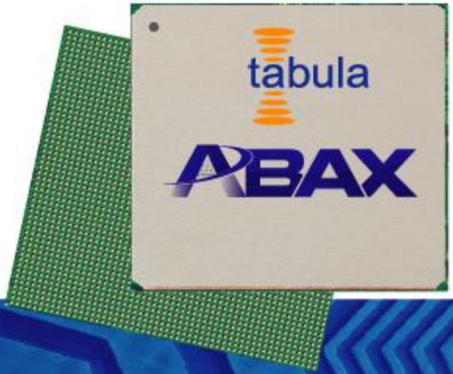
Why programmability is necessary but not sufficient for future logic chips

Steve Teig – Founder and CTO



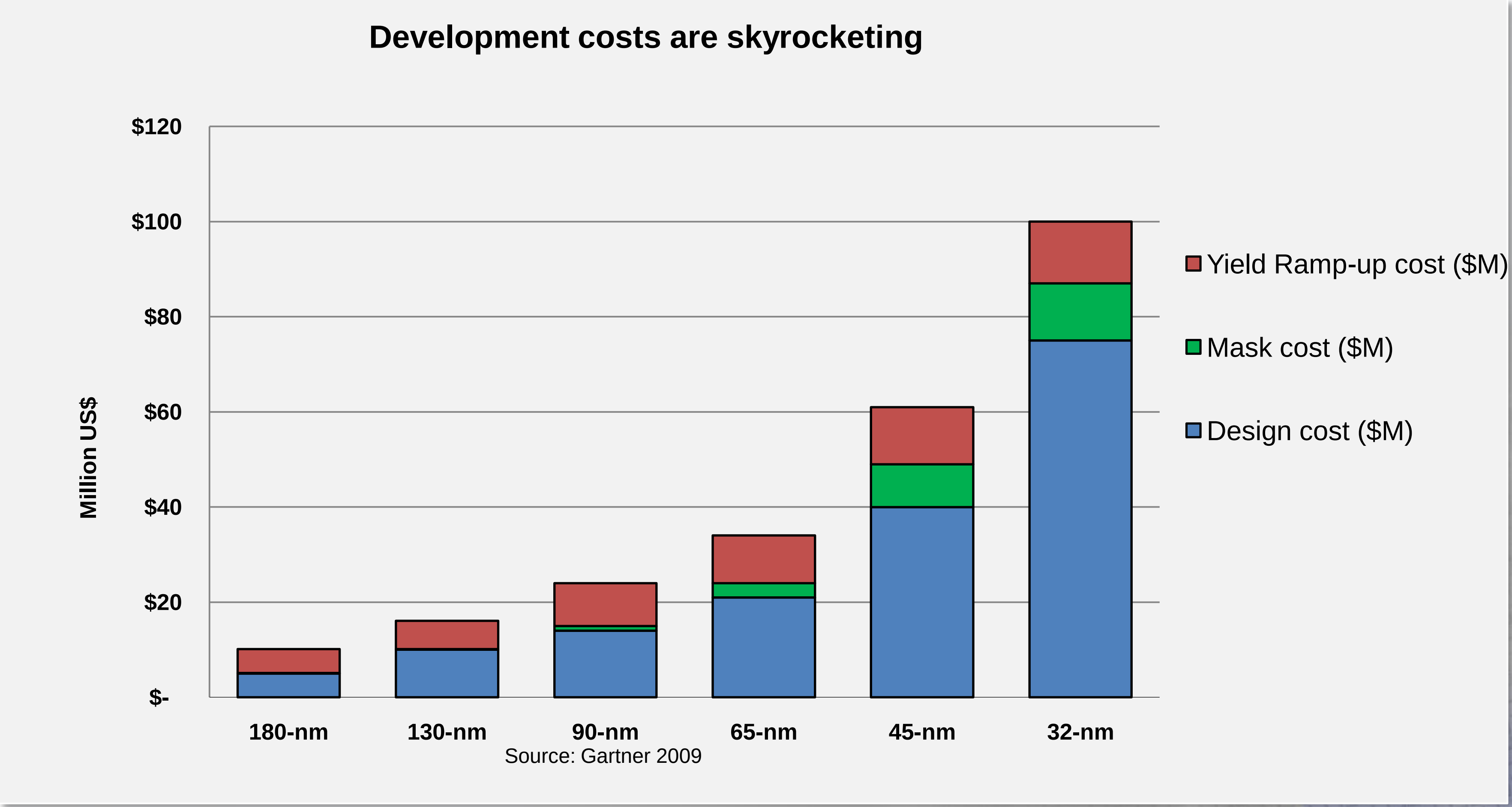
IEEE Nanotechnology Event
November 16, 2010

