Arm's Mali-G78AE Flexible Partitioning: Innovative Mixed-Criticality Support for Safety-Critical Systems

INTRODUCTION

The new Arm Mali-G78AE GPU architecture provides a key innovation in its design: the ability to divide the GPU into partitions that effectively allow one Mali-G78AE GPU to be split into multiple independent mini Mali-G78AE GPUs. These "mini Mali-G78AE" GPU partitions are formed by combining GPU slices, where each partitioned GPU can have one or more slices assigned to it. The partitions provide all the functionality needed to operate as an independent GPU, including their own dedicated connection to the fabric to access memory, allowing each partition to function completely independent of the other partitions.

When a partitioned GPU contains more than one slice, the slave slices have their additional, unneeded hardware disabled to conserve resource usage, allowing the master slice to control the slave slices. This isolated and modular approach allows the hardware to be separated for independent tasks that may have different safety criticality requirements. For example, if a safety-critical application does not have high performance requirements—it is only responsible for a small subset of the main display—it could use only a single GPU slice while, the more demanding, non-safety critical content can use the remaining GPU slices.

GPU Flexible Partitioning provides key benefits in the design of safety-critical systems. In addition to allowing each GPU partition to act as an independent GPU, containing all the required hardware to execute graphics and compute-related tasks, with slices capable of being allocated to partitions as needed, the partitioning management system also enables the slices to be reconfigured at runtime, which allows the system to adjust GPU resource allocation as needed. As these partitioned GPUs act independently, a non-safety critical partition can run a commercial, non-safety critical driver and a safety certified driver can be used on a different partition, enabling mixed critically use cases. The safety critical driver running on a different partition could also composite the content from the non-safety partition, sharing a single display for safety critical and non-safety critical content. This is shown in Figure 1, where a safety critical application (Safety Application Cluster), non-safety critical application (QM Cluster), and a safety island (Partition Manager) have independent AXI control interfaces to the processing cluster and memory.

