Is Hardware Innovation Over?

Arvind
Computer Science and Artificial Intelligence Laboratory
M.I.T.

and

WCU Distinguished Professor, Seoul National University

Seoul National University, Seoul, Korea
October 7, 2009
The power of numbers

- Last year 950M cell phones were sold as opposed to 100M PC
- India & China are each selling > 7M new cell-phone connections per month
  - In developing countries cell phone is the only computer most people have
  - In the developed world cell phone is the only computer people carry all the time

A shift in research is underway from PCs to cell phone, not very different from the shift from Mainframes and Minis to PCs in early eighties.
The future would be dominated by the concerns of

便宜且强大的手持设备

以及

强大的基础设施来支持这些设备上的服务。
Current Cellphone Architecture

Two chips, each with an ARM general-purpose processor (GPP) and a DSP (TI OMAP 2420)

Many specialized complex blocks but must not dissipate more than 3 watts
Real power saving implies specialized hardware

- H.264 video decoder implementations in software vs. hardware
  - the power/energy savings could be 100 to 1000 fold

but our mind set is that hardware design is:

- Difficult, risky
  - Increases time-to-market
- Inflexible, brittle, error prone, ...
  - Difficult to deal with changing standards, ...

New design flows and tools can change this mind set
SoC & Multicore Convergence: more application specific blocks

Application-specific processing units
General-purpose processors
Structured on-chip networks
On-chip memory banks

Is consumer space different from enterprise space?
Server Microprocessors

Also highly regular multicores with lots of specialized processing capabilities for
- compression/decompression
- encryption/decryption
- intrusion detection and other security related solutions
- Dealing with spam
- Self diagnosing errors and masking them
- ...

One way to provide these functionalities is via on-chip FPGAs
Server Multicore

more memory, cores, reconfigurable logic...

Quality-of-Service (QoS) aware on-chip networks and resource management are essential for guaranteeing performance
Architectural Renaissance

- Unprecedented opportunity to rethink parallel architectures
- Unprecedented need to design low-power functional blocks
- Unprecedented opportunity to experiment offered by large FPGAs and high-level synthesis tools
Bluespec: A new way of expressing behavior

- A formal method of composing modules with parallel interfaces (ports)
  - Compiler manages muxing of ports and associated control
- Powerful and **zero-cost** parameterization of modules
  - Encapsulation of C and Verilog codes using Bluespec wrappers
  - Helps Transaction Level modeling

- **Smaller, simpler, clearer, more correct code**
- **not just simulation, synthesis as well**
High-level Synthesis from Bluespec

Bluespec SystemVerilog source

Bluespec Compiler

C

Verilog 95 RTL

Bluesim

Cycle Accurate

VCD output

Debussy Visualization

Power estimation tool

Verilog sim

RTL synthesis

gates

Place & Route
Tapeout

Physical

FPGA

Accurate
Bluespec enables

- **Extreme IP reuse**
  - Multiple instantiations of a block for different performance and application requirements
  - Packaging of IP so that the blocks can be assembled easily to build a large system (black box model)

- **Architectural exploration**

  "Intellectual Property"

*An example*
IP Reuse via parameterized modules

Example OFDM based protocols

- WiFi: 64pt @ 0.25MHz
- WiMAX: 256pt @ 0.03MHz
- WUSB: 128pt 8MHz

- Reusable algorithm with different parameter settings
- 85% reusable code between WiFi and WiMAX
- From WiFi to WiMAX in 4 weeks
- Different algorithms

[MEMOCODE 2007]
## 802.11a Transmitter Design: Preliminary results

<table>
<thead>
<tr>
<th>Design Block</th>
<th>Lines of Code (BSV)</th>
<th>Relative Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>49</td>
<td>0%</td>
</tr>
<tr>
<td>Scrambler</td>
<td>40</td>
<td>0%</td>
</tr>
<tr>
<td>Conv. Encoder</td>
<td>113</td>
<td>0%</td>
</tr>
<tr>
<td>Interleaver</td>
<td>76</td>
<td>1%</td>
</tr>
<tr>
<td>Mapper</td>
<td>112</td>
<td>11%</td>
</tr>
<tr>
<td>IFFT</td>
<td>95</td>
<td>85%</td>
</tr>
<tr>
<td>Cyc. Extender</td>
<td>23</td>
<td>3%</td>
</tr>
</tbody>
</table>

Complex arithmetic libraries constitute another 200 lines of code

[MEMOCODE 2006]
FFT – fold to save area

Reuse the same circuit three times to reduce area
802.11a Transmitter Synthesis results (Only the IFFT block is changing)

<table>
<thead>
<tr>
<th>IFFT Design</th>
<th>Area (mm²)</th>
<th>Throughput Latency (CLKs/sym)</th>
<th>Min. Freq Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelined</td>
<td>5.25</td>
<td>04</td>
<td>1.0 MHz</td>
</tr>
<tr>
<td>Combinational</td>
<td><strong>4.91</strong></td>
<td>04</td>
<td>1.0 MHz</td>
</tr>
<tr>
<td>Folded (16 Bfly-4s)</td>
<td><strong>3.97</strong></td>
<td>04</td>
<td>1.0 MHz</td>
</tr>
<tr>
<td>Super-Folded (8 Bfly-4s)</td>
<td>3.69</td>
<td>06</td>
<td>1.5 MHz</td>
</tr>
<tr>
<td>SF(4 Bfly-4s)</td>
<td>2.45</td>
<td>12</td>
<td>3.0 MHz</td>
</tr>
<tr>
<td>SF(2 Bfly-4s)</td>
<td>1.84</td>
<td>24</td>
<td>6.0 MHz</td>
</tr>
<tr>
<td>SF (1 Bfly4)</td>
<td>1.52</td>
<td>48</td>
<td>12 MHz</td>
</tr>
</tbody>
</table>

All these designs were done in less than 24 hours!

