CURTISS-WRIGHT

Defense Solutions Division

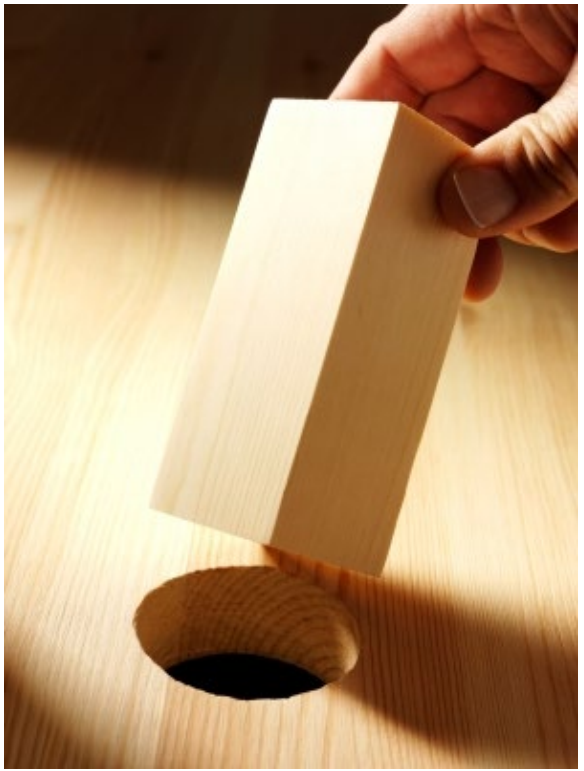
Standard Interfaces and the Benefits of Heterogeneous System Architectures

Jacob Sealander, Chief Architect, C4ISR Systems

CURTISS-WRIGHT

Open Standards

- We've been over this before…
- For the benefit of the warfighter, we need to make sure we're focusing on Open Standards for:
  - Good Interoperability
  - Strong Industrial-base
  - Tech Refresh
  - Fast Fielding (3rd offset)



…but what else?

A High Level View of Processing Types

CPU

- General Architectures for General use
- Optimizations for various common tasks
- Lightly parallel, mainly pipelines in multiple high performance cores

GPU

- Specialized architecture for focused uses (graphics)
- Highly optimized for specific task
- Extremely parallel architecture of many moderate performance cores

FPGA

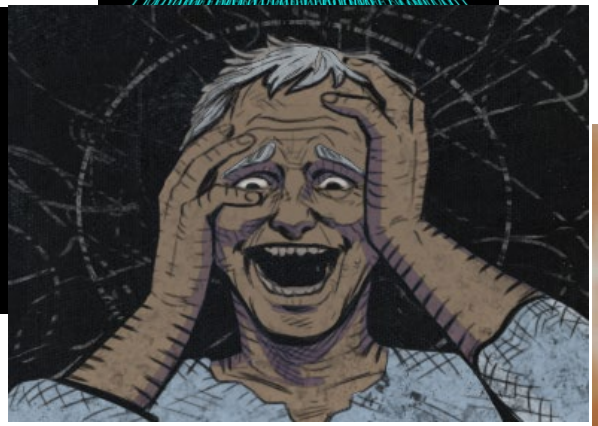
- Flexible building blocks composable for specific applications, including modest CPUs
- Architecture design largely up to user
- Very low latency paths possible

ASIC

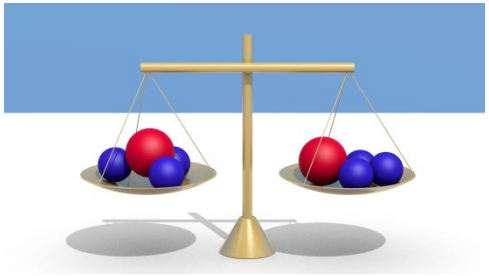
- Application Specific
- Highly optimized
- (Should) work better than a more general approach – whatever "better" means (power, cost, size, etc.)

The System Integrator's Problem

Requirements



How do we do something to meet the needs for now and future using stuff from the past, now, and future in such a way that we do what we need and make sure we accommodate what someone else will want…eventually?



Can we handle all the data?

Heterogeneous System Architectures – what's the point?