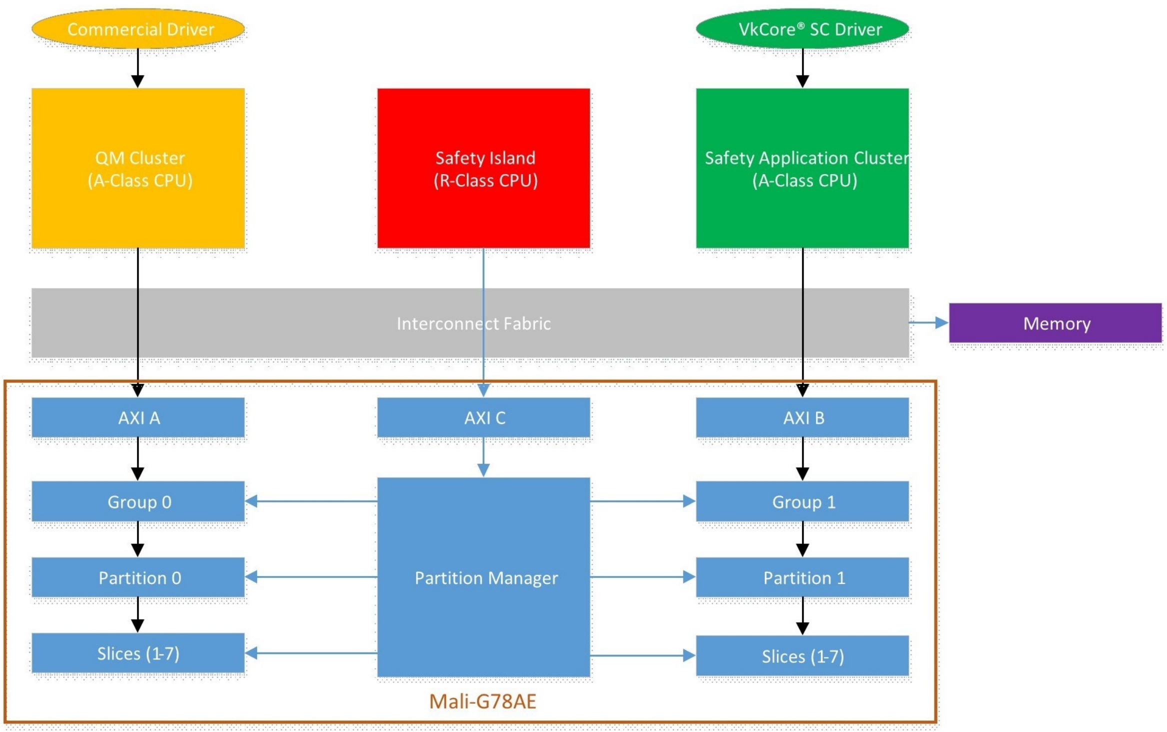

Figure 1: Mixed Criticality Architecture

Safety-critical applications that require minimal interference between very high criticality applications and very low criticality applications typically must rely on the use of multiple GPUs. The reason this is required is simply because most modern GPUs do not provide sufficient mechanisms to limit the amount of rendering and computation time an application uses. To eliminate interface between the applications, multiple GPUs are used so that graphics and compute tasks truly run in parallel and independently. Although this multi-GPU approach works, it has significant down sides with respect to size, weight, and power (SWaP). The use of multiple redundant GPUs increases the overall power usage of the size while also increasing the size and weight to facilitate the use of a second GPU. The Mali-G78AE GPU provides a unique opportunity for application developers to have multiple applications share a single GPU where each application runs truly in parallel, allowing an application to use all the rendering and computation time it needs. In addition to these benefits, a single GPU also provides all the SWaP benefits that come with using a single device.

TYPICAL SAFETY CONCERNS ON GPUS

On a typical GPU there are several concerns regarding safety-critical software. If a GPU is being shared by multiple applications, particularly applications with different levels of safety criticality, we need to ensure that one application cannot influence the execution of another application. These concerns can be broadly categorized to concerns of the “space” domain, and concerns of the “time” domain. For concerns of the space domain, we need to ensure that one application cannot influence the state of other applications. This would include modifying the frame buffer or changing the graphics state. For concerns of the time domain, we need to ensure that a less critical application does not oversaturate the GPU with work or interrupt the execution of a more critical application such that it is not able to update the screen in the expected timeframe.

On a typical consumer GPU, if you want to use it for both a safety-critical workload and a non-safety critical workload, you need to carefully manage how work is assigned to the GPU. Careful monitors need to be put in place to track if the GPU is doing real-time safety-critical work or non-safety critical work. This monitoring is responsible for determining if the GPU has become oversubscribed and is unable to complete the safety-critical task in the needed timeframe. All this careful monitoring and management of the GPU can take a sizable amount of CPU time to process. This can create a significant burden to system integrators who want to use a single GPU for mixed criticality applications.

HOW HARDWARE SEPARATION ADDRESSES SAFETY CONCERNS

The Mali-G78AE GPU takes an innovative approach to solving these safety concerns. The fact that the GPU can be split into isolated partitions means that the concerns about sharing a GPU across safe and non-safe applications are avoided. Each mini Mali-G78AE GPU has its own L2 cache, job manager, tiler, memory management unit, and shader cores. This stacking of hardware in each slice gives them an incredible power to operate independently without influencing the other slices on the GPU.

With this partitioned approach, careful monitoring of the tasks running on the GPU is no longer required. Instead, we can dedicate a group of slices for safety-critical applications. These applications get dedicated access to their own GPU partition and do not compete for shared resources like a conventional GPU does. In addition, each partitioned GPU group has its own memory management module. This means that each partitioned GPU can only access the memory that is assigned to it and cannot interfere with the memory or graphics state assigned to other partitioned GPUs.
This unique approach to GPU design allows the Mali-G78AE GPU to provide unprecedented safety advantages in the commercial GPU space. The Mali-G78AE GPU can safely isolate parts of itself for a safety-critical task without worrying about major interference from other parts of the GPU.

The Mali-G78AE GPU is offered in conjunction with CoreAVI's suite of safety critical GPU drivers and compute libraries in the VkCore Functional Safety Suite for Mali-G78AE. The Mali-G78AE GPU flexible partitioning feature enables a safety critical application to take advantage of CoreAVI's high performance safety critical graphics driver; it facilitates implementation of safety critical applications by using CoreAVI's support for safe compositing allowing mixed-ASIL human machine interfaces (HMIs). Mixed-ASIL HMIs allow simultaneous display of non-safety critical application graphics content with safety critical graphics content composited onto one display. This framework provides application developers with the best of both worlds, all integrated together in one high-performance, feature-rich safety-critical platform.

SUMMARY

Arm Mali-G78AE flexible partitioning in conjunction with the CoreAVI VkCore Functional Safety Suite is a game changer that allows silicon suppliers and system integrators the greatest flexibility in supporting mixed-criticality systems in safety markets including automotive, industrial, and avionics covered by the ISO2626, IEC61508, and DO-1278C/254 safety standards. If you are interested in more detail on Arm Mali-G78AE, please click here.

For more information on CoreAVI's VkCore Functional Safety Suite, please contact sales@coreavi.com.

AUTHOR

Lucas Fryzek
Field Application Engineer

Lucas Fryzek has been a software developer with CoreAVI for 5 years and has recently transitioned over to the role of Field Application Engineer (FAE). Lucas has extensive hands-on experience in developing embedded software systems deployable in DO-178C safety critical environments and has worked in tandem with many of CoreAVI's largest leading customers to deliver safety critical solutions, including those specializing in the division of graphics workloads across multiple CPU cores. Lucas' experience with OpenGL SC and Vulkan APIs allow him to provide expert technical support and guidance both to CoreAVI's customers and internal team.