THE BEST PATH FROM IDEAS TO PRODUCTION SILICON™

Agenda

- ASIC development ~~is becoming~~ has become prohibitive
  - But is there a practical alternative?
- Make it programmable, or you won’t make it
  - The good news
- The specter of interconnect for programmable logic devices
  - The bad news
- The challenge of abstraction
  - The real opportunity

Trouble in ASIC-land



Source: Gartner 2009

Programmability: a lesson from history

- The most important event in the history of the software business was…
- Interactive development at the desktop!
- Remember sending punch cards to a central computer center?
  - Wait 24 hours for a printout, debug, and try again…
- There was no real software industry in that era
- Large-scale software development was almost impossible
  - IBM 360 OS, Apollo moon shot, ?

The rising value of programmability

- Explore alternatives, <u>and try them</u> at your desktop
  - Architectural exploration
  - High-performance verification

- Respond to new customer requirements or competitive threats

- Amortize the cost of developing silicon
  - Standard parts but with unbounded customization
  - Add unique value through system architecture and system software

- Learn from how people use (standard) microprocessors

- Add unique value through system architecture and software

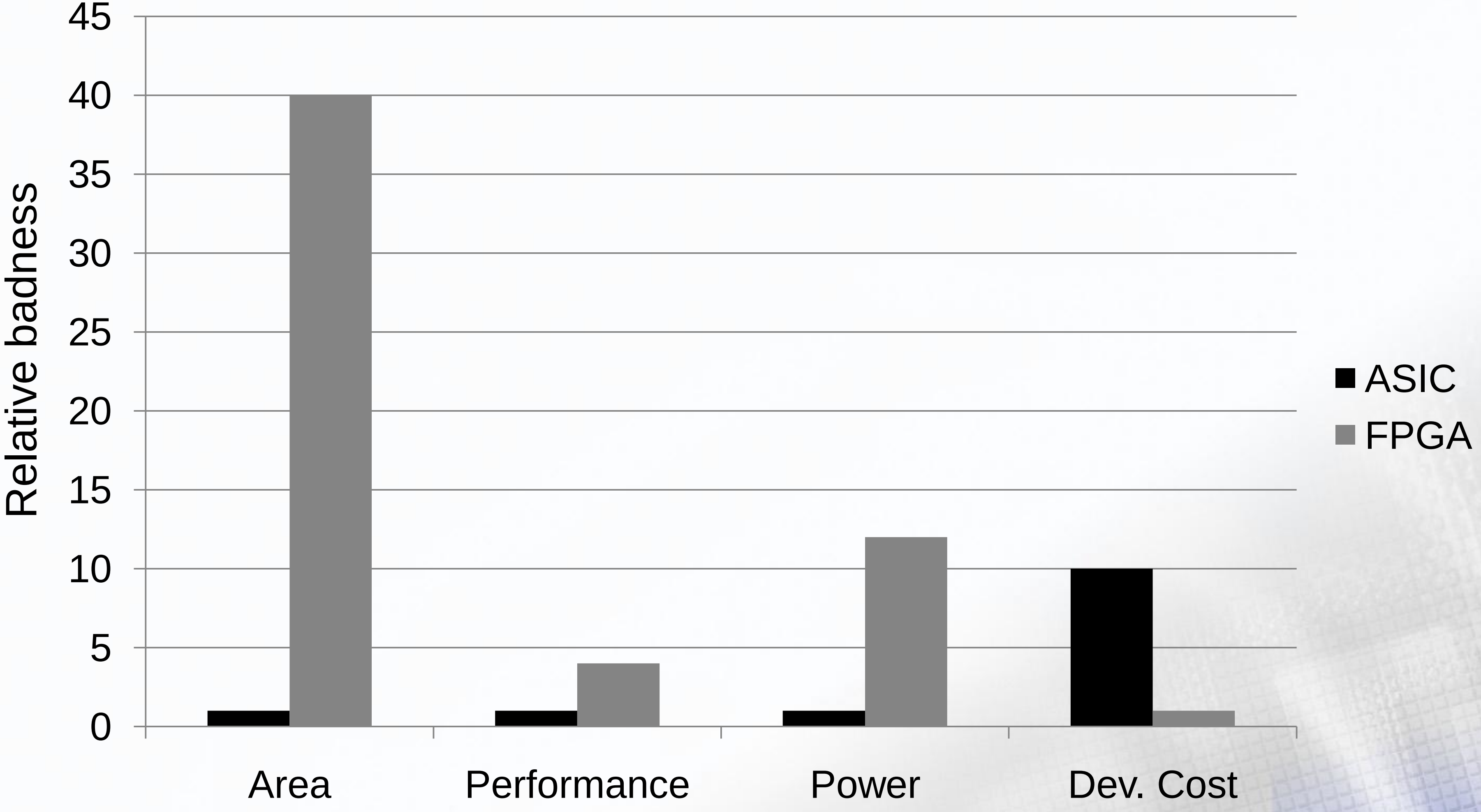
The benefits of programmable logic devices

