To Embed or Not to Embed, That is the Question

WEIGHING THE FACTORS OF DISCRETE VERSUS EMBEDDED GPUS

ABSTRACT

Product developers often conduct trade studies before selecting components for their designs. One of the components that is often considered for these studies is a Graphics Processing Unit (GPU). GPUs have become a rather ubiquitous staple of most any electronic device or computer; however, the parameters that need to be collected to choose which GPU best fits a given application have grown increasingly complex. This paper addresses both discrete and embedded GPUs by comparing performance, cost, development environments, obsolescence, and certifiability. When all GPU variables and parameters are identified and quantified (or "costed"), a truly comprehensive design trade study can then be conducted by product/platform developers.

INTRODUCTION

In the 1980's and early 1990's, most computer displays were driven by a Central Processing Unit (CPU) that did all the graphics computation and rendering work. As displays developed higher resolutions, faster update rates, and more complex formats, the graphics requirements became more sophisticated, so graphics co-processors or accelerators were created. At the time, the Computer Aided Design (CAD) high-end workstation market[1] was driving the technology. As the video gaming market grew, these workstation components were co-opted to feed gaming systems' insatiable appetite for realism, which spawned the GPU market we have today. GPUs are now in most every electronic processing device we interact with including computers, tablets, phones, automobiles, electronic kiosks, etc. GPUs are in virtually every market space – consumer, automotive, industrial, aerospace, and military. As GPUs have become more sophisticated, researchers in a variety of fields have realized that their many-core design architectures can do a lot more than simply render images on a display. Today, five of the top seven super computers use off-the-shelf GPUs as foundational components in their architectures[2].

In addition to their massive number crunching capabilities, today's GPUs are also being configured to run artificial intelligence algorithms like machine learning and image processing[3] by configuring the GPU as a neural network to do inferencing. The same GPU that renders realistic images at 60 frames per second in your Call of Duty® video game is executing machine learning algorithms to do facial recognition at airports. To meet this broad spectrum of applications, there are GPUs for small tasks and ones for very complex tasks. GPUs come in stand-alone/discrete "chips", or they come embedded with CPUs and other system components in "System on Chip" (SoC) configurations. For most platform and product designers the question is not whether to use a GPU, but which one to use and whether to use a discrete GPU or one that is embedded next to a CPU on the same piece of silicon. The number of variables and the complexities that influence the decision of "which GPU" has grown to include factors such as performance, recurring cost, total dissipated power, compatibility with operating systems and board support packages, driver availability, temperature range, obsolescence, certification paths, and advanced compute features. This paper can guide product developers through this maze of variables by presenting an overview of these factors and the relative trade-offs being made with today's GPU devices.

DISCRETE GPU VERSUS EMBEDDED GPU ARCHITECTURES

The most popular discrete GPUs from leaders in the marketplace employ hundreds of specialized processing cores to render graphics. They communicate with a CPU over a many-lane, high speed bus like PCI Express. They generally provide video output interfaces for up to five or six high resolution displays using industry standards like DisplayPort™ or HDMI. They have engines for video encoding and decoding and, for Multi-Chip Module (MCM) configurations they include Video RAM on the same substrate. Although the GPU cores and/shader engines are programmable, the discrete GPU architecture is relatively fixed; there are no programmable logic blocks available for customization.

Embedded GPUs generally employ fewer processing cores and share memory with the adjacent CPU cores. This "unified" memory architecture is used for everything: program instructions, scratchpad storage, video frame buffering, and more. Video encoding and decoding may or may not be a part of the SoC architecture. Generally, an embedded GPU has only one or two video outputs (most often DisplayPort), but in many cases two or more embedded GPUs can be instantiated in the overall SoC architecture. Embedded GPUs are oftentimes installed alongside programmable logic arrays that provide significant flexibility with system configurations.

PERFORMANCE

Discrete GPUs are built and bred for both 2D and 3D high performance applications. They support the highest shader core clock rates along with memory interfaces that are generally quite broad (i.e., 256 bits wide) and quite fast. Today's discrete GPUs are generally capable of performance/power ratios of around 40 GFLOPs/Watt. They easily support 4K display output resolutions with update rates of 120 Hz or higher. The video encoders and decoders built onto these devices can handle multiple streams of 4K video using H.264/H.265 compression/decompression standards.

