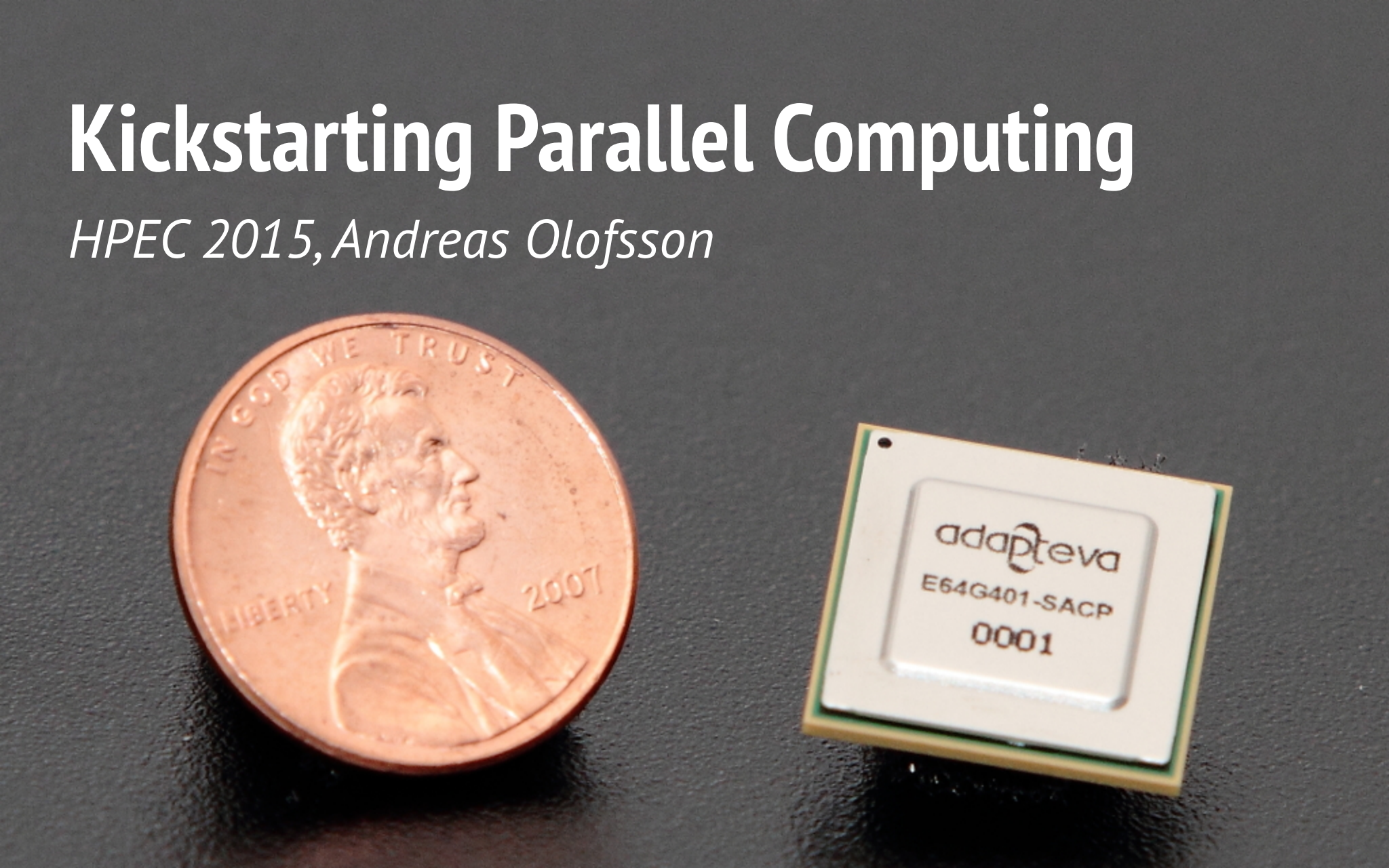
Kickstarting Parallel Computing

HPEC 2015, Andreas Olofsson

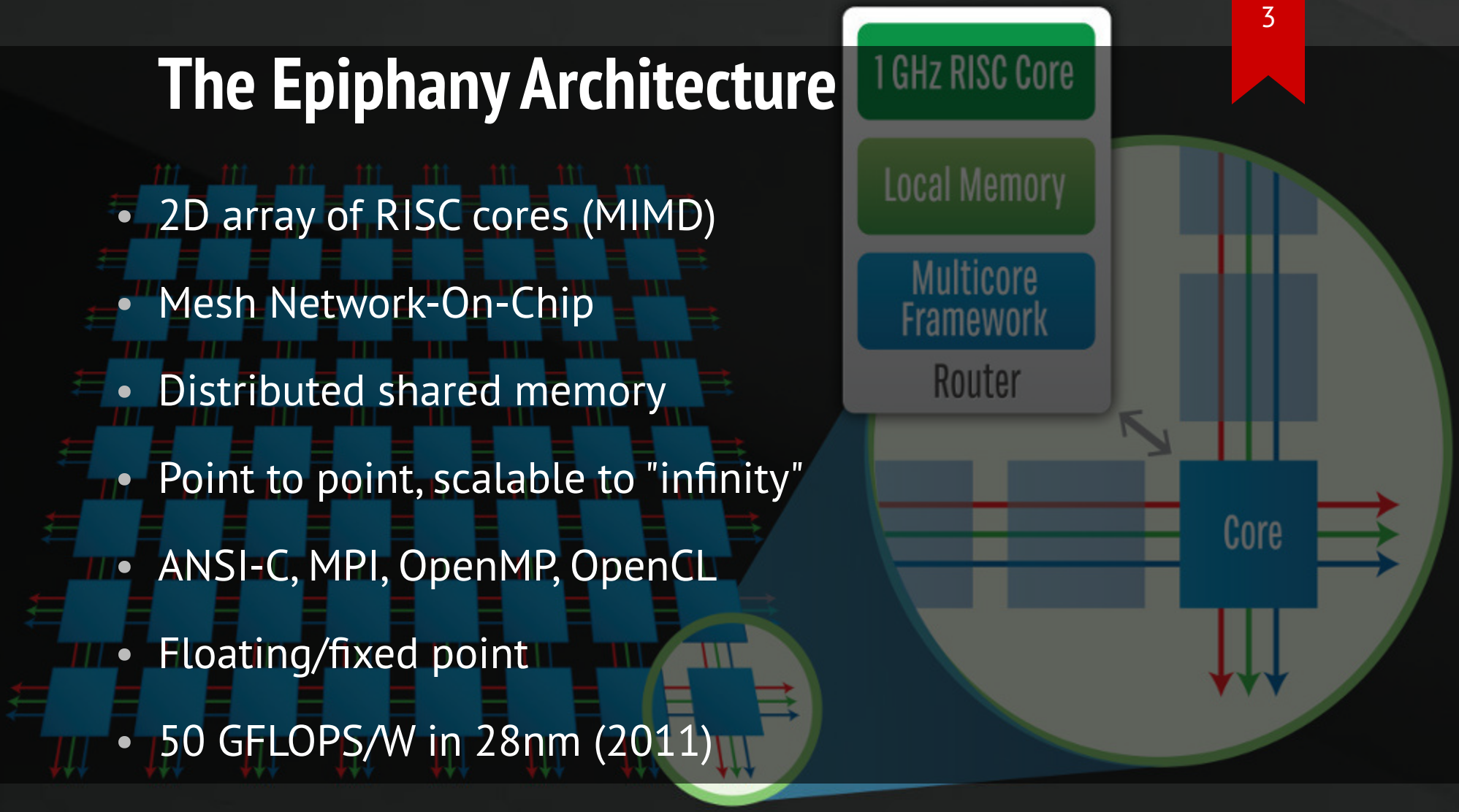


Agenda

1. Epiphany manycore architecture overview
2. The Parallella story
3. Community building basics
4. Parallella success stories
5. Conclusions