- Mainstream high-end FPGAs
  - No upfront cost!
  - Up to ~200K LUTs (≈ 3 Mgates) for $200 in volume
  - Up to ~500K LUTs (≈ 7 Mgates) for $1000 in volume
  - Up to ~300 MHz performance
  - Up to 48 high-performance SerDes plus ~900 general-purpose I/Os

- Same family of chips can serve many applications, many markets

- Mainstream higher-volume FPGAs
  - No upfront cost!
  - $5-50 unit cost
  - 10K-100K LUTs (≈ 150Kgates – 1.5 Mgates)
  - Up to ~150-175 MHz performance

FPGAs have their own challenges…



I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs", *International Symposium on Field Programmable Gate Arrays (FPGA)*, Monterey, CA, 2006, pp 21-30

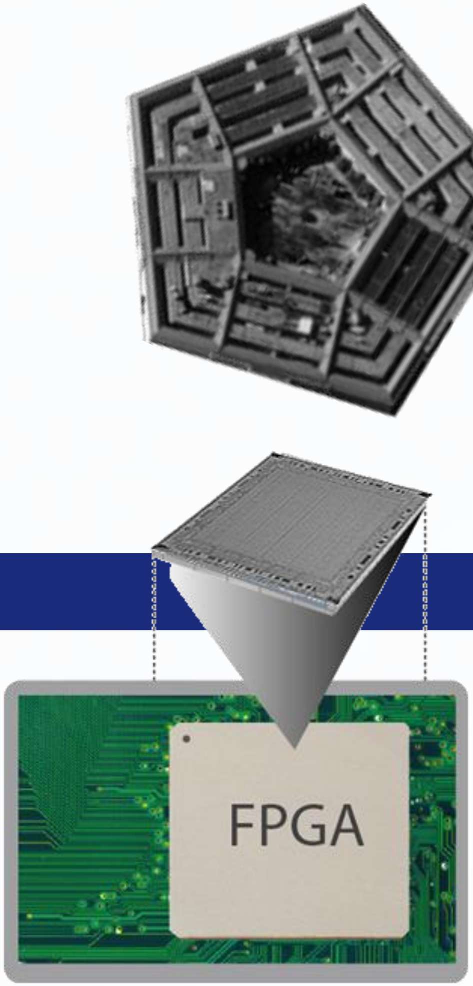
The specter of interconnect on programmables

- Programmable device must support every possible netlist
  - Vs. ASIC, which has just one
- Rent's Rule: more logic → much more interconnect
  - Super-linear growth of interconnect requirements

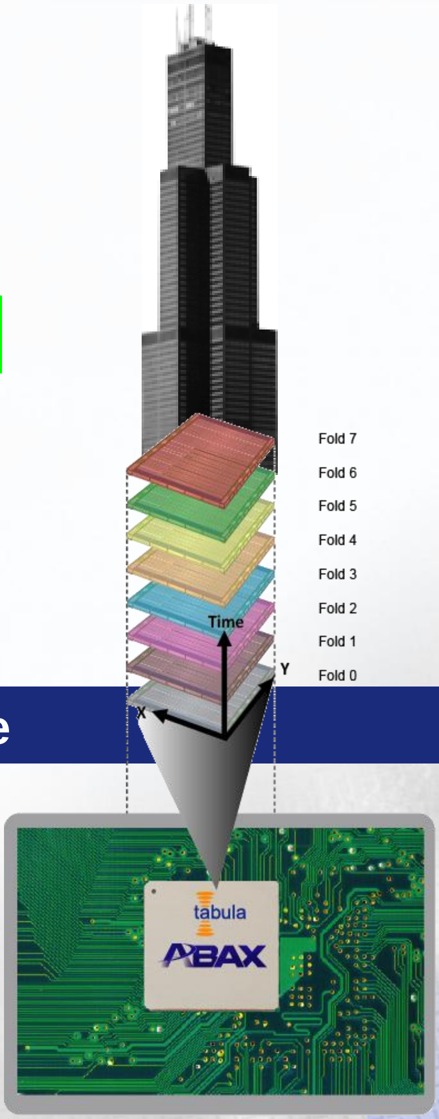
- Up to 90% of FPGA real estate is interconnect
- Up to 70% of FPGA delay is interconnect
  - Had to spend exponential area (4-LUT → 6-LUT) to try to manage it
- >10x more power from interconnect than from LUTs
- >10x more power from wires than from routing MUXes
  - Support for all possible netlists is costly…

- I/O bandwidth requirements increase every generation
  - How can the fabric support the additional, high-performance interconnect?