- The broad technology world is embracing the use of different processing technologies for different optimizations
  - CPU: general, cloud, virtualization
  - GPU: graphics, machine learning
  - FPGA: low latency, machine learning, security, offload, network
  - ASIC: Any of the above… but optimized
- The technologies are increasingly used in some sort of mixed grouping – hence the "heterogeneous"
- Lots of goodness, but adds complexity

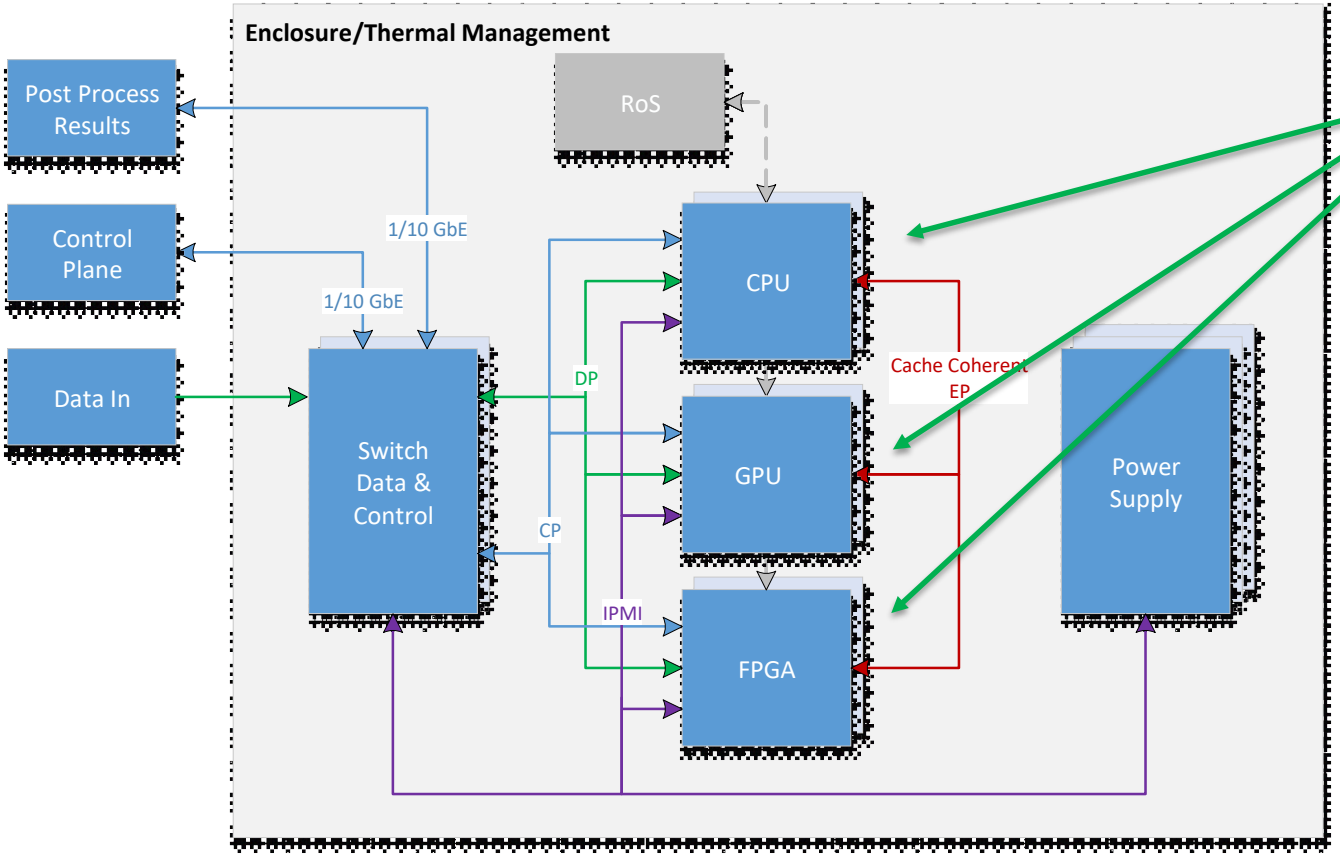
The Key Interfaces (all standardized!)