Until recently, embedded GPUs were best suited for modest 2D graphics, but that is changing. CPU and GPU companies are putting their most capable GPU architectures into their own line of SoCs. The latest generation of 2D and 3D rendering engines are now integrated alongside an array of multicore CPUs. Embedded GPUs have traditionally lagged discrete GPU performance by a 5:1 ratio but this is now changing to less than a 2:1 performance difference.

RECURRING COSTS

For those applications where recurring cost is a primary driver, there is little debate about the advantages of embedded GPUs in an SoC over discretely implemented GPUs and CPUs. Moderately performing SoCs can be purchased for around $50 compared to discrete GPUs that go for >$100 (in addition to the cost of the necessary CPU). However, issues arise with scaling when trying to get a collection of SoCs to meet high performance requirements. The concept of stringing multiple low performance SoCs together to achieve high levels of graphics rendering has proved to be problematic.

TOTAL DISSIPATED POWER

Because of their high-performance capabilities, discrete GPU devices oftentimes consume considerable power (15 to 125 Watts). For the embedded processing market, the threshold of pain for most discrete GPU devices is generally around 35 Watts of total dissipated power. That number applies to both a discrete GPU and a discrete CPU, so if used

together on the same board, the combined 70 Watts of both CPU and GPU power generally presents itself as a limiting factor for most embedded processing module thermal tolerance levels. If an application requires high performance levels, the design generally must enable an environment that can deal with modules that dissipate these kinds of power levels. It should be noted that just because a discrete GPU has a high TDP rating, it can oftentimes be operated at much lower power levels. Today's market leading discrete GPUs incorporate several power saving features, so predicting thermal performance without an application in mind is very difficult. Power dissipation is affected by image complexity, content update rates, display resolutions, and total number of displays being driven. An example of how the thermal performance of a discrete GPU (AMD's e9171) rated at 35 Watts TDP might actually perform in the real world (driving one to five high resolution displays) is shown in Figure 1.

