Eight Concerns with Graphics Processors in Safety Critical Applications

Introduction

This white paper examines the concerns with using COTS Graphics Processors (CGPs) in applications for accelerated 2D and 3D safety critical graphics rendering. Given that the concerns are similar for avionics, railway, automotive and other environments requiring safety critical applications, this paper will discuss these issues from an avionics perspective as the avionics industry has identified specific CGP concerns. This white paper will discuss each of the identified concerns along with techniques that may be used to address or mitigate the concern.

CGPs have been available on the commercial market for many years and can provide functionality that previously needed to be implemented in software. The use of CGPs significantly improves graphics performance and capabilities by reducing processor overhead while enabling further application features. While CGPs are COTS devices, they are currently considered out of scope for COTS device safety certification guidance and have CGP specific guidance (which will most likely be challenged or changed in the future as the CGP market continues to evolve). For example, CGP enabled devices are now available with extended availability similar to NXP's SoC devices or select AMD devices that are part of Core Avionics & Industrial's (CoreAVI) 20-year supply program. Also, certain SoC devices with built-in CGP functionality are developed following a functional safety process to include built-in safety mitigation features and design process documentation to ease the certification effort.

There are two guidance documents identifying the concerns regarding the use of CGPs in avionic safety critical applications:

- EASA certification memorandum CM-SWCEH-001 Issue 01 Section 10
- Certification Authorities Software Team (CAST) position paper CAST-29 which is referenced when the FAA is the certification authority.

The concerns raised in each of these is essentially the same with the CM-SWCEH-001 having one additional area of concern.

For clarity, CGP will be used to reference the entire graphics capability which is divided into two major areas of function. The first is the Graphics Processing Unit (GPU) which performs all the graphics rendering to create the image in memory (frame buffer). The second is the display system which takes the image from memory (frame buffer) and drives a physical interface (electrical interface at physical pins) with standard timing characteristics, such as DVI, to interface with the display(s). No distinction is made between discrete CGP and SoC-integrated CGP.

Concerns Using Graphics Processors in Safety Critical Applications

Item a - Hazardously Misleading Information (HMI)

One of the main concerns involved in the development of a safety critical display system is the potential for displaying Hazardously Misleading Information (HMI). HMI could come in the form of incorrect or missing alerts, or incorrect or missing information such as speed or frozen display. If HMI is not flagged, it may result in actions or inaction with potentially hazardous consequences. A believable, incorrect display is typically a more severe hazard than a display that is completely failed (blank) or obviously incorrect.

This white paper will discuss the following potential HMI contributors along with some associated mitigation techniques:

a. Hardware failures within the CGP

b. Design errors within the CGP

c. Failures or inappropriate responses to external events, such as EMI, lightning, high operating temperature, or "out of nominal" input power specifications.

Hardware Failures Within the CGP

It's difficult to substantiate a safety case for detecting hardware failures within a highly complex COTS component such as a CGP. Manufacturing testing, extended temperature screening and module and system level testing prior to deployment can help detect failing devices before deployed use. However, the area of concern here is hardware failures that may develop during use and deployment in flight. Fortunately, there are some current and new techniques to help detect hardware failures at run time. To simplify this concern, the following discussion will treat the GPU and display system separately.

Figure 1 illustrates a general GPU high level architecture diagram. The programmable architecture of modern GPUs is supported by the OpenGL® SC 2.0 safety critical API which was introduced in 2016. What this means is that there is a large legacy library of safety critical graphics software based on the previous OpenGL SC 1.0.1 API which supports a fixed function graphics pipeline (i.e. not programmable). With modern GPUs no longer supporting the full fixed function pipeline, companies like CoreAVI provide OpenGL SC 1.0.1 driver library suites to support this legacy software based on fixed function pipeline emulation using the programmable capabilities of the GPU. Figure 2 shows the OpenGL SC 1.0.1 Fixed Function Pipeline Architecture. The operations in red are emulated with the transform and lighting which is emulated on vertex shaders with the remainder emulated on fragment shaders.