Towards a programmable device that can actually compete with ASIC



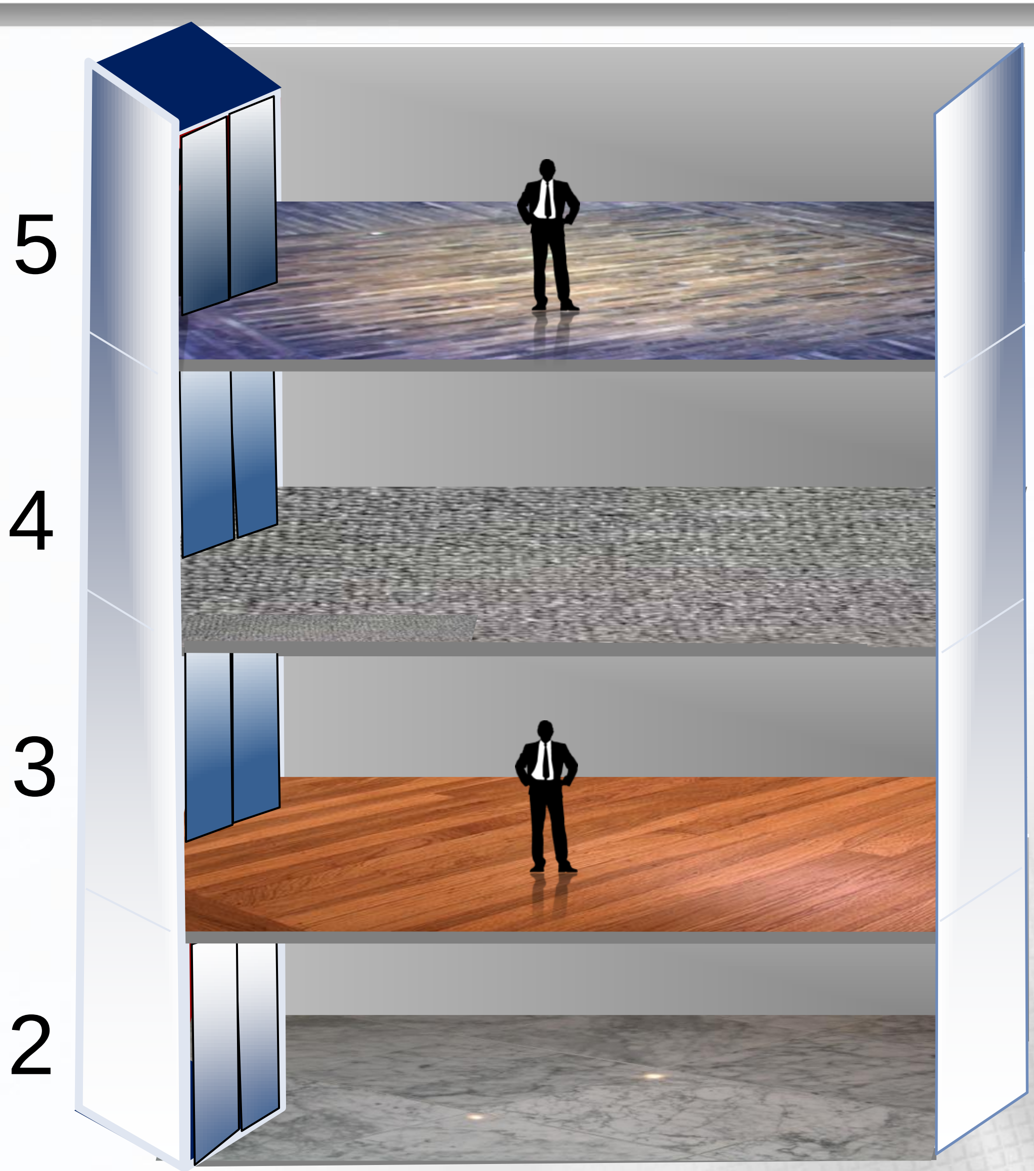
← Vs. →