TSMC .18 micron; numbers reported are before place and route.

The same source code
Some cool projects

- Video decoder – H.264
- AirBlue – A new platform to experiment with cross-layer wireless protocols
- IBM PowerPC Prototype and Cycle-accurate performance models
- Hardware software co-generation
H.264 Video Decoder
Chun-Chieh Lin, K Elliott Fleming [MEMOCODE 2008]

Different requirements for different environments
- QVGA 320x240p (30 fps)
- DVD 720x480p
- HD DVD 1280x720p (60-75 fps)

May be implemented in hardware or software depending upon ...
H.264 in Bluespec

- Initial Design: Base profile
  - Eight man-months
  - 8K lines of Bluespec
    - in contrast to 80K lines of C standard
  - Decoded 720p@32FPS

- Major architectural explorations over 3 months to meet different performance or cost criteria
  - High performance designs (4.2 mm sq in 180nm)
    - 720p@75FPS, 1080p@65FPS,
  - Low cost designs
    - QCIF@15FPS (2.2mm sq), 720p@30FPS (2.4mm sq)

Current focus is on high performance FPGA implementations
AirBlue: A platform for Cross-Layer Wireless Protocol development

- **Cross-layer protocols** are the hottest area of research in wireless
  - Jointly optimizing PHY, MAC, network layers
- **Realistic experimentations are difficult**
  - PHY (baseband) layer requires a lot of computation: traditionally in hardware
  - MAC typically done in firmware
  - Higher layers in software

Collaboration with Professor Hari Balakrishnan
Cross-layer wireless protocols require a platform that offers both flexibility/programmability and performance.
Several cross-layer experiments have already been conducted on full-speed 802.11a/g implementation

- SoftPHY: Exposes signal quality to higher layers
  - Enables new protocols: MIXIT, PPR, better rate-adaptation
- Efficient allocation of OFDM channels
  - Variable demands, heterogeneous SNRs

Fits in Nokia N95 phones

Each new protocol required less than 100 lines of code
IBM: PowerPC Prototype

K. Ekanadham, Jessica Tseng (IBM)
Asif Khan, M. Vijayaraghavan (MIT)

Goal: Implement a multithreaded, multicore, in-order PowerPC on an FPGA platform and boot Linux on it in 12 months

Team:
- 2(IBM) + 2(MIT) + Linux and FPGA help

The team accomplished the goal (Nov 2008)
- Bluespec PowerPC boots Linux on FPGAs in 10min;
- 100M instructions to reach “Hello World”; 
- 15K lines of Bluespec generated 90K lines of Verilog

IBM synthesized the generated Verilog using their tools in 40nm library
- ran at 500MHz on the first try!
Phase II: IBM/MIT Collaboration
March 2009 –

Goal: Produce a cycle-accurate and parameterized model of multithreaded, multicore PowerPC to run on FPGAs

- Architecture models in software can be flexible and have high fidelity but tend to be slow
- Can we gain 1000X speedup by running the models on FPGAs?

Use cheaper and widely available FPGA boards
- Xilinx 110 as opposed to 330

Target open source distribution by summer 2010

Lots of technical challenges
Currently trying to boot linux
Could we have done these projects in C, C++, SystemC?
Hardware synthesis from C does not work very well:
Reed Solomon Results

<table>
<thead>
<tr>
<th></th>
<th>Bluespec</th>
<th>C-synthesis</th>
<th>Xilinx IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalent Gate Count</td>
<td>267,741</td>
<td>596,730</td>
<td>297,409</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>108.5</td>
<td>91.2</td>
<td>145.3</td>
</tr>
<tr>
<td>Steady State (Cycles/Block)</td>
<td>276</td>
<td>2073</td>
<td>660</td>
</tr>
<tr>
<td>Data rate (Mbps)</td>
<td>701.3</td>
<td>89.7</td>
<td>392.8</td>
</tr>
</tbody>
</table>

WiMAX requirement is to support a throughput of 134Mbps

For the same area!

Abhinav Agarwal, Alfred Ng
Hardware innovation is far from over

- Ubiquitous mobile devices and demand for new services are ushering in a new era of computing
- Large FPGAs are offering an unprecedented opportunity to experiment
- High-level synthesis tools like Bluespec are making architecture exploration and SoC development much easier
  - High quality synthesis
  - Modules with formal interfaces (not just wires)
  - Parameterized modules (higher-order functions)
  - Strong type system
  - Ability to interact with modules written in C, Verilog, ...

Thanks!
Exploiting Multiple Clock Domains in Bluespec for Hw/Sw cogeneration

- MCD allows us to run parts of the design at different speeds
- Each GAA/Method is associated with a clock
- Special Module to Cross Clocks
- The idea works even if some of the domains are implemented in software

Software
Slow Clock Domain

Rule A
State

Rule B
State

Hardware
Fast Clock Domain

HW/SW Interface

Nirav Dave, Myron King