Building a Universal Silicon Compiler

Andreas Olofsson
Program Manager
Defense Advanced Research Project Agency (DARPA)

The Salishan Conference on High Speed Computing,
Gleneden Beach, Oregon
April 24
Modern day hardware design has a cost problem!

Complexity is the root cause!
Why can’t we build low volume systems at low cost?

Disruption

Source: General Dynamics

The last 50+ Years of Moore’s Law

<table>
<thead>
<tr>
<th></th>
<th>1946</th>
<th>Today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed, mph (S)</td>
<td>78</td>
<td>102</td>
</tr>
<tr>
<td>Efficiency, mpg (E)</td>
<td>14.6</td>
<td>22</td>
</tr>
<tr>
<td>Cost, $K (C)</td>
<td>1.7</td>
<td>27</td>
</tr>
<tr>
<td>( \frac{(S \times E)}{C} )</td>
<td>669</td>
<td>83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1946</th>
<th>Today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed, OPS/S (S)</td>
<td>359</td>
<td>17.9e15</td>
</tr>
<tr>
<td>OPS/W (E)</td>
<td>0.002</td>
<td>2e9</td>
</tr>
<tr>
<td>Cost, $M (C)</td>
<td>6.5</td>
<td>97</td>
</tr>
<tr>
<td>( \frac{(S \times E)}{C} )</td>
<td>0.11</td>
<td>3e23</td>
</tr>
</tbody>
</table>

[https://www.allpar.com/history](https://www.allpar.com/history)  
[https://www.caranddriver.com/chevrolet/cruze](https://www.caranddriver.com/chevrolet/cruze)  
We need to keep reaching for the Landauer Limit!

\[ E_{\text{MIN-BIT}} = kT \ln 2 = 3 \times 10^{-21} \text{ Joules} \]
1970’s: Heroic human chip design efforts

- 10-um feature size
- 2,300 transistors

Source: https://en.wikipedia.org/wiki/Intel_4004

Rubylith operators

Source: http://www.computerhistory.org/revolution/artifact/287/1614
1980’s: Tackling chip design complexity with CAD

Birth of Modern EDA

- Intel 80386 (1985-2007)
- 1-um feature size
- 275,000 transistors

Synthesis

Place and Route

Frameworks

Layout Systems

Source: https://en.wikipedia.org/wiki/Intel_80386
Source: http://opencircuitdesign.com/magic/
Source: Introduction to VLSI systems by Carver Mead

Synopsys  Cadence  Mentor Graphics
1990’s-Today: Barely keeping up with complexity

Source: https://nvidianews.nvidia.com/file?fid=59129280a138351b9447113c

- NVIDIA V100 (2017-)
- 0.012um feature size
- 21,000,000,000 transistors

“It took several thousand engineers several years to create, at a development cost of $3 billion.”
–Jensen Huang

Death by a million papercuts...functional correctness, security, safety, reliability, performance, IP integration, power management, firmware, system integration, wire delays, placement, routing, clocking, design rules, antenna effects, ESD, multi voltage domains, power gating, multi threshold, floor-planning, I/O structures, flip-chip, wirebonding, 2.5D packaging, TSVs, RDL routing, area minimization, routing congestion, on-chip variability, self heating, stress and proximity effects, electro migration, SEUs, signal integrity, power delivery networks, decoupling, modeling, low voltage operations, cooling, scan insertion, BIST, ATPG, STA, yield optimization, power minimization, and all the EDA tools needed to make this work.
Epiphany-V: My last chip before joining DARPA