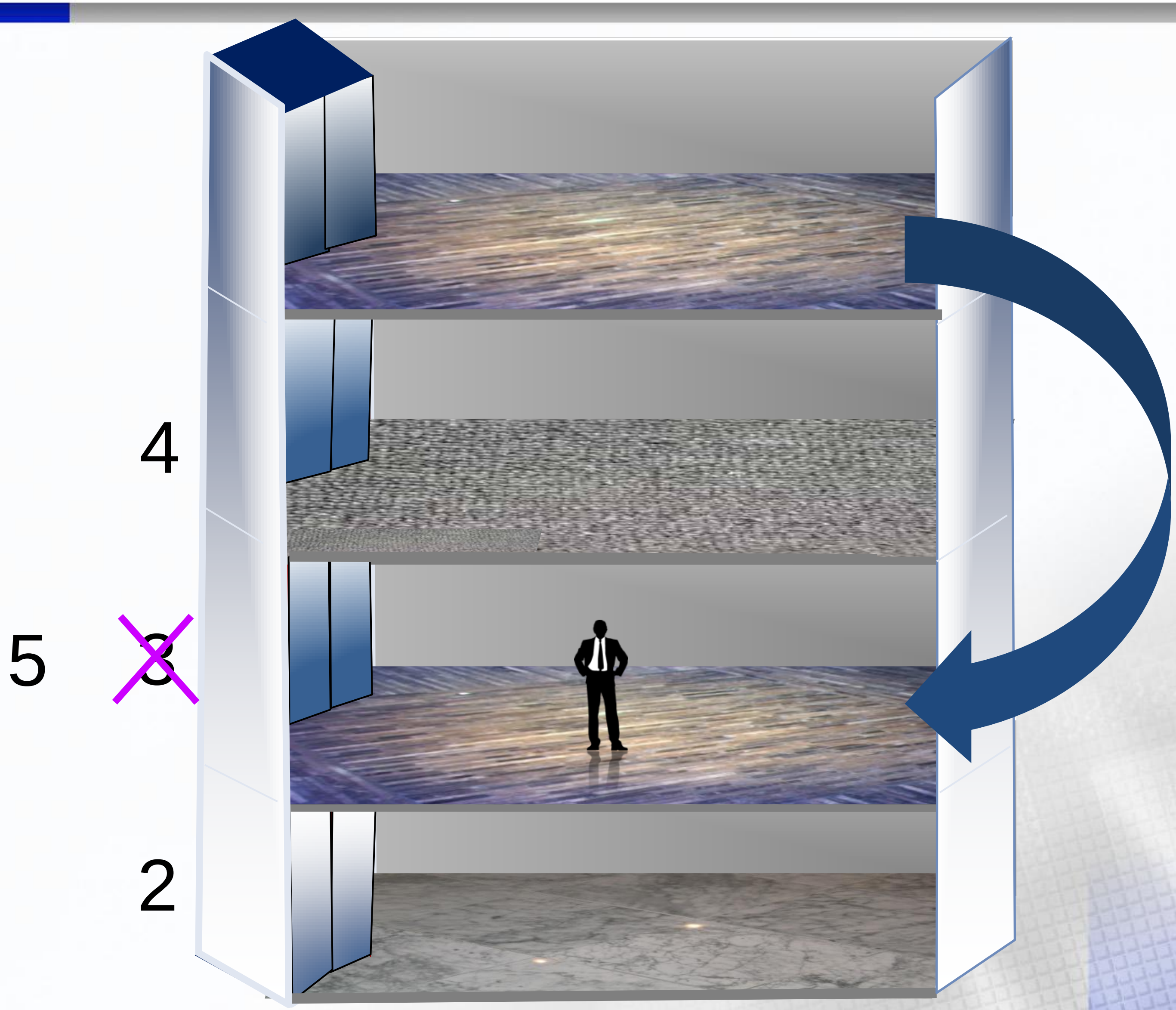
Silicon = $1B per acre

Reducing chip area
is critical

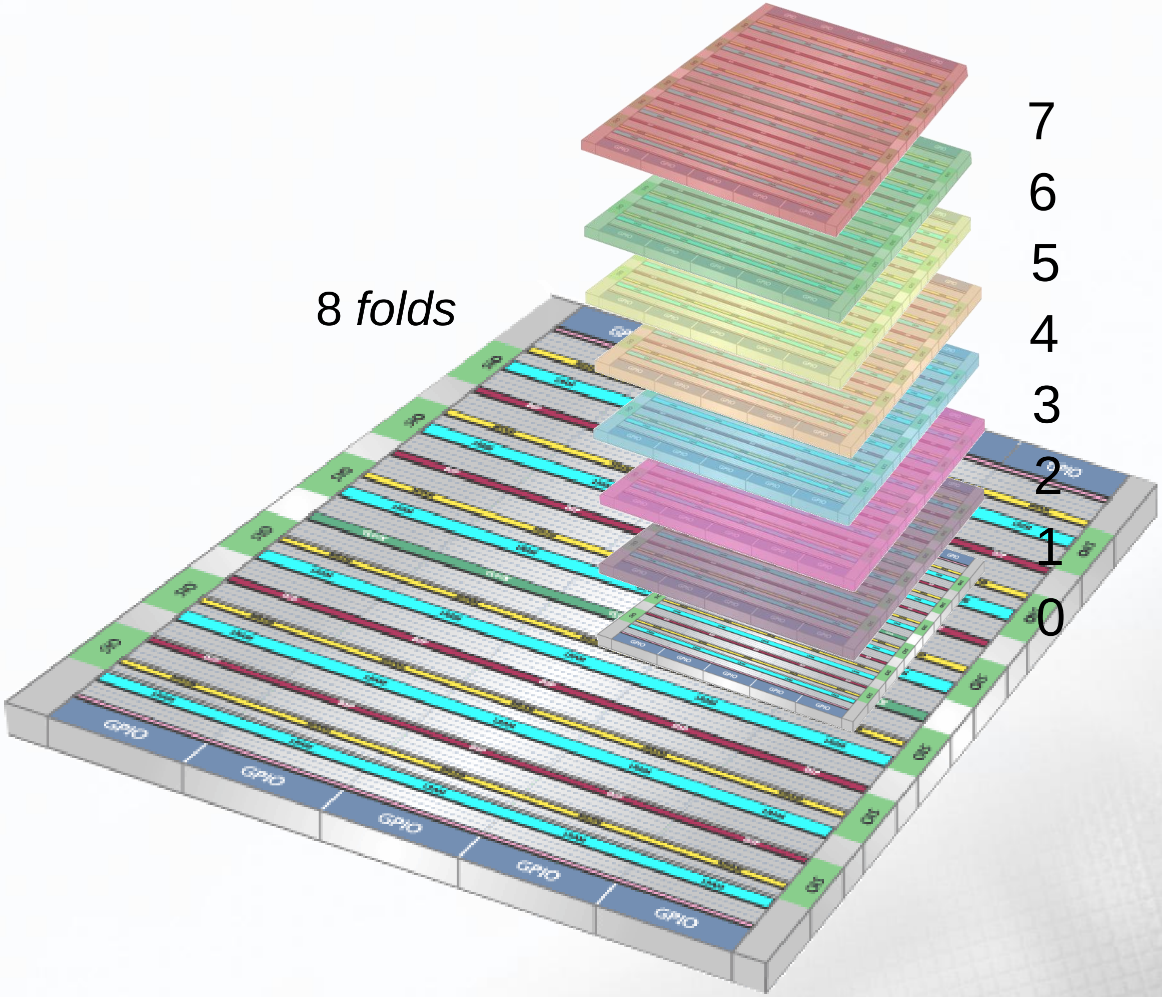
Moving in the third dimension



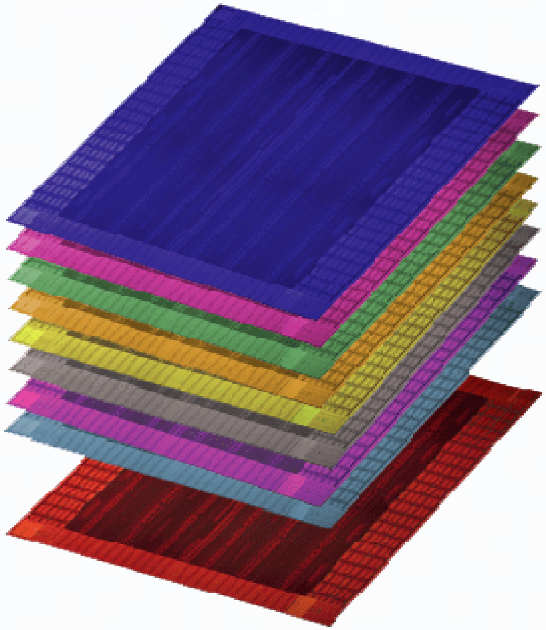
Moving in the third dimension, reconsidered