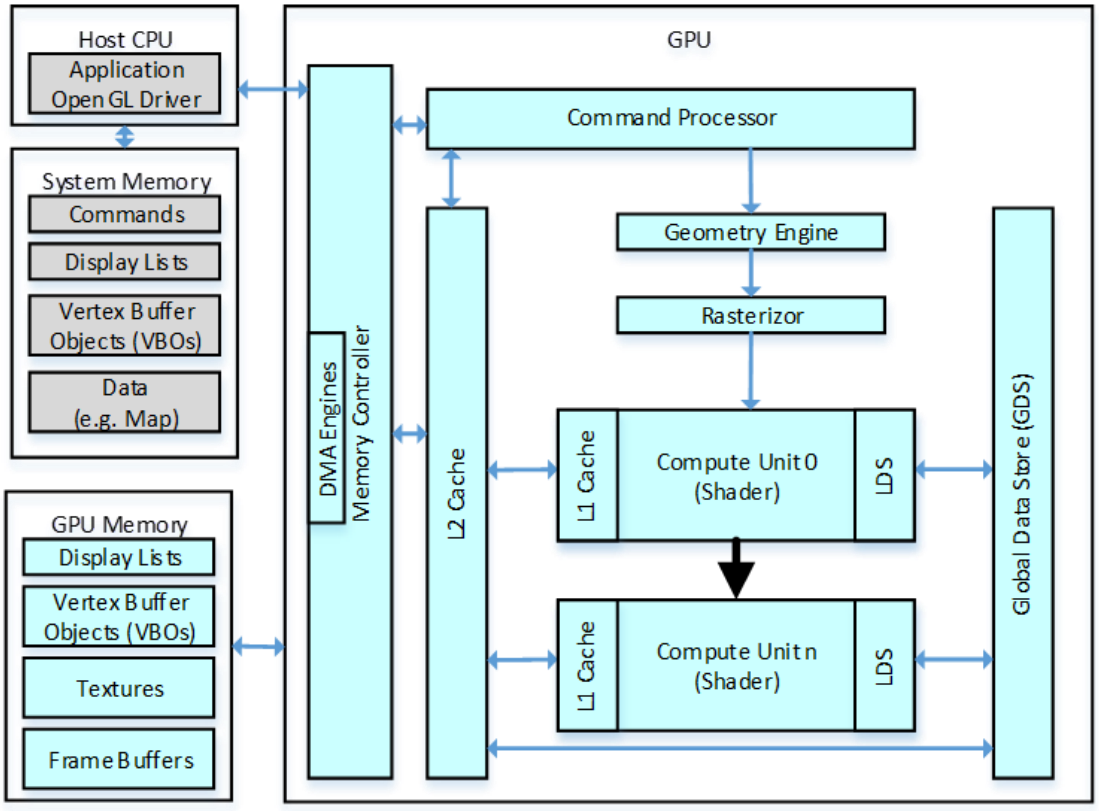

Figure 1: General GPU High Level Architecture Diagram

Basic Built-In-Tests (BIT) can verify the interface from the host CPU to the GPU - that is, that the command processor is processing commands and the interface to the graphics memory. CoreAVI provides this basic BIT functionality within its OpenGL driver libraries.

The brute force approach is to read pixels back from the frame buffer and verify critical areas were correctly rendered. This method adds to CPU overhead and presents challenges in monitoring within a suitable timeframe. For example, a speed indicator needle position is based on a sensor input; therefore, the CPU would also need to calculate where the needle should be drawn independent of the GPU to verify correctness. This can be slow and prone to false failures if minor differences in pixel locations occur due to computation differences between the CPU and the GPU that otherwise do not cause HMI to be displayed. This issue has lead to other approaches to minimize CPU overhead.

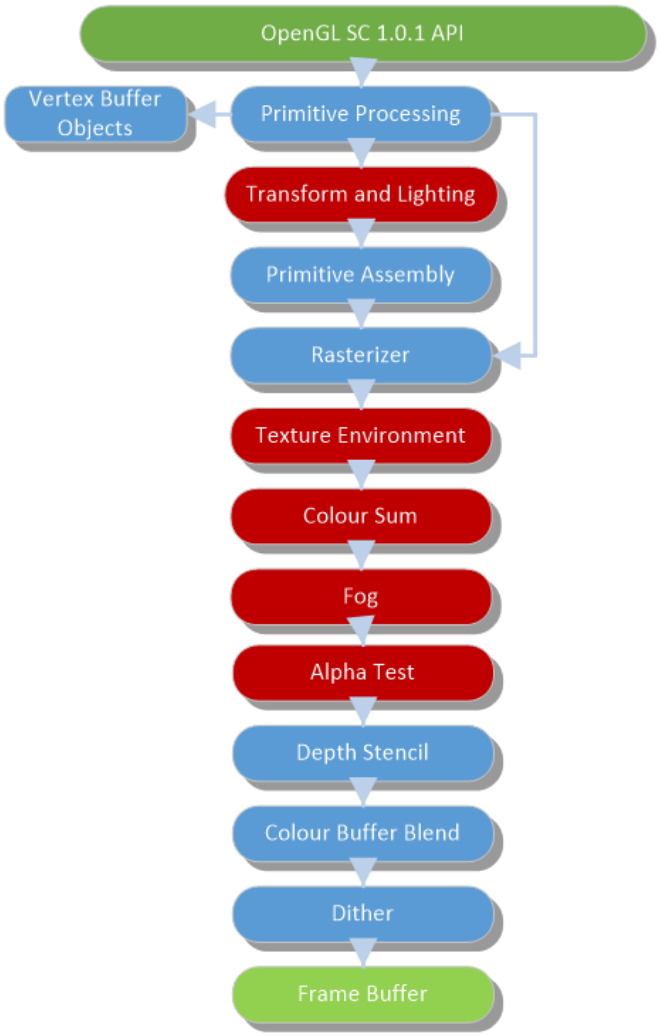

Figure 2: OpenGL SC 1.0.1 Fixed Function Pipeline Architecture

A typical approach to monitor the GPU for hardware failures is to create a known test pattern(s) in the non-display part of the output and verify the test pattern in an FPGA that is added between the output of the GPU and the input to the display, as shown in Figure 3. The test pattern would be created using functions that the application is using, and represents a proxy for GPU operation correctness, verifying that the GPU is performing the functions correctly. Then by inference, if the GPU is performing the functions correctly, the applications' use of the functions would also be correct and the resulting image in the frame buffer is therefore assured to be correct (providing the application has passed the necessary verification and validation testing). This approach has some drawbacks in terms of size, weight, power and cost if it is the only function of the added FPGA.

Figure 3: FPGA-based Integrity Monitor

Another, similar approach removes these drawbacks. This approach consists of removing the need for an FPGA-based integrity monitor by using CoreAVI's TrueCore™, a software only GPU integrity monitor. This is like the FPGA approach in that known small test patterns are created in frame buffer memory and verified. However, the test coverage is defined in terms of the fixed function pipeline for the first release of TrueCore and the verification of the small test patterns is performed independently by the CPU. Test coverage is maximized through highly engineered BIT tests, each utilizing a different combination of functions across different subsets of the rendering pipeline to minimize the number of tests. This enables quick detection of a failure. Therefore, with TrueCore verifying the operation of the fixed function pipeline operations, which inherently verifies the underlying emulation programs running on the compute units (shaders) of the GPU, the monitoring of the GPU for hardware failures is simplified. The Federal Aviation Administration (FAA) has favorably reviewed TrueCore, and the FAA Chief Scientists concur that the product addresses the certification concerns associated with use of a complex COTS Graphics Processor in systems requiring Level A compliance (the most stringent avionics safety critical level). TrueCore is a complete solution from CoreAVI that's delivered pre-integrated with the OpenGL driver library, extending the BIT coverage with a common interface for application control to speed time to market.

One caveat with both the FPGA and TrueCore-based integrity monitor approaches is that the test pattern creation may not be using the same hardware as the application image creation due to the dynamic allocation of GPU resources that is beyond external control of the driver and application. That is, GPUs have functions that control the allocation of internal resources which may not be configured or easily manipulated. These integrity monitor approaches may use the same physical hardware as the application or a parallel hardware path, selected by algorithms built into the GPU, providing the same underlying functions representing a reasonable proxy for correct GPU operation. If additional assurance is required, Figure 13 shows an architecture for an approach to increase assurance that the GPU is not rendering HMI.