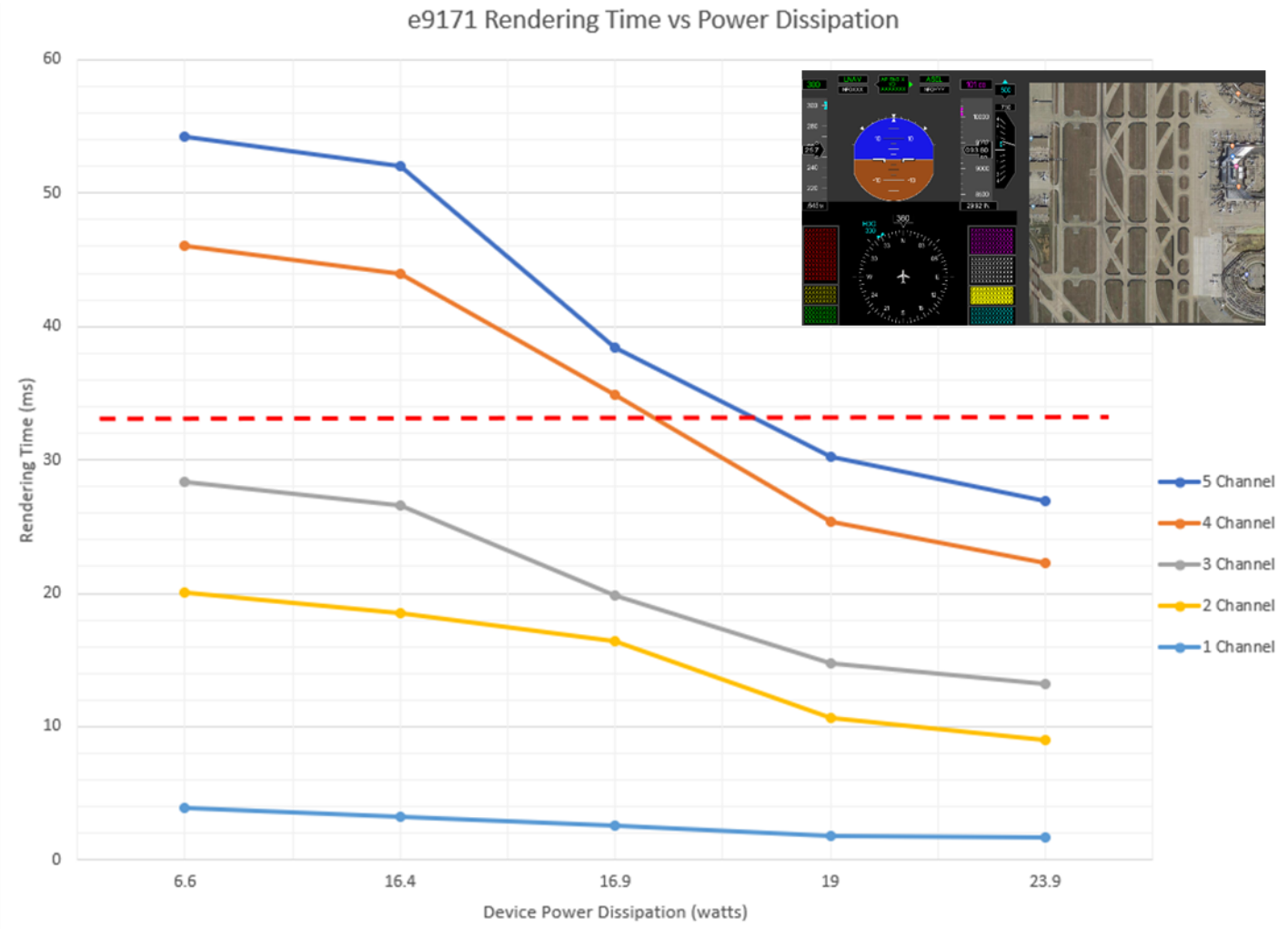

Figure 1: AMD E9171 Discrete GPU Power Dissipation Benchmarks

Embedded GPUs present significant power savings when compared to discrete GPUs, thus greatly simplify thermal design for embedded processor products. SoCs with moderate performance GPUs oftentimes run in the 10-to-15-Watt range.

OPERATING SYSTEMS

Embedded processing systems generally require the use of Real Time Operating Systems (RTOSs) to ensure determinism, safety, and reliability. RTOS kernels are supplemented by Board Support Packages (BSPs) that contain a variety of drivers unique to the CPU/GPU silicon and the board/module design that they reside in. Although RTOS vendors oftentimes produce a "reference design" for the most popular processors, product developers are regularly required to tailor a BSP to their application. There is an intimate link between GPUs, RTOS/BSPs, and the graphics drivers and libraries that applications use to generate images. Although standard graphics libraries like OpenGL® ES 2.0/3.0 or SC 1.0/2.0 are the friendliest for application reuse, there is often a need to tailor these libraries or drivers to the RTOS/BSP and/or CPU/GPU combination. Further complicating the RTOS landscape are the areas of multi-core operations (with hypervisors and guest OSs), partitioning, and multiple design assurance level support.

When selecting a discrete CPU/GPU or an embedded GPU in an SoC, RTOS/BSP considerations are especially important. An off-the-shelf combination of hardware and software that has already been tested and integrated can save product developers hundreds of thousands of dollars in program costs. When selecting a hardware component (discrete GPU or embedded GPU), product developers/designers would be wise to investigate RTOS/BSP/graphics driver support as part of their trade studies and selection process.

DRIVERS AND LIBRARIES

When graphics application programming interfaces (APIs) were first introduced (e.g. Windows 95), order replaced chaos in the wild west of graphics programming on PCs. For workstation applications, Silicon Graphics in 1992 provided some degree of order with their version of OpenGL. It was not until 2006 when OpenGL was adopted as a standard by The Khronos Group that true stability entered the embedded processor market for graphics. OpenGL, with subsets for embedded systems (ES) and safety critical (SC) applications has provided software developers with significant stability for applications that generate or process graphics. To provide even more platform agnosticism, Khronos introduced a new, low-overhead, cross platform 3D graphics computing API in 2016. That API, Vulkan®, provides significant advantages over OpenGL: OS-independence, reduced driver overhead, batching capability, better multi-core CPU support, more detailed GPU "exposure", and unified management of kernels and shaders (see Figure 2).