The Epiphany Architecture

- 2D array of RISC cores (MIMD)
- Mesh Network-On-Chip
- Distributed shared memory
- Point to point, scalable to "infinity"
- ANSI-C, MPI, OpenMP, OpenCL
- Floating/fixed point
- 50 GFLOPS/W in 28nm (2011)



Epiphany-IV Chip Features

- 64 x 800MHz RISC processors (28nm)
- 32 bit IEEE floating point
- 2MB on-chip SRAM
- North, East, West, South IO links
- 100 GFLOPS peak performance
- 1.6TB/s local memory BW
- 102GB/s NOC bisection bandwidth
- <2W peak power!

Epiphany RISC Processor

- 32 bit, dual issue in order, 5/8 stage pipeline
- 16/32 bit instruction set
- 64 general purpose registers
- IEEE754 floating point (FMADD,FMUL,..)
- Nested interrupts, 2-Channel DMA, debug unit
- ISA: B, BL, JR, JAL, LDR, STR, TESTSET, ADD, SUB, ASR, LSR, LSL, ORR, AND, EOR, BITR, FADD, FSUB, FMUL, FMADD, FMSUB, FABS, FIX, FLOAT, MOV, MOVT, MOVFS, NOP, IDLE, RTS,

Epiphany Memory System

- 32 bit addressing
- Shared flat address space, no HW caches!
- Upper 12 bits specify coordinates in 2D map (64 x 64 mesh)
- 32KB SRAM per core in E16G301. Accessible by all cores.
- 4 independent 64 bit local memory transactions per cycle
- Fetch, load, DMA, emesh supports 32 byte access per cycle
- Strict local memory ordering, VERY relaxed remote odering.

Epiphany Network-On-Chip

- 3 meshes: on-chip writes, read requests, off-chip writes
- 104 bit atomic single cycle packets
- Non-blocking round robin routing
- x/y static routing
- 8 bytes transfered per cycle (on chop write mesh)
- ~1 clock cycle latency / hop
- Extends off chip to I/O (elinks)

Adapteva History

- 2008: Parallel processor company founded in Lexington, MA
- 2010: 16-core 65nm processor (25GFLOPS/W)
- 2011: 64-core 28nm processor (50GFLOPS/W)
- Accomplished with 3 engineers and $2M

But it wasn't enough!

$2B+ of parallel architecture R&D

Achronix	Brightscale	Cradle	Mathstar	Sandbridge
Adapteva	Calxeda	C-Switch	Mobileye	Silicon Sp.
Ambric	Chameleon	ElementCXI	Monarch	Stream Proc
Asocs	Clearspeed	Greenarrays	Octasic	Stretch
Aspex	Cognivue	Icera	Picochip	Tilera
Axis Semi	Coherent L.	Intellasys	Plurality	Transputer
BOPS	CELL	IP-flex	PACT	XMOS
Boston C.	CPUTech	Larrabee	Quicksilver	Zilabs

My 2012 "2nd Epiphany"