Figure 4 illustrates a general high level architecture diagram of a display system. A CGP may have one or more display system pipes. At the heart of the display system is the display controller. Figure 5 shows the functionality of a display controller. The display controller stores enough data in its input FIFO to optimize memory transfers from the frame buffer in CGP memory (typically around 2 KB). The FIFO is drained by the display controller output clocker. When the FIFO reaches the low-level watermark, it is refilled. The display controllers use simple row/column logic to skip rows and columns not in the displayable region. If a CGP display controller buffers a full frame, the person in the loop will not be able to detect a frozen display and another mitigation mechanism is required.

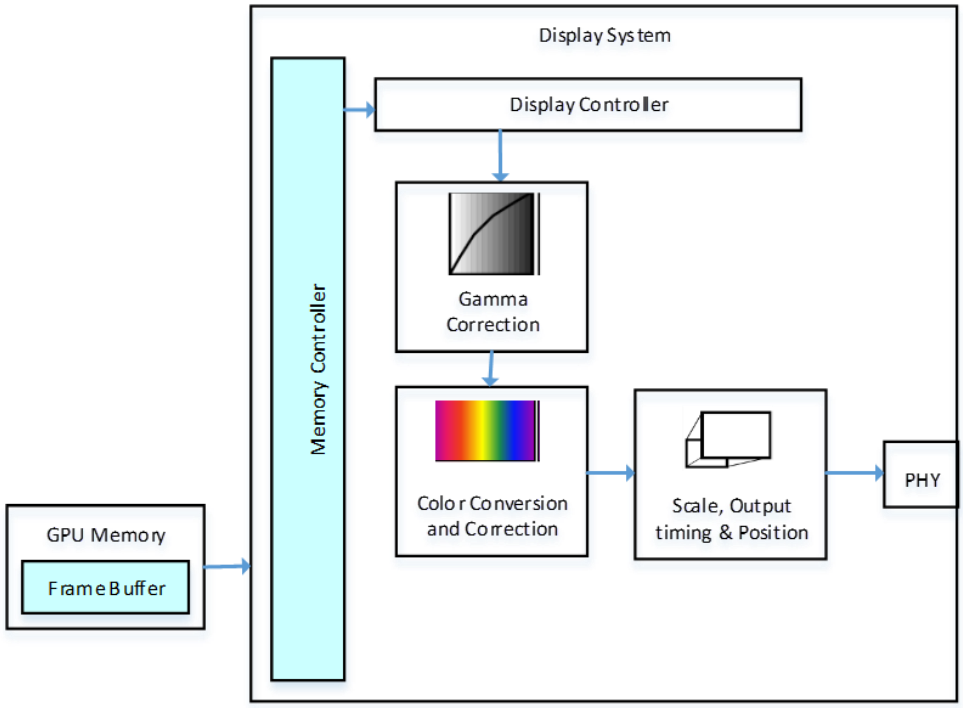

Figure 4: General Display System High Level Architecture Diagram

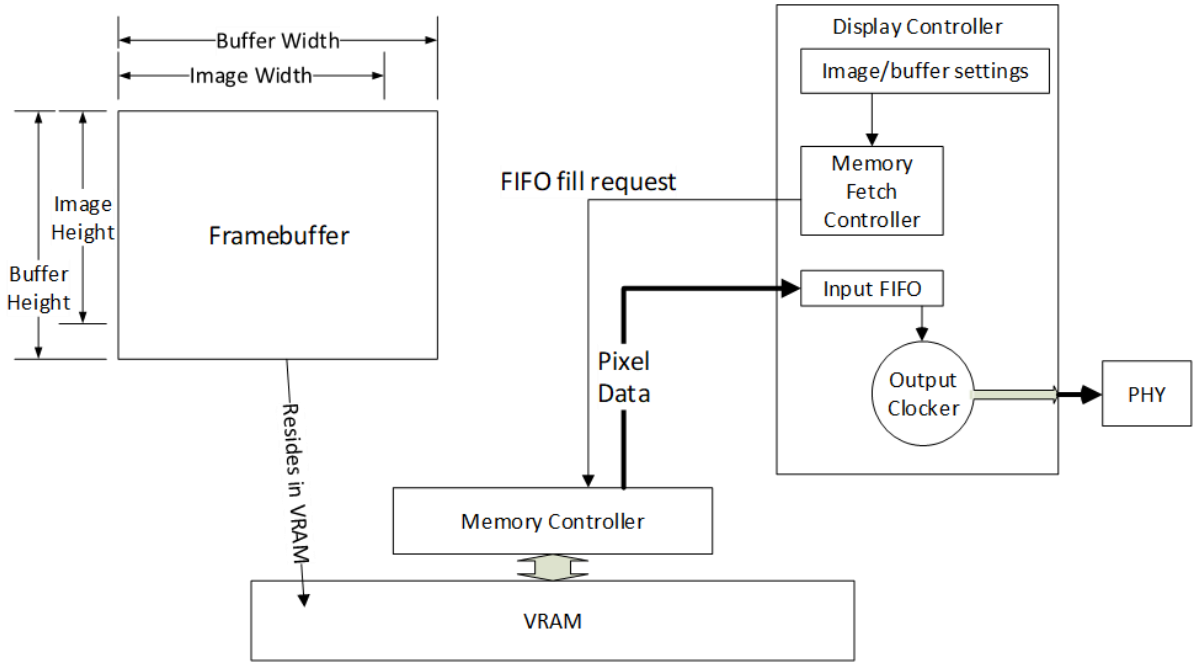

Figure 5: Display Controller Functionality

The display controller could fail to refill its FIFO due to an inability to read from GPU memory, an error issuing read request to GPU memory controller or an error in updating the FIFO pointers. Figure 6 shows the visual display results for this type of display controller error, which leads to the loss of the display.