3PLD: a better programmable device than FPGA

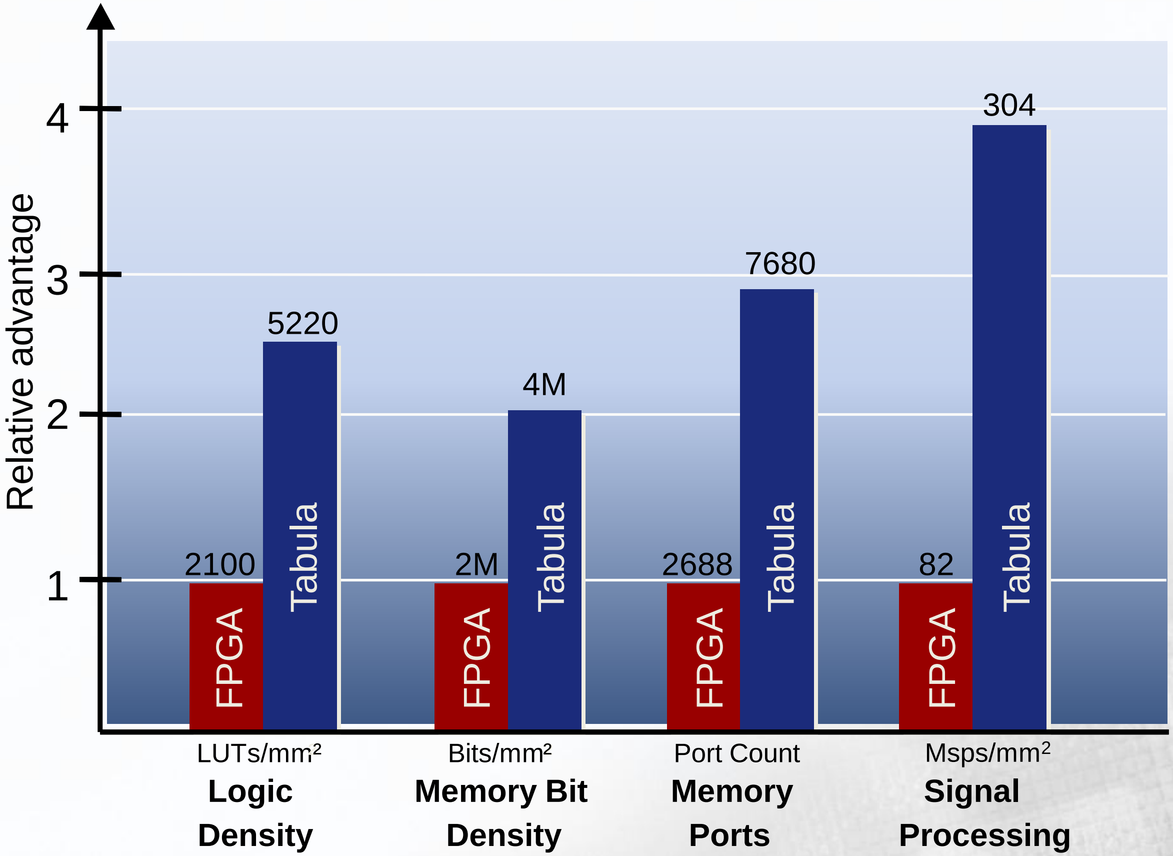


Spacetime™ vs. 40nm high-end FPGAs



Spacetime Advantages vs. FPGA

- 2.5x LOGIC DENSITY
- 2x MEMORY DENSITY
- 3x MEMORY PORTS
- 4x DSP PERFORMANCE



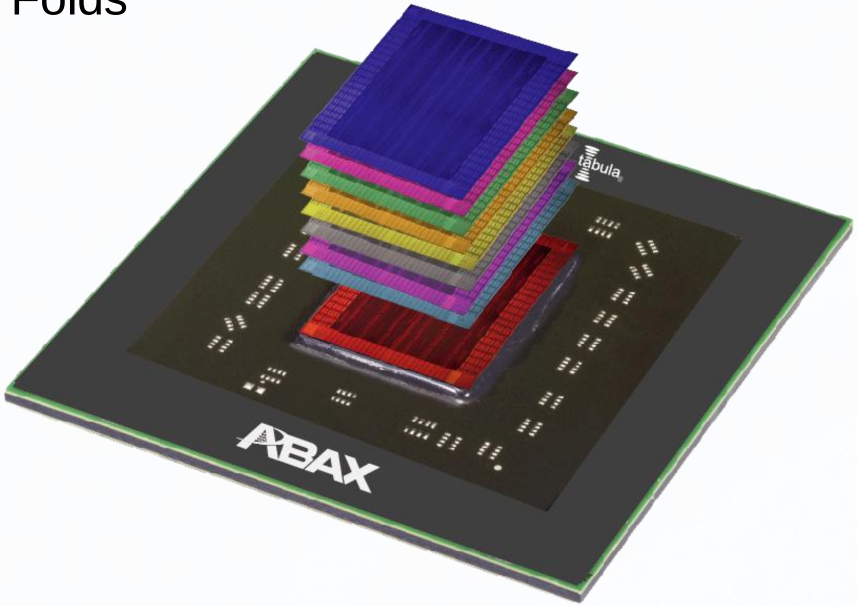
global sources EE Times Asia "…capability unmatched by traditional FPGAs or CPLDs."

