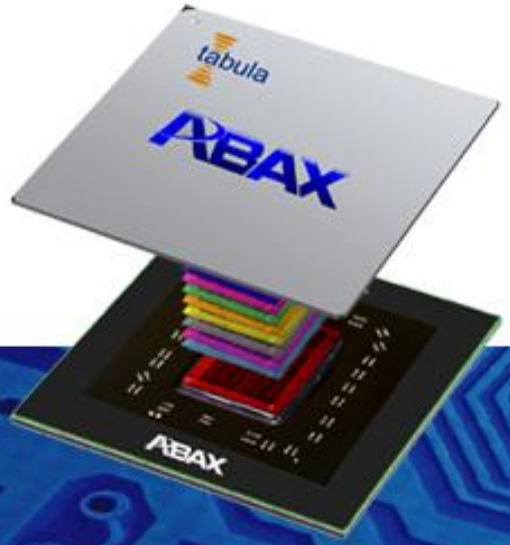
Programmable logic devices in 2032?

Steve Teig
CTO, Tabula

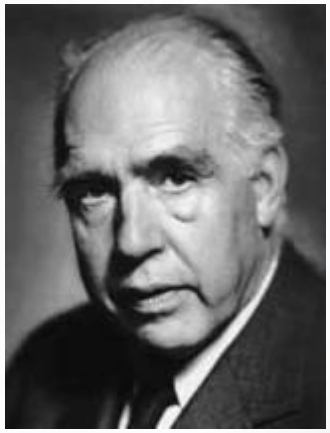


February 22, 2012

THE BEST PATH FROM IDEAS TO PRODUCTION SILICON ®

Prediction is very difficult…

… especially about the future.



Niels Bohr (1885-1962)

20 years is an eternity in technology

- Some (e.g., Ray Kurzweil) believe that all technologies accelerate exponentially
  - Is that true?
  -
- 19

1990's
SMS, Web browsers (1993)
MP3 files (1994)
GPS (1994)
Amazon.com (1994/5)
DVDs (1995)
Consumer digital camera (1995)
HDTV (1996)
Google (1998)
Viagra (1998)
Blackberry (1999)

Selected aspects of future PLDs – my view

- Som
  -
  -
  -
  -
- … s
  -
  -
- … a
  -
  -
  -

day
Ds
ntin
oint

Programming and software

Language	Verilog
Abstraction level	Very low
Data movement	Explicit
Safety	Very error-prone
Support for concurrency / parallelism	Explicit and operational, hard to reason about
Computational model	RTL
Scalability	Very poor

Serial / parallel computation

- Serial vs. parallel is the wrong dichotomy for the programming abstraction
  - Needlessly operational
  - Presumes a needless simultaneity
- Temporally dependent vs. temporally independent
  - Dependent: A must precede B
  - Independent: Don't care whether A precedes B
- Temporal dependency can almost always be determined automatically…
  - E.g., B's input depends on A's output
- …if destructive write is forbidden (or sufficiently sequestered)
  - E.g., in "*pure*" languages such as Haskell, in which "a = a + 1" is meaningless
- Only I/O must explicitly specify dependency

- Pure computations can be automatically scheduled by the compiler
  - Can automatically exploit idle hardware to improve throughput and/or to reduce power
- Pure computations compose → are easier to reason about → scale

Debug, observability, and verification

- Essentially all large software has bugs – lots of them
- 10-15 Mg designs (~1 M LUTs) ≈ small ASSPs ≈ very small programs

- Debug and verification already dominate ASSP design
- As we scale 100-1000x, verification will be everything

- Current debug technologies are a good first step… but still primitive
  - FPGA debuggers require design modification, recompile, difficult (impossible?) timing closure, significant extra real estate, very low-level view of design

- Need observability of every value in the user's program (à la simulation), but…
- …on the device, at speed, non-intrusively
- Need abstraction of implementation – views – to match designer's (programmer's!) mental model

Throughput: the need for speed

- Bandwidth needs are likely to increase rapidly for quite a while
  - Video, 3D video, holograms, 3D avatars, who knows?
- Processing inside and to/from wireless devices will likely continue to increase rapidly, too

- Massive parallelism is one piece of the puzzle… but is not nearly enough
- High-speed serial processing (4+ GHz?) must not be ignored
  - Amdahl's Law
  - Most computations are temporally incompressible, although…
  - Already have throughput at 2 GHz

- Ultra-high-bandwidth memory is an even more vital piece of the puzzle
- So is low memory latency!
- So is very high memory capacity

Performance: from memory → machine learning

- Not just 80 GB / sec (off-device) or 2-3 TB / sec (on-device) sustained
- 3-5 TB / sec throughput from off-chip: e.g., 10K TSVs @ 4 GHz
  - A killer app for chip stacking: ½ of (2012) Wikipedia per second
- 100+ TB / sec throughput on-chip with 1 ns latency
  - Necessary for ~25 TFLOPS sustained throughput
  - ~1/3 of (2012-vintage) Library of Congress holdings… per second
- Since on-chip throughput >> off-chip throughput, must have lots of on-chip RAM
- At least 150 MB but possibly >1 GB
- Since on-chip throughput >> off-chip throughput, on-chip computation is cheap
  - E.g., extreme "data compression"
- Machine learning is fundamental
- E.g., transmit (derived) models and operations vs. explicit pixels/voxels

Power efficiency

- GPU: ~550 GFLOPS at ~250 W = ~2 GFLOPS / W
  - Sustained, double-precision
- At least 100 GFLOPS / W in the next 20 years
  - 20 GFLOPS / W should happen in the next few years

- Can we drastically improve power efficiency?

- What if we compromise on exactitude most of the time?
  - One wrong pixel per month… or week… or minute … or second?
  - Occasional retransmission (or re-reception) of a network packet

- Assertion: if power efficiency is what matters most, we need to trade exactitude
- Heisenberg-like principle characterizing intrinsic energy cost of precision

Power reduction via exactitude reduction

- Bits cost power → exactitude costs power
- Despite reversible computing (theory), practical computing appears to require some energy per bit
  - E.g., to mitigate thermal noise, SEUs, etc.
- Biological systems achieve energy efficiency by requiring exactitude only rarely
  - E.g., DNA transcription
- Everything else is approximate
  - Random motion of molecules in cytoplasm, blood cells in blood
  - Walking motion, visual processing, thought…

- For really low-power computation, need to use exactitude much more selectively
- ECC? High-precision arithmetic? Triplicated logic? – nope!
  - At least, not pervasively

- Long-term future of programmable devices depends on power-exactitude tradeoff

Summary

- Prediction 20 years into the future is pretty hopeless
- That said, my current thoughts include the following:

- Programming model will be a pure (or mostly pure) functional language
  - Won't be RTL or C or imperative or object-oriented
- Debugger/verification technology will advance enormously
  - Full observability at speed will be the norm
- Performance depends on memory throughput, latency, and on-chip capacity
- When off-chip memory is the bottleneck, data compression is crucial
  - Sophisticated machine learning → approximate models vs. raw data
- Power efficiency → compromise of exactitude
  - Heisenbergian principle of intrinsic energy cost per bit of exactitude
  - Approximate almost everything; pay energy for more exactitude only occasionally
- Virtualization provides infinite capacity at the cost of latency

- PLDs have been surprisingly stagnant over the last 20 years
- Let's do a lot better in the next 20 years