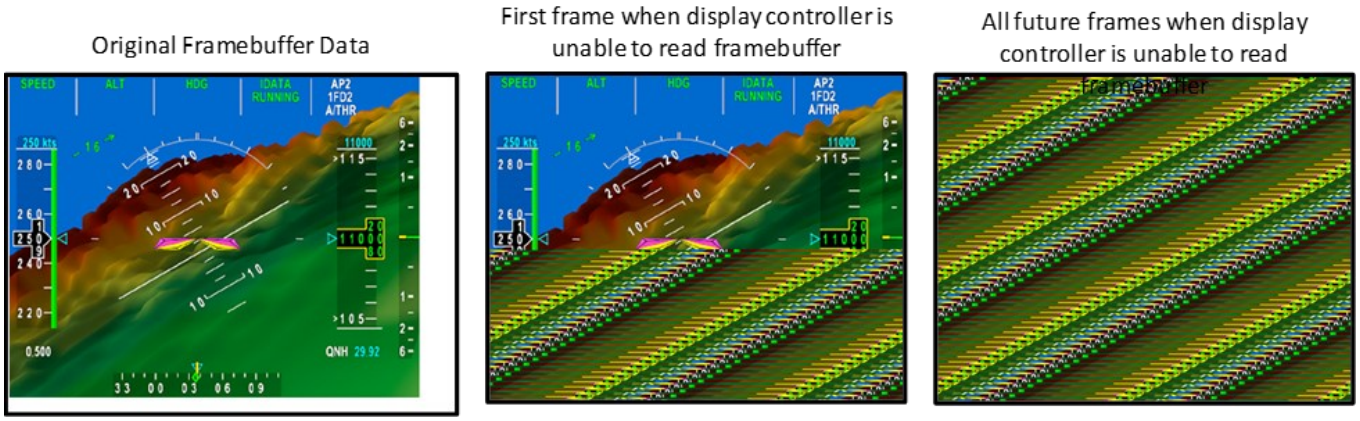

Figure 6: Display Controller Failure - Unable to Refill FIFO

The display controller could fail to skip unused pixels in the frame buffer due to a change in the buffer width setting (for example from a Single Event Upset which is caused by a single energetic particle causing non-destructive soft error) or fault in the display controller buffer width logic. Figure 7 illustrates the visual display results for this type of display controller error, which leads to the loss of the display.

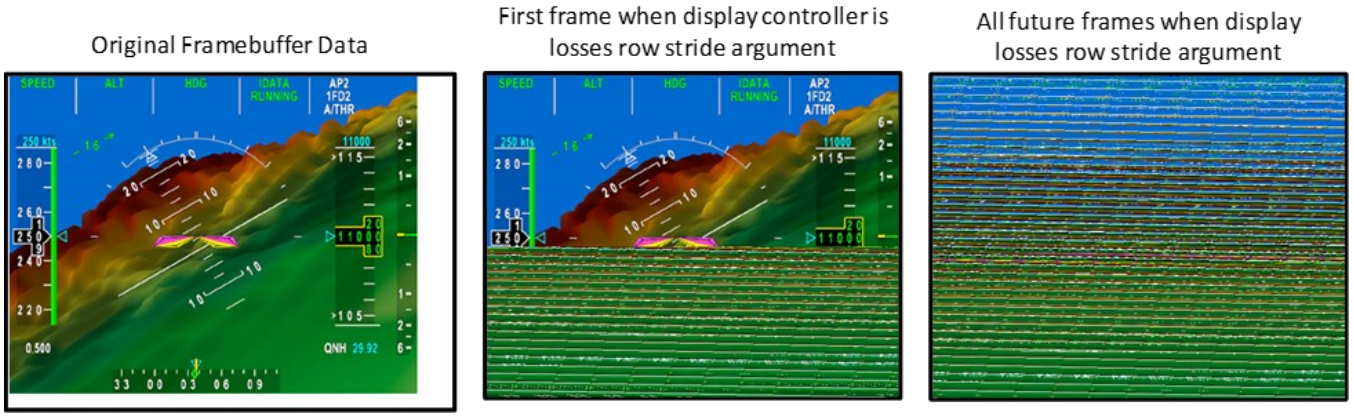

Figure 7: Display Controller Failure – Does Not Skip Unused Pixels in Buffer

Gamma correction could fail such that the resulting display is lighter or darker compared to what it should be, as shown in Figure 8. While this does not lead to HMI or loss of display, it may result in the display becoming harder to read which may result in delayed action in response to critical data on the display.

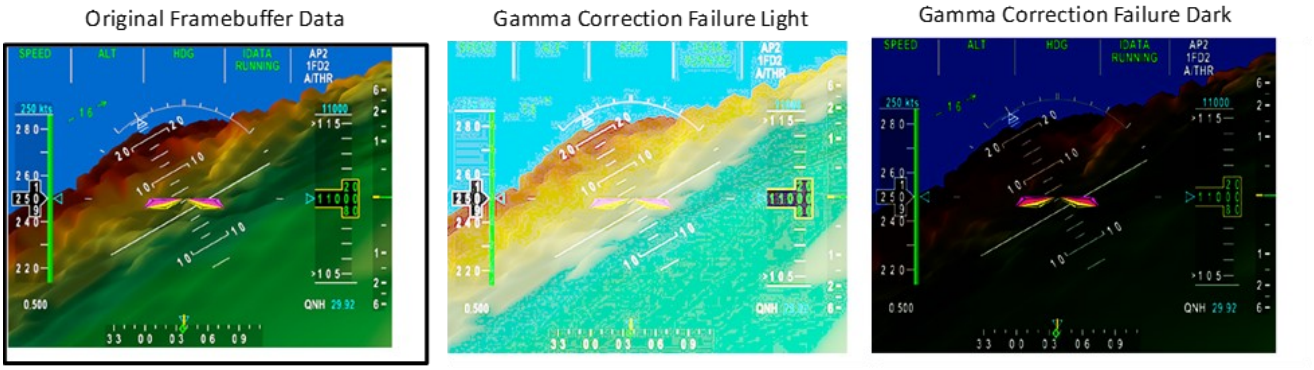

Figure 8: Gamma Correction Failures

Color conversion could fail to interpret the color space correctly, such as reading RGB555 data as a 32-bit color as shown in Figure 9. This leads to loss of display.