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>TSMC 16FF+</td>
</tr>
<tr>
<td>Transistors</td>
<td>4.5B</td>
</tr>
<tr>
<td>Die Area</td>
<td>117 mm²</td>
</tr>
<tr>
<td>Flip Chip Bumps</td>
<td>3,460</td>
</tr>
<tr>
<td>I/O Signals</td>
<td>1,040</td>
</tr>
<tr>
<td>Clock Domains</td>
<td>1,152</td>
</tr>
<tr>
<td>Voltage Domains</td>
<td>2,052</td>
</tr>
<tr>
<td>Frequency</td>
<td>500Mhz*</td>
</tr>
<tr>
<td>32 bit Performance (Peak)</td>
<td>2 TFLOPS</td>
</tr>
<tr>
<td>64 bit Performance (Peak)</td>
<td>1 TFLOPS</td>
</tr>
<tr>
<td>Memory Bandwidth</td>
<td>16 TB/sec</td>
</tr>
<tr>
<td>NOC Bisection Bandwidth</td>
<td>0.75 TB/sec</td>
</tr>
<tr>
<td>Typical Power</td>
<td>~10W</td>
</tr>
<tr>
<td>Minimum Power</td>
<td>1mW</td>
</tr>
</tbody>
</table>

## Epiphany-V: Performance Comparisons

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>CPU1</th>
<th>CPU2</th>
<th>E5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>16nm</td>
<td>14nm</td>
<td>14nm</td>
<td>16nm</td>
</tr>
<tr>
<td>Processor Nodes</td>
<td>56</td>
<td>72</td>
<td>24</td>
<td>1024</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>610</td>
<td>683</td>
<td>456</td>
<td>117</td>
</tr>
<tr>
<td>Transistors</td>
<td>15.3B</td>
<td>7.1B</td>
<td>7.2B</td>
<td>4.5B</td>
</tr>
<tr>
<td>Power (W)</td>
<td>250</td>
<td>245</td>
<td>145</td>
<td>~10</td>
</tr>
<tr>
<td>TFLOPS</td>
<td>4.7</td>
<td>3.6</td>
<td>1.3</td>
<td>1</td>
</tr>
<tr>
<td>GFLOPS/mm²</td>
<td>7.7</td>
<td>5.27</td>
<td>2.85</td>
<td>8.55</td>
</tr>
<tr>
<td>GFLOPS/W</td>
<td>18.8</td>
<td>14.69</td>
<td>9.08</td>
<td>100</td>
</tr>
<tr>
<td>SRAM/mm²</td>
<td>0.034</td>
<td>0.05</td>
<td>0.15</td>
<td>0.54</td>
</tr>
<tr>
<td>Nodes/mm²</td>
<td>0.09</td>
<td>0.11</td>
<td>0.05</td>
<td>8.75</td>
</tr>
</tbody>
</table>

*World leading performance at less than $1M design cost*
“Server Farm”:

- One 2010 Dell PowerEdge T610 with a quad-core Xeon 5500 and 32GB DDR3
- One RTL to GDS EDA license

<table>
<thead>
<tr>
<th>Designer</th>
<th>Responsibility</th>
<th>Effort (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contractor A</td>
<td>FPU</td>
<td>200</td>
</tr>
<tr>
<td>Contractor B</td>
<td>Verification</td>
<td>200</td>
</tr>
<tr>
<td>Contractor C</td>
<td>EDA Services</td>
<td>112</td>
</tr>
<tr>
<td>Ola Jeppsson</td>
<td>Simulator/SDK</td>
<td>500</td>
</tr>
<tr>
<td>Andreas Olofsson</td>
<td>Everything else</td>
<td>4,100</td>
</tr>
</tbody>
</table>

Why is low cost chip development not common place?
...why I joined DARPA

Modern Software Compilation

A Universal No Human In the Loop Hardware Compiler

Source Code (.c, .cpp, .h)
- Preprocessing
  - Preprocessor (cpp)
  - Include Header, Expand Macro (.i, .ii)

Assembly Code (.s)
- Compilation
  - Compiler (gcc, g++)

Machine Code (.o, .obj)
- Assemble
  - Assembler (as)

Static Library (.lib, .a)
- Linking
  - Linker (ld)

Executable Machine Code (.exe)

Software compilers moved beyond humans in the loop 50 years ago!

POSH

Source Code, Schematics, Constraints
- Synthesis/Generators
  - Circuit
- IDEA
  - Intelligent No Human In the Loop Layout Engine
  - Partial layout
- Linker
  - Final layout

Qualified Source Code
- Qualified Circuits
- Qualified Layout

A no-human in the loop hardware compiler addresses cost, schedule, resource, trust challenges in current SoC design cycle.
How is hardware “compilation” (layout) handled today?