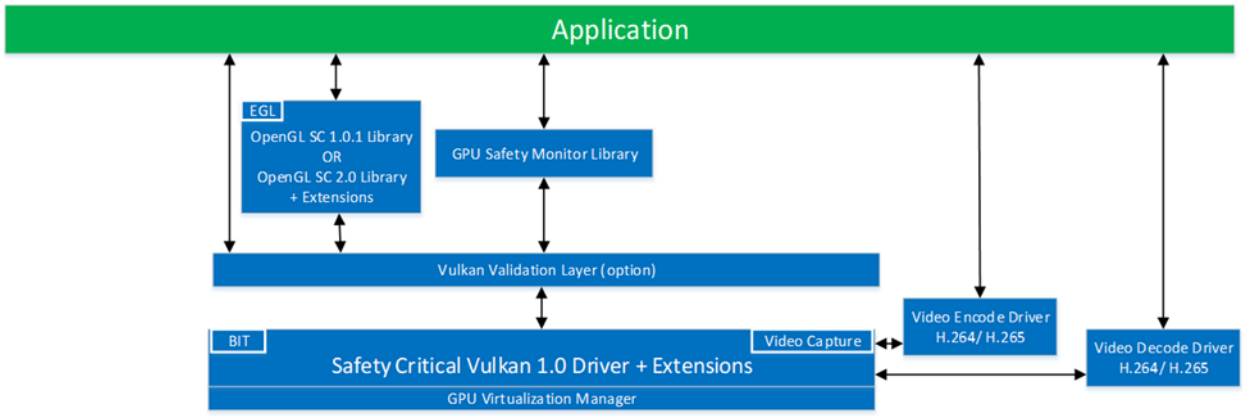

Figure 2: Vulkan Ecosystem Example[4]

Application developers would be wise to consider the graphics driver and library environment when selecting either a discrete GPU or an SoC with an embedded GPU. Starting with a combination (GPU and drivers/libraries) that already has some integration history will go a long way to reducing development costs and mitigating risks. System integrators regularly want the latest and greatest technology with the least amount of risk. Vulkan libraries, capable of hosting OpenGL drivers, represent the best compromise for those two often conflicting objectives. Some examples of GPU types and OpenGL driver support is outlined in Figure 3.

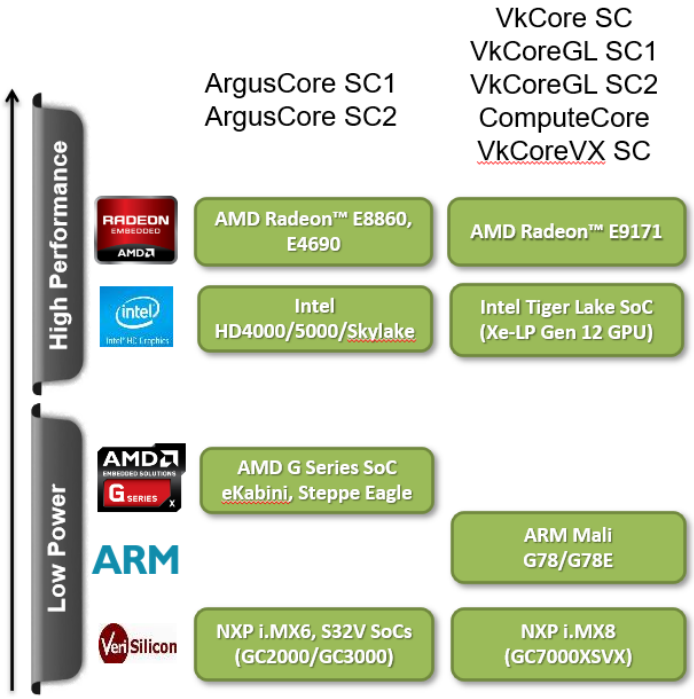

Figure 3: Discrete and Embedded GPU Types Along with OpenGL Driver Possibilities

TEMPERATURE RANGE

Due to production volumes, the primary market for GPU designs (both discrete and embedded) is often commercial applications. These products usually operate in more benign environments than something like a military helicopter. Aerospace and military products usually require components that operate at extended temperature ranges—for example, -40°C to +105°C)—and few GPU suppliers care to go to the expense and trouble of screening their parts for this small market. Some silicon suppliers solved this problem by developing partnerships that allowed a 3rd party to “up screen” their parts to meet the needs of aerospace and military customers.

OBSOLESCENCE

The incorporation of commercial-off-the-shelf GPUs into embedded processors proved to be a double-edged sword for aerospace and military applications. Products in these markets take years to develop and then are deployed for decades (see Figure 4). Although COTS GPUs provide the cutting-edge performance that these markets are looking for, the

technology moves so fast that any one part can easily become obsolete in as few as 18 months. This poses quite a dilemma for aerospace and military markets, in that product developers have to choose to either 1) regularly redesign (and requalify) their products, or 2) purchase large quantities of parts at the end-of-life point in a part's lifecycle.