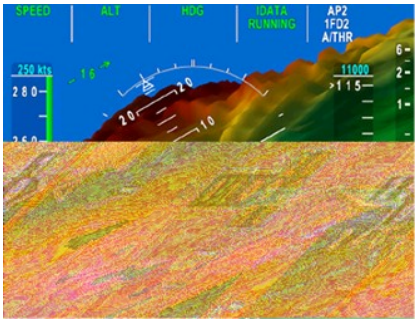

Figure 9: Color Conversion Failure Example

Position failures result when synchronization pulses are mis-timed with the buffer, as shown in Figure 10. This makes it extremely difficult to read the displayed data and leads to loss of display.

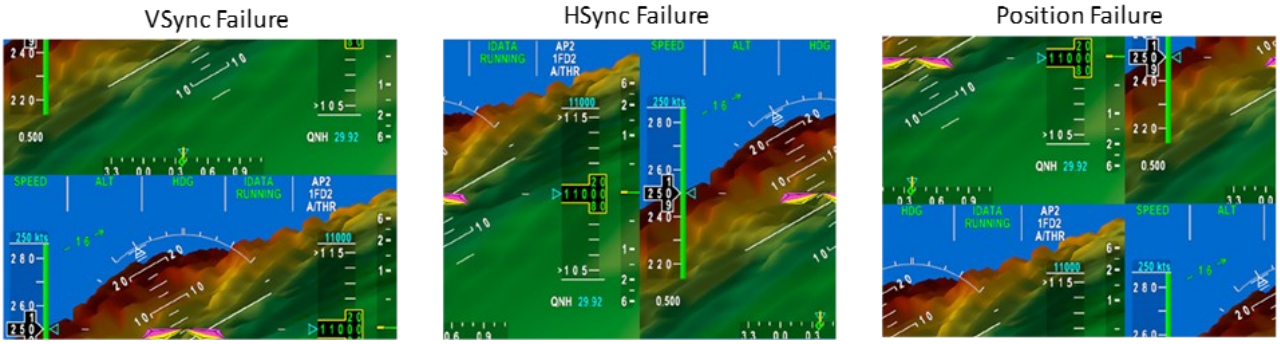

Figure 10: Position Failures

Figure 11 shows two examples of scaling failure. The first example shows what happens when trying to output a larger frame buffer on to a smaller display. For example, when trying to display an HD video source on an NTSC display, the failure of scaling simply crops the top left corner of the display (since the stride is still working, it just jumps lines). The second example shows what would happened if the stride also failed. This leads to loss of display.

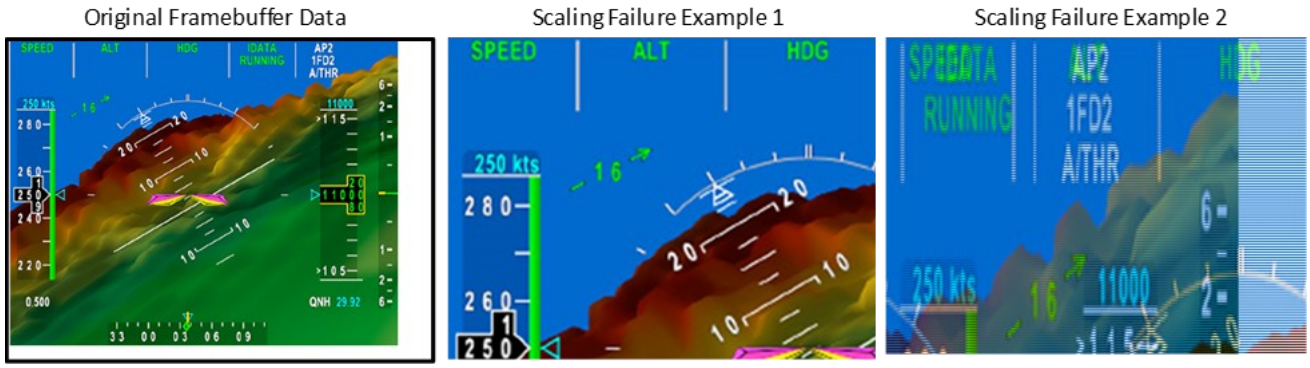

Figure 11: Scaling Failures

Figure 12 is an example of an output timing failure and shows what happens if the bytes are being fed to the output at twice the rate that the actual output clock is sending the bits to the display (in other words, skipping every other bit). This leads to loss of display.

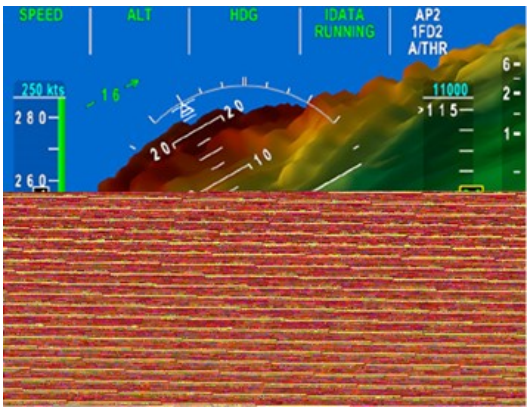

Figure 12: Output Timing Failure Example

Some display systems have a crossbar switch to select between different physical outputs (PHY) and a failure of this function could result in the display of information on the wrong display or a loss of display due to no signal driving it (i.e. remaining dark).

A traditional method for verifying the display system is to perform a Cyclic Redundancy Check (CRC) over the whole frame or sub-frame of the output with an FPGA, like Figure 3, and monitor to ensure the data is changing. This verifies that the display controller is updating the output (that is, no frozen data condition). If the display controller does not buffer an entire frame, monitoring this failure is not necessary as the failure of the display controller to update from the frame buffer would result in the failure a shown in Figure 6. This would be obvious to the person in the loop without crosschecking with a redundant display.

Similarly, should any of the display system functions, gamma correction, overlay, color conversion and correction, scale, output timing and position and output crossbar for display fail, the result on the display would also be obvious to the person in the loop without the need to crosscheck with a redundant display. As demonstrated, most display system failures result in the loss of display with some just making the display harder to read. However, it is still possible to use the display. That is, most failures of the display system do not result in the display of HMI.