- Moving Lots of Data in and out of the system
  - Networks, big pipes
- Controlling the system
  - Networks, fast pipes = low latency
- Coherent processing & data within the system
  - Networks…sort of, but getting better (RDMA / RoCE)
  - PCIe…with enhancements, and more coming (CCIX, CXL)
  - New on top of old (GenZ on PCIe & IEEE802.3 PHY)

What are we looking for?

- Applications are converging… heading to a "C5ISR/EW" box!
  - CMOSS / SOSA are a big part of this, leveraging OpenVPX standards
- Build for these without doing this →
- Take advantage of the inherent flexibility of Heterogeneous Architectures

An Example Heterogeneous Architecture



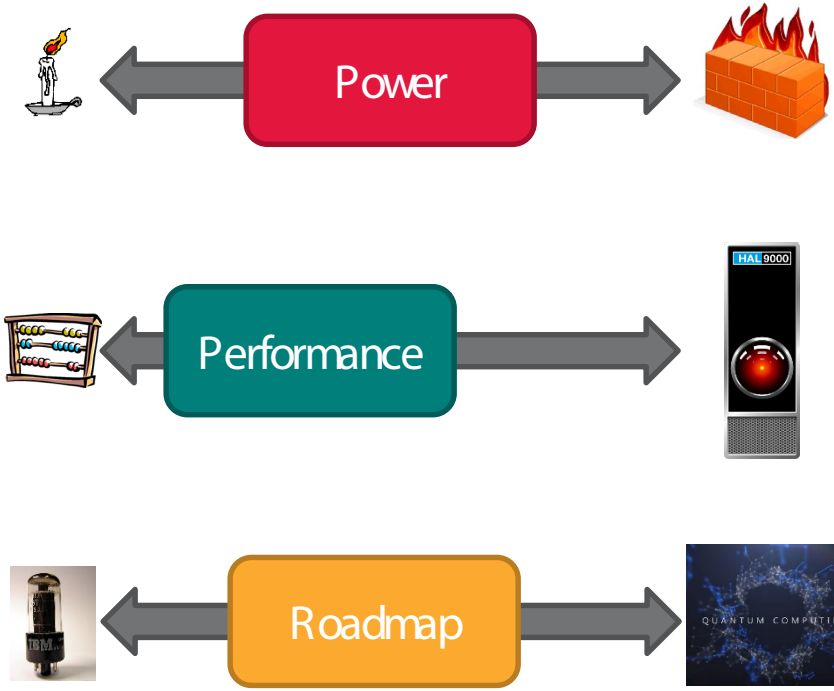
Note mix of technologies are essentially interchangeable

Consider:

- Overall power & thermal
- Application optimization
- Technology Availability & Roadmaps

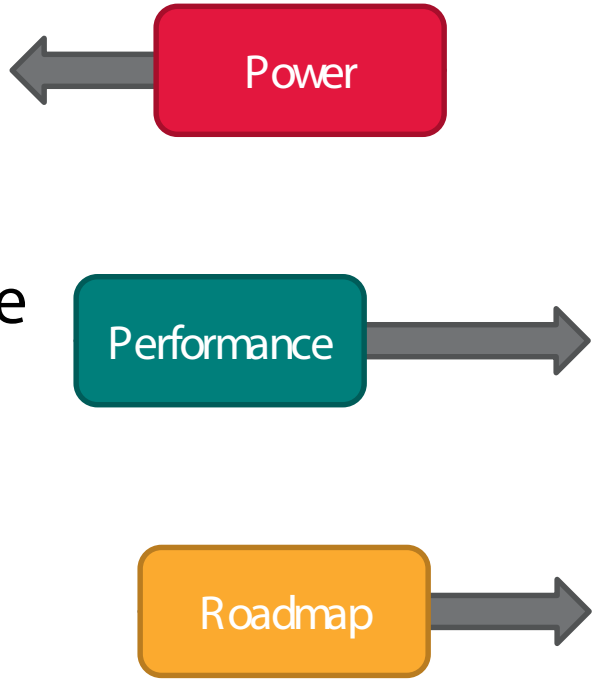
What's the value with all this?