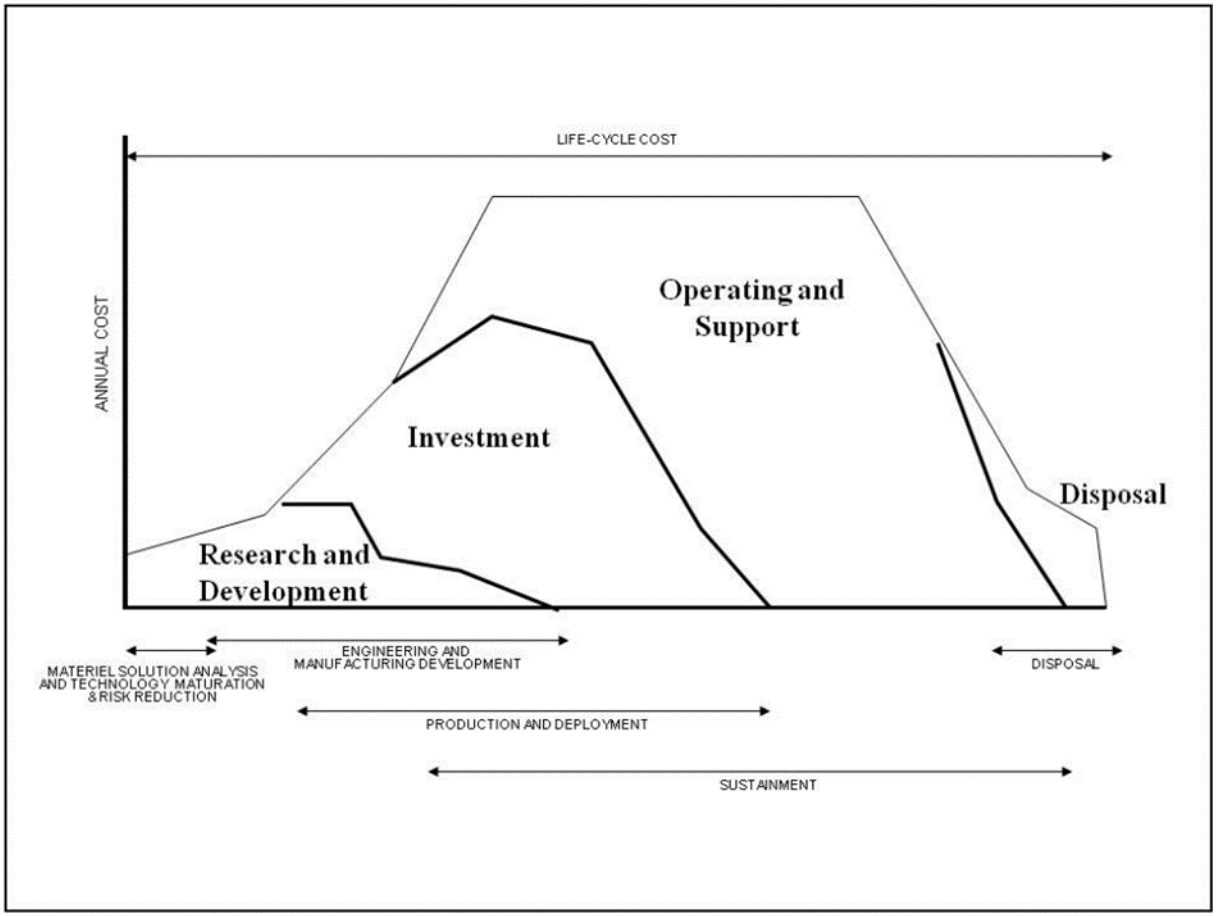

Figure 4: Typical Product/Component Lifecycle for Aerospace Market[5]

In the discrete GPU marketspace, some commercial companies responded to this problem by promising both availability and support for as long as seven years for some products. In the embedded space, others have a track record of supporting certain parts for 10 to 15 years. Furthermore, other companies (like CoreAVI) have stepped up to provide cash-flow friendly purchase and storage options for end-of-life GPUs needing long term support.