An alternative to relying on a person in the loop is to utilize two independent CGPs rendering the same data and then perform a crosscheck verification. While it may have been possible to make this comparison in an FPGA, modern CGPs no longer provide a mechanism to synchronize their output timing. However, the high level of programmability offered in modern CGPs enables the capability to do the comparisons within the CGP domain using software, provided each CGP has access to its own frame buffer result as well as the frame buffer generated by the other CGP. Figure 13 illustrates an example block diagram for this configuration. A simple approach is to cross-DMA the frame buffers to each other over PCIe and compare at that level. While that would detect failures of the GPU, the display system would still rely on the person in the loop to determine which of the mismatching displays is correct. An improved approach is to capture each other's output, (illustrated by the additional path shown in Figure 13 in red), and compare, which would also include the display system into the detection path. Basically, both checks must pass to confirm the display is correct. The downside is that if the second (slave) CGP used for monitoring is the one that fails, a good display may be flagged as having HMI (false negative). This is more ideal in a complete digital graphics domain but if there is analog data involved the verification process becomes more complex to account for variances in analog to digital conversions. While the probability of two similar CGPs having the same failure at the same point in time should be low, there is typically insufficient reliability based on actual field experience to back up such a calculation. It may therefore be necessary to utilize two dissimilar CGPs. However, depending on the application, there may be difficulty in synchronizing the actions of the two CGPs due to performance differences. There may also be complexity added during verification of the data. While both frame buffers may be correct, they may be different due to, for example, differences in floating point accuracy between the two dissimilar CGPs.

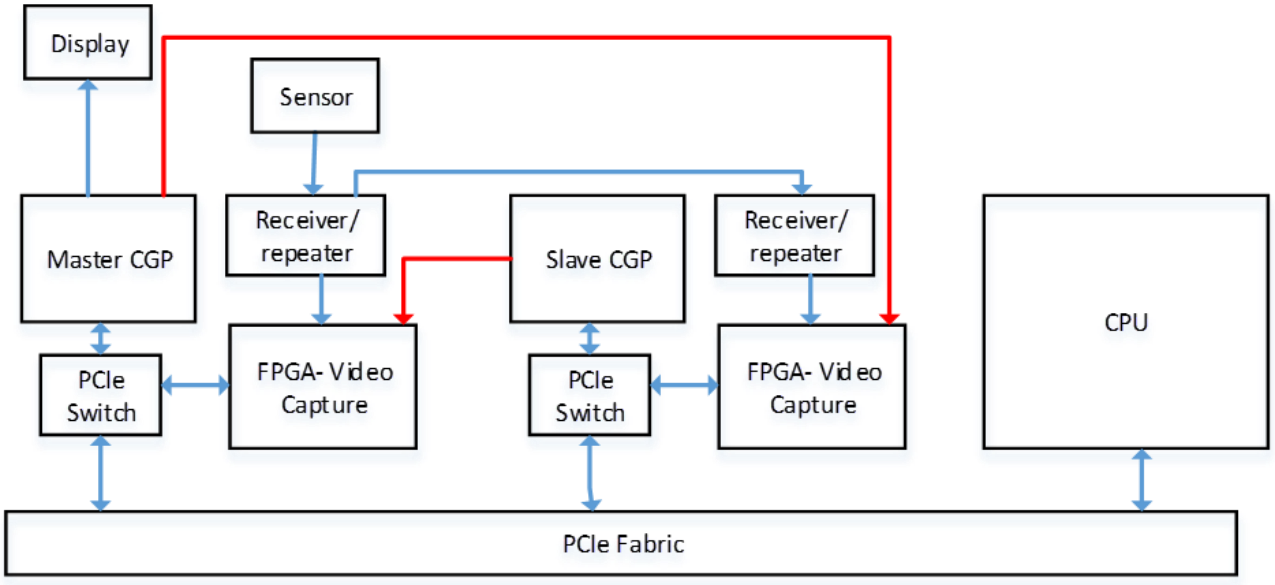

Figure 13: Redundant CGP with Cross-Monitoring

Design Errors Within the CGP

It is difficult to provide a high degree of confidence that the CGP does not contain any design errors. However, this is no different than other COTS devices being used in safety critical applications. In some cases, the CGP could be coming from the same manufacturer, such as AMD, Intel®, NXP® and Xilinx®. All of these are major COTS device manufacturers with a development process and internal quality standards that have long been used for devices delivered in large quantities. While any significant design error would typically be fed back to improve the process, this does not provide assurance against new design errors. The risk of a design error can be minimized through component selection. For example, selecting a part being used in high volume like NXP's i.MX 6 or a part that is based on technology that has already been proven in a high-volume situation like AMD's Embedded Radeon™ E8860 should result in known errata and behaviour.

One key to mitigation, as with all COTS devices, is to ensure the supply chain back to the CGP manufacturer has a demonstrable errata and Product Change Notice (PCN) communication process that feeds into the system change impact analysis process. For CGPs procured through CoreAVI, CoreAVI performs a change impact analysis of errata and PCNs on its products and provides that along with the errata and PCNs to its customers.

Design errors can not typically be detected by comparing outputs of two identical CGPs, as both would have the same error. However, a monitor like the TrueCore monitor discussed above is effective at detecting incorrect operation of the GPU due to a design error.

Failures or Inappropriate Responses to External Events

Most external events can be detected with safety monitors such as temperature, voltage and clock monitoring. While this is traditionally external to the CGP, some SoCs, such as the S32V234, that are developed to a functional safety process may have these monitors built in. Requirements for safety monitors are identified during the system safety assessment and flowed down to system and module requirements. This is no different to other COTS devices used in safety critical applications. The potential for field failures can be reduced through good design and component selection. For example, select CGPs with an extended temperature range from the manufacturer, such as NXP, or through extended temperature screening.