"…unmatched capability and affordability." EE Times europe

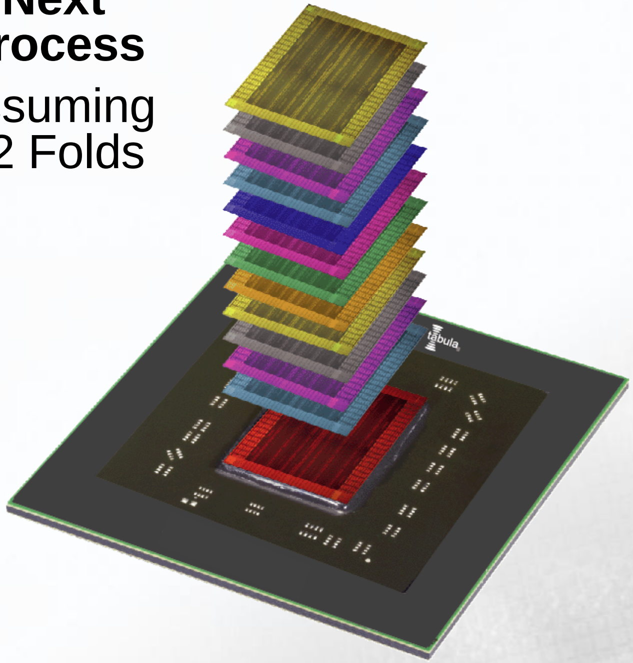
global sources EE Times India "…can surpass performance of FPGAs or CPLDs."

More than Moore’s Law

40nm
8 Folds



Next process
assuming
12 Folds



- 40nm — New process density increases density by ~2x per generation
- 8 folds — New process speed increases our density by additional ~1.5x

Raising the level of abstraction: the big prize

- Physical design and verification for ASIC/ASSP impedes abstraction
- Programmable devices beg for a software-like programming model
- This is the big prize!

- We can put billions of transistors on a programmable chip
- We can moderate the interconnect requirements with Spacetime
  - More than Moore's Law benefits from process advancement
- We've already built production Spacetime H/W and RTL-to-silicon flow

- But…
- Designing large systems demands a higher level of abstraction
- It ain't C (or C++ or Java or SystemC or SystemVerilog or …)

Thinking about designing large systems

- EDA keeps trying to make hardware look like C
  - C-like syntax for HDLs
  - “High-level synthesis”, “Silicon compilers”, etc.
- C’s model of computation looks like a single μP, not custom hardware!
  - Serial computation, one thread, (over-)scheduled by the programmer
  - All computation from (register1,register2,op) ➔ register
- To design large, reliable systems in hardware
  - Find a different software model that still looks “software-ish”, but…
  - Allows for automatic translation to fine-grained hardware (which C doesn’t!)
- Spacetime makes this much easier
  - Software based on “pure functional programming” can be automatically translated to Spacetime hardware
- Large systems at advanced process nodes will need software advancements
  - As well as hardware advancements

Summary

- ASIC development costs are prohibitive

- FPGAs address development costs, but are…
  - 20-40x larger area than ASIC
  - 2-4x slower than ASIC
  - 9-12x higher power than ASIC

- FPGAs are dominated by interconnect
  - Even next-generation FPGAs will not address this fundamental problem

Summary, cont.

- Spacetime offers a programmable fabric that addresses interconnect
  - 5x price/performance advantage vs. FPGA ☺
  - FPGA-like power ☹ (at 40nm…)
  - Unique, more-than-Moore's-Law advantage

- Spacetime changes the game for logic chips in advanced processes
- Advanced process + Spacetime = first credible PLD alternative to ASIC
  - ASIC-competitive area ☺
  - ASIC-competitive performance ☺
  - Power penalty vs. ASIC ☹

- Programming model would change the game again
- Advanced process + Spacetime + programming model >> ASIC