**Analog Design:**
- Schematic input
- 1K-100K signals
- 100% EDA assisted manual labor
- 2-4 experts
- 3-18 months

**Digital Design:**
- Verilog netlist input, constraints, scripts
- 10M-1B signals
- 99% automated place and route
- 1-100 experts
- 3-18 months

**Board Design:**
- Schematic input
- 1-10K signals
- 100% EDA assisted manual labor
- 2-4 experts
- 3-6 months

**Package Design**
- Excel spreadsheet input
- 1K signals
- 100% manual labor
- 2-4 experts
- 2-8 weeks

Sources: Intel, NVIDIA, Adapteva
Sources: Axis, Adapteva
Sources: Analog Devices, Raspberry Pi

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)
Why now? What has changed?

Traditional Algorithmic EDA Research

1983
Optimization by Simulated Annealing
S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi

1987
An Intelligent Compiler SubSystem for a Silicon Compiler
David L. Johansson, Steve K. Traubert, Ken McElvain
3/27/87

1988
How to automate analog IC designs
Knowledge-based systems are relieving the labor-intensive bottlenecks usually associated with such building blocks as op amps and voltage reference.

1993-2018 (stable evolutionary progress)
Optimization algorithms, Productivity & Integration

Cadence Synopsys Mentor Graphics

A New Machine Learning Based EDA Approach

- ML Algorithm Innovations
- Data driven
- Massive compute (Moore’s Law)
- Replacing existing heuristics/humans

Can we map a layout cost function to ML?
Can we access/label enough quality data?

Ref-2003: The Tides of EDA, Alberto Sangiovanni-Vincentelli
Why now? (how this approach is different)

**Today**
Designer provides manual constraints to layout person (or EDA tool)

- Max 10µm from main supply, 0.5µm width
- Place dummies, interdigitize

**IDEA**

- Circuit Classifier
- Assign Strategies
- Auto-Placement
- Auto-Routing

**Novelty:**
Auto create layout constraints by classifying circuit patterns and applying strategies from knowledge database.
IDEA will create a no-human-in-the-loop hardware compiler for translating source code to layouts of System-On-Chips, System-In-Packages, and Printed Circuit Boards in less than 24 hours.
Reinventing Board Development

Today

- 100% Manual
- Error prone
- Rarely optimal

Manual Part Selection

Manual Schematic

Manual Layout

Google

IDEA

Intent

System Generator

Pruning

Goal Optimizer

Schematics

Potential Solutions

Open Parts DB

Schematic

Layout Generator

New Concept: Machine synthesized board from intent and open COTS parts library.
An Open 5M+ Component IC Database

**Today**
- 5M+ parts in circulation
- Information embedded in datasheets and reference designs
- No standard models
- Automatic optimization not possible

**IDEA**
- IC standard models (LEF, LIB, IP-XACT)
- Extend standards for boards / SIPs
- Creation of 5M+ part DB
- **Model all properties needed for constraint based system optimization**

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)
Reinventing the Chip IP Stack

**Software**

- User Content
- Billion Dollar Company Code
  - MemCache
  - Thrift
  - Cassandra
  - Apache
  - PHP
  - Jenkins
- LINUX

**Current SoC Hardware Design**

- Chip Company A
- Chip Company B
- Chip Company C
- Chip Company D

**Infinite Layer Stack**

- $15B+
- Open Source

**POS**H will create a viable open source hardware design and verification ecosystem that enables cost effective design of ultra-complex SoCs.
The State of Open Source Hardware

Still a long way to go!

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)
### How is Hardware Different from Software?