CERTIFICATION

For many years, aerospace and automotive product developers designed their products to standards sought by government certification authorities (like DO-178 for the FAA, or ISO 26262 for the NHTSA) by employing process rigor to ensure success. Lately, that philosophy has expanded to include component selection. This comprehensive

"certifiability" approach goes beyond the selection of the operating system to include the graphics libraries/drivers, as well as key hardware components. Fortunately, many silicon suppliers are now providing Functional Safety (FuSa) evidence for certain components to meet the needs of safety standards like DO-254 and ISO 26262. The Khronos Group is supporting safety certifiable standards like OpenGL SC 1.0/2.0 and Vulkan SC. Companies like CoreAVI are bringing all these certifiable components together in a tested and validated "stack" running on Open System Architecture (OSA) hardware that simplifies development risks for platform developers.

ADVANCED APPLICATIONS

Both discrete GPUs and embedded GPUs can be configured to support the convolutional neural network architectures commonly used for artificial intelligence/machine learning (AI/ML) and general-purpose processing or "Compute" applications. In general, performance levels for these functions will scale to that of 3D graphics performance. A discrete GPU with 1,200 GFLOPs of graphics performance will yield six times the AI/ML algorithm performance of an embedded GPU with 200 GFLOPs of graphics performance. Machine learning is an evolving technology but GPUs, as they provide a neural network foundation, are proving to be quite capable of inference engine processing at the "edge" (see Figure 5). But, for deterministic or safety critical operations, machine learning or compute applications that are likely to be safety certified (automotive, aerospace, urban air mobility) the list of possible candidates and environments starts to rapidly narrow. Although algorithms running on a GPU with CUDA or Pytorch using training from the "cloud" may be easy to integrate and test in the lab, there are significant hurdles to overcome before such a configuration can be fielded. AI/ML developers may want to consider foundational elements that are built for certification (like GPUs capable hosting Vulkan and OpenVX™) as a more reliable path to deployment for safety critical applications.

Figure 5: Using a GPU to Efficiently Execute Edge Detection Algorithms

CONCLUSION

In the quest to find the perfect part for a new design, trade studies are conducted to match component attributes with platform/product requirements. In the case of GPUs, which are now commonly used in almost every computer processing product, the extent of the variables to be considered has grown considerably from just a few years ago. Besides the basic choice between a discrete GPU or one embedded in an SoC, there are a multitude of parameters to consider including performance, recurring cost, total dissipated power, compatibility with operating systems and board support packages, graphics driver availability/compatibility, temperature range, obsolescence, certification paths, and advanced compute feature support. The evolving relationships between hardware and software suppliers—so very relevant to GPU selection—further complicate the decision-making process and demands that designers stay up to date with the latest business and partnership developments in addition to technology trends and performance. When all these parameters are identified, quantified, and understood, appropriate design trades can be executed, and component selections made.

REFERENCES

1. Techspot, The History of the Modern Graphics Processor, by Graham Singer, January 7, 2021. https://www.techspot.com/article/650-history-of-the-gpu/
2. HPC Wire, GPUs Power Five of World's Top Seven Supercomputers, by Tiffany Trader, June 25, 2018. https://www.hpcwire.com/2018/06/25/gpus-power-five-of-worlds-top-seven-supercomputers/
3. ZDNet, How the GPU Became the Heart of AI and Machine Learning, Colin Barket, August 13, 2018. https://www.zdnet.com/article/how-the-gpu-became-the-heart-of-ai-and-machine-learning/
4. CoreAVI, Vulkan SC Graphics and Compute, https://coreavi.com/product_category/safety-critical-graphics-and-compute/
5. US Department of Defense, Operating and Support Cost-Estimating Guide, Office of the Secretary of Defense, March 2014. https://www.cape.osd.mil/files/os_guide_v9_march_2014.pdf

AUTHOR

Michael Pyne
Director Strategic Accounts & Solutions Architect

Mike's role as Director Strategic Accounts & Solutions Architect at CoreAVI allows him to bring together the rapidly evolving world of "open" software infrastructures with the safety-critical requirements of both manned, pilot assisted, and autonomous platforms. Mike has over 40 years of experience in Defense and Aerospace markets. Previously, as an engineering fellow with Honeywell, Mike developed cockpit architectures and systems for a variety of airborne defense platforms like the F-15, F-16, F/A-18, C-130, OH-58D, CH-47, and the V-22. Mike's focus on all of these platforms was in using open system architectures for high reliability/mission critical roles in rugged military environments.

Mike is based in Albuquerque, New Mexico, has a BSEE from Brigham Young University, and holds two patents in the area of sensor processing.