One area where CGPs differ from other COTS devices is with the nature of graphics processing it may be acceptable to use a CGP that is susceptible to radiation induced events. While a significant radiation event such as a nuclear event can be detected and result in a shut down to protect the electronics, there is continuous low-level radiation all around us with higher levels at higher altitudes and latitudes. This is normal in the earth's atmosphere and may cause Single Event Upsets (SEU) or Multiple Bit Upsets (MBU). Not all CGPs include parity error or ECC error detection and correction. While some CGPs do not have mitigation, let alone detection, for SEU and MBU events, the regenerative nature of each graphics frame in itself mitigates this (i.e. an SEU or MBU would be cleared out on the generation of the next frame buffer). However, there are some areas that are recommended to have BIT coverage:

a. Registers: verify register configuration settings on a regular basis to verify correct clock speeds, display controller settings and key functionality are unaffected. CoreAVI provides assistance in identifying the key registers to monitor.
b. Programs loaded in the programmable pipeline: CoreAVI's TrueCore software GPU safety monitor is an effective BIT to verify that the OpenGL SC 1.0.1 fixed function pipeline is un-affected.
c. Display lists, Virtual Buffer Object (VBO) and persistent or semi-persistent data: Verify on regular basis that these are un-affected.

With all these cases, the selected HMI mitigation approach(es) results from the system safety assessments to build the safety case for using a CGP in a safety critical system, which may even be at the most severe criticality level.

Item b - Multiple Display Failures Due to Common Failure Mode/Display System Availability

There is a concern that a common failure mode error associated with the use of common CGPs across all displays could have a significant impact on the availability of the entire display system. The worst case is where the entire display system is all associated displays on the platform. Depending on the safety criticality of the data being displayed, data may be shown on multiple displays such that if there was a failure in one path, a second display remains available as backup. That is, the loss of data on a single display would have less effect on safety than the simultaneous loss of the data on all displays.

This includes not only faults and design errors within the CGP, but also the hardware supporting the operation of the CGP. Events such as the loss of cooling air, extreme vibration or mechanical “shock”, etc., should not cause the loss of multiple displays. If the probability is more than what is commensurate with the hazard of the loss of multiple displays, then dissimilar CGPs is one solution which extends into independently designed redundant system elements to mitigate common failure mode errors to meet requirements for display system availability. CoreAVI supports multiple and dissimilar CGPs with a common API (OpenGL/EGL) which can be used with the same application software to make this solution easier.

Item c - CGP Device Variations During Production Life

There is a possibility that the CGP, during the production lifetime of the device, may exhibit variations or degradations in its performance or operating characteristics.

CGPs are manufactured on the same, or similar, production line as other COTS devices used on safety critical systems (there are only so many chip manufacturing facilities and lines in the world that have extensive production controls). To mitigate against changes in process, ensure that the supply chain back to the CGP manufacturer has an errata and Product Change Notice (PCN) notification process and that all issued PCNs are evaluated as part of a change impact analysis.

In addition, extended temperature CGPs are recommended as these devices have been through additional testing, either at the CGP manufacture or through extended temperature screening. Additional testing is used to provide CGPs with known operation over the specified environment range as well as screening to ensure the predetermined functionality and performance parameters are maintained over temperature.

Another method for mitigation is to implement a managed component supply program, either through CoreAVI or internally. Using a prediction of future requirements for the CGP, the required quantity of chips would be immediately procured. While this may not limit the total quantity procured to a single CGP production run, it should minimize the number of production lots involved. A sample from each CGP production lot could then be fully tested and qualified.

Item d - CGP Configurable Devices

Many CGPs contain configurable devices, such as separately loadable microcode or hardware “straps”. This capability leads to concerns regarding the configuration control of the CGP installed in the display system.

Some CGPs require Video BIOS (VBIOS) initialization which is a mix of ROM based straps (initialization data) and execution code that may be run on x86 instruction set processors. This can take two forms: one is a physical device and the second is to include a VBIOS image into the OpenGL driver library which CoreAVI supports. In both cases the VBIOS image is created and provided by the CGP manufacturer. The physical VBIOS device includes executable code that may be executed on x86 instruction set processors that utilize a BIOS bootstrap mechanism prior to starting the operating system. The topic of certifying with a BIOS/VBIOS is beyond the scope of this paper as this is complex software developed by third party specializing in BIOS software. There is one section of the VBIOS that applies to all processors configurations, the size of which may vary between CGPs (in the range of 40 32-bit words for the AMD Embedded Radeon E8860 for example), that contains register initialization data that the CGP automatically loads directly from the VBIOS (or loaded via OpenGL driver library when no VBIOS device is present). This portion of the VBIOS is treated as data for the purposes of safety certification and would be under configuration control at the hardware level for the physical devices or at the software level when included with the OpenGL driver library. These register settings are typically critical to the CGP operation and therefore BIT routines to monitor these registers continuously is recommended to verify proper initialization and the absence of data corruption.

Some CGPs require microcode be loaded for one or more engines. The microcode is developed by the CGP manufacturer and is not normally developed to safety certifiable development guidelines. To minimize concerns about the use of this microcode, CoreAVI includes it, as data, in its OpenGL driver library which is developed to safety certifiable development guidelines and extensively tested, including the OpenGL conformance suite on the target CGP hardware. This verifies the correct operation of the microcode to provide the complete functionality of OpenGL and applicable extensions. The microcode is therefore under configuration control through the configuration management of the OpenGL driver library.

There are also hardware strap options that may apply to some CGPs. These strapping configurations are determined during the design of the hardware configuration item utilizing the CGP based on system and module requirements. The hardware manufacturing process should include test coverage to verify the correct configuration. Additionally, the configuration should be understood to provide BIT monitoring coverage commensurate with the safety criticality of the item and configuration. Hardware strapping configuration is under configuration control at the hardware level.

Item e - Continued Monitoring of Supplier Data

There is a concern with the potential lack of awareness of any changes to the CGP that may affect the display system certification, and that could require existing analyses to be reassessed. These include but are not limited to the concerns expressed below:

a. Changes in fit, form or manufacturing techniques that may affect the physical layout, mechanical, electrical or thermal characteristics of the CGP
b. Changes or additions in functionality, including those aspects that are not used in the display system application, including firmware, device drivers and libraries
c. Performance enhancements, such as an increased operating frequency

CGP manufacturers have formal PCN and errata communication processes like other COTS devices used in safety critical applications. For example, NXP provides both processors like the Power Architecture® P5020 and P4080 and SoCs with integrated CGPs like the i.MX 6 and S32V234 all of which are finding use in safety critical applications.