<table>
<thead>
<tr>
<th></th>
<th><strong>Software</strong></th>
<th><strong>Hardware</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programmers</strong></td>
<td>Millions</td>
<td>Thousands</td>
</tr>
<tr>
<td><strong>Writing Code</strong></td>
<td>Easy</td>
<td>Hard</td>
</tr>
<tr>
<td><strong>Reading Code</strong></td>
<td>Hard</td>
<td>Very hard</td>
</tr>
<tr>
<td><strong>Debugging</strong></td>
<td>Hard</td>
<td>Near impossible</td>
</tr>
<tr>
<td><strong>Cost of bugs</strong></td>
<td>Low</td>
<td>Very high</td>
</tr>
</tbody>
</table>

**What technologies are needed to make open source hardware viable?**
### Digital Circuit IP Blocks

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA Fabric</td>
<td></td>
</tr>
<tr>
<td>Multi-core 64-bit RISC-V processor</td>
<td></td>
</tr>
<tr>
<td>sub-system</td>
<td></td>
</tr>
<tr>
<td>GPU (OpenGL ES 3.0)</td>
<td></td>
</tr>
<tr>
<td>PCI Express Controller</td>
<td></td>
</tr>
<tr>
<td>Ethernet Controller</td>
<td></td>
</tr>
<tr>
<td>Memory Controllers</td>
<td></td>
</tr>
<tr>
<td>USB 3.0 Controller</td>
<td></td>
</tr>
<tr>
<td>MIPI Camera Serial Interface controller</td>
<td></td>
</tr>
<tr>
<td>CPU Subsystem</td>
<td></td>
</tr>
<tr>
<td>H264 encoder/decoder</td>
<td></td>
</tr>
<tr>
<td>AES256 encrypt/decrypt</td>
<td></td>
</tr>
<tr>
<td>SHA-2/SHA-3 accelerator</td>
<td></td>
</tr>
<tr>
<td>Secure Digital Controller</td>
<td></td>
</tr>
<tr>
<td>High Definition Multimedia Interface</td>
<td></td>
</tr>
<tr>
<td>Serial ATA Controller</td>
<td></td>
</tr>
<tr>
<td>J ESD204B Controller</td>
<td></td>
</tr>
<tr>
<td>NAND Flash Controller</td>
<td></td>
</tr>
<tr>
<td>CAN Controller</td>
<td></td>
</tr>
</tbody>
</table>

### Mixed Signal Circuit IP Blocks

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard I/O interfaces PHYs</td>
<td>DDR, PCIe, SATA, USB, XAUI, CPRI</td>
</tr>
<tr>
<td>PLL</td>
<td>Range: 10MHz – 10GHz</td>
</tr>
<tr>
<td>DLL</td>
<td>Range: 10Mhz – 10GHz</td>
</tr>
<tr>
<td>Analog to Digital Converters</td>
<td>Range: 1 – 10,000 MSPS</td>
</tr>
<tr>
<td>Digital to Analog Converters</td>
<td>Range: 1 – 10,000 MSPS</td>
</tr>
<tr>
<td>Voltage Regulators</td>
<td>Input: 1.8V – 12V, Output 0.25V – 1.8V</td>
</tr>
<tr>
<td>Monitor circuits</td>
<td>Temperature, voltage, process</td>
</tr>
</tbody>
</table>

How can we cost effectively develop and maintain a high quality catalog of portable open source digital and analog components?
IDEA/POSH End State – A Universal Hardware Compiler

```markdown
$ git clone https://github.com/darpa/idea
$ git clone https://github.com/darpa/posh
$ cd posh
$ make soc42
```
Creating a viable low volume electronic ecosystem

Source: General Atomics

Image Sources: General Atomics, Tesla, Apple

Distribution Statement “A” (Approved for Public Release, Distribution Unlimited)
**Specialization:** The future of supercomputing!

**General Purpose (Exascale Report)**

<table>
<thead>
<tr>
<th></th>
<th>Scientific Machine (ref. Anton)</th>
<th>Machine Learning*</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPS (Exa)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DRAM (TB)</td>
<td>3,600</td>
<td>100</td>
</tr>
<tr>
<td>Power (MW)</td>
<td>68</td>
<td>2</td>
</tr>
<tr>
<td>Cost</td>
<td>$500M</td>
<td>$10M</td>
</tr>
</tbody>
</table>

*$4/GB DRAM, 5 TOPS/W, shared MPW, IDEA/POSH working, market silicon pricing at 14nm

Source: IBM
Source: Cray

https://www.scientificcomputing.com