1. Pricing and paranoia was killing us!
2. Openness is the "secret" to building platforms
3. Follow the leaders:

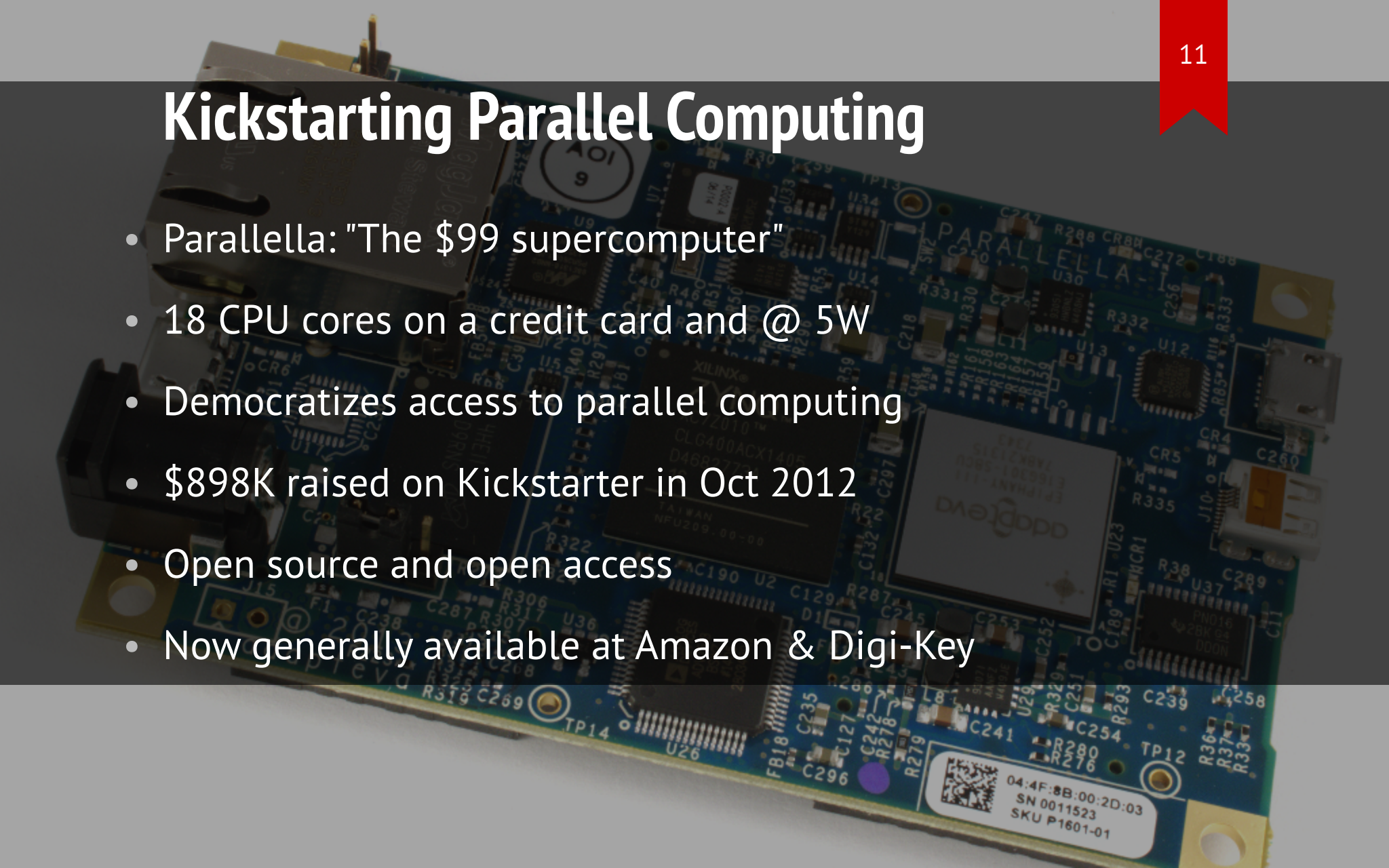
   -Arduino & Raspberry Pi --> low price breeds adoption

