they found consensus around four main values:

- **Individuals and interactions over processes and tools**
- **Working software over comprehensive documentation**
- **Customer collaboration over contract negotiation**
- **Responding to change over following a plan**

“*The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability.*”
## HW and IC Design Teams Must Evolve

<table>
<thead>
<tr>
<th>Yesterday</th>
<th>Tomorrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component-Level Aggregation</td>
<td>System-Level Architecture</td>
</tr>
<tr>
<td>One-time Usage</td>
<td>Platform Reuse</td>
</tr>
<tr>
<td>Hierarchical Requirements and Stepwise Design Management</td>
<td>Concurrent Engineering and Agile IC Methodology</td>
</tr>
<tr>
<td>Dedicated Resources</td>
<td>Shared Resources</td>
</tr>
</tbody>
</table>
What an Agile IC Methodology Might Look Like

Specification
Prototype
Design
Simulation
Verification
Emulation
Physical Design

Waterfall

Time
Agile IC Methodology

Specification
Prototype
Design
Simulation
Verification
Emulation
Physical Design

Time

Change the slope! The more vertical the better!
Short term actions

Sonics’ Plan
- Get the discussion started
- Create the communications links
- Gather the interested parties

How you can help
- Join the LinkedIn group
- Participate in the conversation
- Invite others to get involved too

Step 1: Join the “Agile IC Methodology” group on LinkedIn
Step 2: Join the conversation – participate!
Partnering to Win with Sonics
Now, and in the Future

Sonics has

• The best technology
• The strongest commitment
• The largest team
• The most experience
• The best support
• The strongest roadmap

What can we do to help your team?