The change control and change impact analysis that would already be in place will work equally well for CGP devices. Given the complexity of the CGP devices, CoreAVI as well as the CGP manufacturer's Field Application Engineering (FAE) team may be able to help in providing a clear understanding of the PCN or errata for an accurate change impact analysis as needed.

Item f - Unintended CGP Functionality

This concern is: does the CGP contain any functionality, used or un-used, documented or undocumented, in the application, which would cause HMI to be displayed or otherwise affect the integrity of the displayed data?

It should be noted that the European Aviation Safety Agency (EASA), the European certification authority for aviation, recognizes it is extremely problematic to show that a CGP, or any other very complex microprocessor device, does not contain any undocumented functionality. While 100% assurance of this point is not expected, evidence of best effort through extensive testing, including a large amount of robustness testing is expected to determine whether the CGP contains any undocumented features or functions that could affect the final design of the display system.

Since there is no practical way to uncover undocumented functionality, this may be approached by verifying the CGP continues to operate correctly during qualification and robustness testing. Confirming GPU operations with TrueCore to continuously monitor the GPU integrity, which demonstrates the GPU is performing the graphics pipeline operations correctly, is a method to show that if there is any undocumented functionality, it does not have any adverse effects. Also, given that all CGP interfacing from the application is through a driver library, this would show that no undocumented feature, or un-used feature, can have an impact on the correct rendering of the application data.

Item g - Open GL Software Drivers

There is a concern that because CGPs usually require complex software drivers that are resident in the main system processor, such software drivers might not have been developed to appropriate safety certifiable development processes with evidence and artifacts.

CoreAVI offers OpenGL SC 1.0.1, OpenGL SC 2.0, as well as DecodeCore™ video decode and EncodeCore™ video encode driver library suites (which support hardware acceleration engines included in many modern CGPs), along with the TrueCore software GPU safety monitor, all developed to a process compliant to ED-12C/DO-178C/ISO 26262/IEC-EN5128 to the most stringent assurance levels. These software items are developed from the ground up and do not include any third-party software IP. A certification evidence/artifact data kit is available to support system and higher level certification for these software modules for select CGPs.

CoreAVI also offers a configuration of the OpenGL driver library to include HyperCore™, a GPU virtualization manager to support hypervisor architectures. The GPU virtualization manager is also developed to the same process with certification evidence artifacts as the OpenGL driver library suites. These driver library suites support RTOSs used for safety critical applications as well as bare metal (no RTOS). With industry standard APIs, all leading human-machine interface tools are also supported to help minimize development time.

Item h - CGP Component Failure Rate

While CGP manufacturers typically perform reliability testing, actual reliability based on field data is not typically available. This is mainly due to the fact that most module (board) manufactures using CGPs do not go through the effort of returning failed devices, and if they do the manufacturer is not motivated to test to understand the failure due to effort (cost), particularly if only a few similar failures are observed. Also, when manufacturers do test returned parts, the usual fault identified is Electrical Over-Stress (EOS), which does not provide a path to clearly understanding the cause. EOS is typically associated with poor handling, but this is not always the cause.

The display system architecture should be such that the availability of the displayed critical data complies with the numerical probability required by the safety assessment process. If the display system fault trees use specific failure rates for CGPs, substantiating data or other appropriate justification for these failure rates is needed. An acceptable method of calculating an estimated failure rate or determining an appropriate empirical one should be agreed to with the certification authority beforehand.

Bonus 1 – Failure Modes and Effects Analysis (FMEA)

CoreAVI can provide Failure Modes and Effects Analysis (FMEA), which provides evidence of the effectiveness of CoreAVI's TrueCore™ in detecting situations that may lead to the display of Hazardously Misleading Information. The FMEA considers CoreAVI software (such as the ArgusCore SC™ OpenGL® driver library suite and TrueCore GPU safety monitor) for failure effects and detection for the selected hardware item.

Bonus 2 – CGP Service History Experience

CoreAVI can provide data on the number of CGPs sold along with guidance for the basis of a conservative estimate of hours of service history per CGP for select CGPs. It is unknown how many of the hours were in the aviation or safety industries, although military/aerospace can usually be identified and separated from commercial. The commercial hours would conservatively be counted as non-safety, even though some of those hours are likely in safety industries, such as medical or industrial. As device failures are not typically reported by GPU customers, actual failure rate in operation data is not available. CoreAVI supplies a safety manual, which contains the service history. This data can be used to supplement DO-254 lifecycle data.

Summary

CGPs are approved and in use on modern aircraft in applications with the highest safety requirements such as the Primary Flight Display (PFD). By following best practices, CGPs will continue to be used for safety critical display applications across many industries to provide accelerated and feature rich safety certifiable solutions.

Where to Find Additional Information Supporting the use of CGPs in a Safety Critical Application

CoreAVI provides high Technology Readiness Level (TRL) ArgusCore OpenGL SC 1.0.1 and ArgusCore OpenGL SC 2.0 drivers with TrueCore option with optional certification evidence for avionics, automotive and railway safety critical graphics functions.

More information about CoreAVI's ArgusCore OpenGL drivers, with extensions, along with a comparison of driver features can found here: CoreAVI safety certifiable graphics drivers.

More information about CoreAVI's TrueCore software GPU safety monitor can be found here: CoreAVI TrueCore

More information about CoreAVI's HyperCore GPU Virtualization Manager can be found here: CoreAVI HyperCore and described in a white paper: How to Implement Multiple GPU Applications in an Embedded System

More information on CGPs supported by CoreAVI can be found here: Graphics Processors

General information on CoreAVI certification data packages can be found here: Safety Certification

Contact CoreAVI to find out what we are working on and to discuss your demonstration/evaluation requirements: Sales@CoreAVI.com

Links to the EASA Certification Memo and CAST 29