   -Kickstarter --> capital transparency

   -Linux --> open collaboration leadership

   -ARM --> importance of strong corporate partners

Kickstarting Parallel Computing



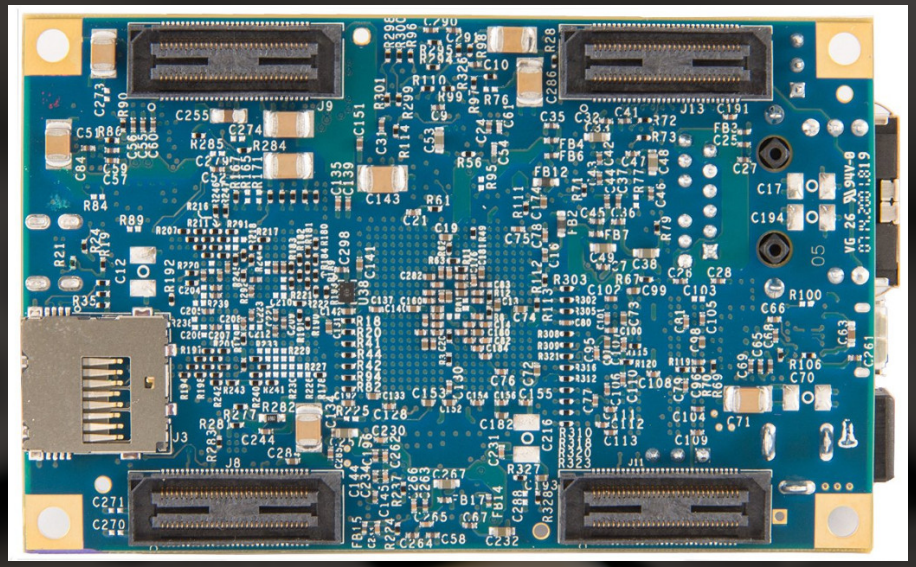
- Parallella: "The $99 supercomputer"
- 18 CPU cores on a credit card and @ 5W
- Democratizes access to parallel computing
- $898K raised on Kickstarter in Oct 2012
- Open source and open access
- Now generally available at Amazon & Digi-Key

Parallella Specs (http://parallella.org)

Features	Zynq-FPGA, 16X Epiphany, GigE, uSD, USB
Performance	~30 GFLOPS
Memory	1GB DDR3
IO	~25 Gb/s (48 GPIO)
Size	credit-card
Weight	38g
Power	<5W
Cost	$99 -> $249

Parallella IO

- 0.5mm Samtec connectors
- 48-pin/24Gbps FPGA link
- 2 Epiphany links (20Gbps)
- JTAG, UART, I2C, SPDIF
- LVDS/CMOS
- Adjustable I/O voltage



Parallella SDR Platform

- 70MHz - 6GHz
- ADI FCOMMS board
- Parallella carrier
- FMC adapter board
- 100% Open source SW

Openness: Before & After

Metric	Before	After	Boost
Customers	5	10,000	2000x
Universities	1	200	200x
Site traffic	20	1,000	50x
Twitter Followers	200	6,000	30x
Publications	2	30*	15x
Govt customers	2	10	5x
Yearly Sales	$45K	$1.8M	40x

Hurdles to Community Building

Typical Skill	Unicorn	Researcher	Maker	Consumer
Board design	Yes	No	No	No
Software plumbing	Yes	No	No	No
System administration	Yes	No	Yes	No
Soldering, assembly	Yes	No	Yes	No
Building application	Yes	Yes	Yes	No
RTFM	Yes	Yes	Yes	?
Total Reach	10	10K	10M	1B

Programming Models

- PAL (read/write/load) (github.com/parallella/pal)
- OpenMP Device (paragroup.cs.uoi.gr/wpsite/?download=524)
- OpenCL (github.com/browndeer/coprthr)
- MPI (browndeertechnology.com/code)
- Threads (browndeertechnology.com/coprthr.htm)
- BSP (github.com/coduin/epiphany-bsp)
- Erlang (Uppsala University, not released yet)

Community Success Stories

- **Commercial**
  - Ericsson showed 25x efficiency edge over Intel
- **Govt**
  - ARL programmed Epiphany using efficient standard MPI
- **Academia**
  - ANU demonstrated 85% of peak performance
- **Hackers**
  - S. Munaut showed that Epiphany core =~ 1.5x ARM-A9

Conclusions

- Adapteva thriving thanks to unprecedented transparancy
- It's the software stupid, you have a responsibility to help.
- If you are working in the following areas, come see me!
    - SDR, computer vision, robotics, drones, secure processing, CubeSats, CNNs.
- Thousand core processors coming soon, are you ready?