- Every system has constraints
- Every system has key requirements / performance goals
- Every system is limited by what's available now

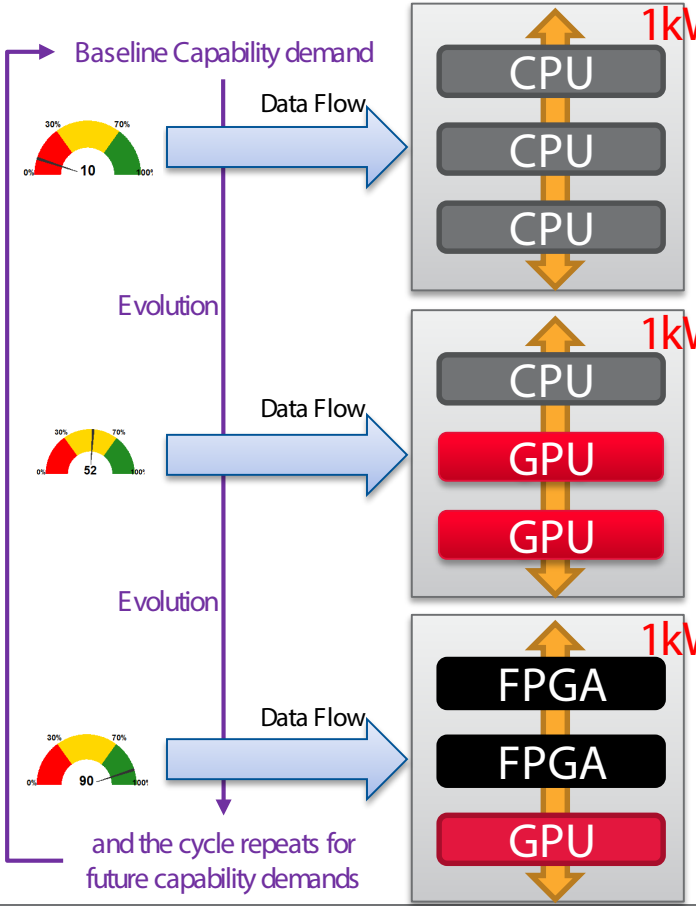


Optimizing through choice and standards

- Choosing technologies for the application while knowing the constraints are fixed
- Understand that performance gets better with roadmaps, often at same power consumption (Moore's Law)
- Also understand that performance can get better by optimizing / re-hosting / leveraging software and processing technologies…but this can take engineering cycles (= time)



System Evolution Example



Today: The data flows through the architecture with non-optimized software on general purpose CPUs. Proves the architecture (and backplane, middleware, etc.), but not the performance. Baseline power consumption = locked physical

Near Term: Partial software re-targeting to GPU creates performance improvements leveraging the same architecture. Increases performance, while staying within the same power envelope = avoid physical redesign

Deployment: Complete migration of types, building on previous efforts and newer tech. More optimization for performance in FPGA, downscale of CPU to within FPGA. All while staying within the same power envelope = avoid physical redesign

Application Examples

- Machine Learning
  - Optimization on FPGA and GPU, just run slower on CPU. Key metric = inferences / second / watt
- Electronic Warfare (or high frequency stock trading…)
  - Input / Analysis / Reaction / Output Latency – better with FPGAs
  - Outer Loops and additional management with CPUs
- Mission Processing
  - Many CPUs for virtualization
  - GPU and FPGA available for acceleration of analysis, data fusion, future evolution to include machine learning for analytics

Vision: Same backplane, different card loads

Key Takeaways

- Open Standard interfaces enable flexibility in the type of processing = flexible heterogeneous processing architectures
- Design and build with evolution in mind
- Allow for agile iterations of optimizations and refresh while keeping to key physical constraints (Size, Weight, Power)
- Get the architecture to the field ASAP and refresh the building blocks within the constraints rather than re-designing entire systems and interconnects

Remember…

You can upgrade the computers…

…without upgrading the building!

Jacob Sealander, Chief Architect
C4ISR Systems
jsealander@curtisswright.com

Q&A

www.curtisswrightds.com

