



# Programmable Storage Controllers Permit Rapid Response to New Technologies

Chris Bergman - Burlywood


# What do we mean by "Programmable Storage Controller"?


- Software running on a processor?
  - ARM, Tensilica, etc.
- Configurable hardware engine/sequencer?
  - ECC engine, ONFI/Toggle bus, etc.
- Reconfigurable logic platform?
  - FPGA

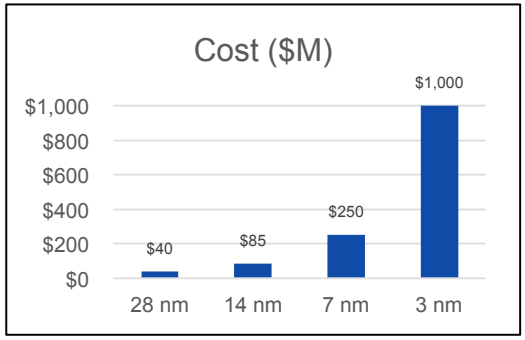
**All of the above? Yes!**


# Why is programmability important?


- Increasing NAND node pace and densities
  - 64 layer->96 layer->128 layer 3D NAND (18 months -> 12 months)
  - SLC->MLC->TLC->QLC
  - Lost opportunity if can't deploy on latest nodes on time
- Complexity of the new nodes/densities
  - Programming algorithms
  - Higher error rates (and what we have to do about it)
  - New flash manufacturing entrants in China
- Higher Performance Expectations (NVMe)
  - Requires more hardware automation
  - But flexibility needs to be maintained to support new media and features
- New Applications
  - Computational Storage, Hybrid solutions, Open Compute
  - End customer applications change over the life of a given storage device
- New non-volatile media technologies
  - MRAM, 3D XPoint, …

# What's wrong with the current model?






- Prohibitive ASIC costs
  - Fab NRE
  - Design Engineering
  - Tools/Verification
  - Lost Opportunity Costs
- Lead times
  - Fab (40-100 days)
  - Design/Verification (6-24 months)
  - Packaging, assembly on circuit board, circuit board test (weeks)
  - With a programmable controller this can be weeks to months
- Less Flexibility
  - Stuck with a fixed feature set and capabilities
  - Not quickly scalable
  - Need to guess what may be needed 2-3+ years in advance (more risk)
  - Imagine finding a HW issue during RDT!

**Cost & lead times amplify the risks**


# What's changed to make this possible?


- More powerful and integrated processor complexes
  - Multiple 32/64-bit ARM cores
  - Flexible, performant interconnect between cores and logic
- Increased effective gate counts
- Native support for SATA, SAS, PCIe PHYs, DRAM controllers, etc.
- More on chip memory
- More effective gates and capability per $
- Pushing process to sub 10nm
- All of this allows an FPGA to be used in the way it was meant to be
  - Traditional ASIC staging used FPGAs, but in ways that reduced overall performance and limited feature set

**FPGAs are no longer only a development tool**


# Why is the programmable approach faster?


- Fab lead times
  - Hours vs. 1-3 months
- Firmware/hardware co-development
  - Staged, incremental development
  - Don’t need to guess and wait for ASIC to return or use crippled staging vehicles.
- Complementary verification and validation
  - Use ASIC techniques where appropriate
  - Use true in system test for full validation
  - Imagine an ASIC that had millions of real-time test hours prior to tape out!


# What are some misconceptions about a Programmable Controller?


- Higher Cost
  - Per component cost offset by fab NRE, design NRE, ASIC tool cost, lost opportunity cost, multiple chip turns
  - Less of a total percentage of the BOM at higher capacities (media dominates)
  - NRE for leading edge process nodes amortized across multiple customers and markets
- Higher Power
  - Less of a total percentage of the power budget at higher capacities
  - Higher performance systems are rarely idle.

**Ideal for today's rapidly evolving data centers**


# Summary


- Programmable controllers…..
  - Allow more rapid development and deployment of new media technologies, architectures, and features
  - Reduce risk
  - Reduce overall solution cost
  - Allow INNOVATION!
- It's real!


# References


- https://semiengineering.com/big-trouble-at-3nm/
- https://semiengineering.com/battling-fab-cycle-times/
- https://semiengineering.com/fpgas-becoming-more-soc-like/



# Backup/Reference Slides

# Examples



## Configurable Controller