Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

A Safety Critical Compositor for OpenGL SC 1.0.1 and OpenGL SC 2.0

Introduction

Modern embedded systems are capable of integrating more capabilities than ever before. With this advancement, there is a growing need to integrate third party applications that were previously held in separate systems into one cohesive system. For graphics-based applications, this means being to able to effectively and seamlessly share the display(s) amongst many applications.

A common solution used in graphical user interfaces is multi-windowing, where each application's content is rendered into its own window. A less common approach is to allow each application to open a non-overlapping window onto the display. While this method allows for faster drawing, it requires extensive integration and system testing efforts. For efficient multi-windowing to occur, a compositor is required.

This white paper details how a compositor works, the benefits and drawbacks of using different compositor solutions, and why using a compositor is conducive to safety certifiability to the most stringent levels for avionics, automotive, rail and other environments requiring safety critical operation.

Types of Windowing

In the past, Linux-based desktop solutions and some embedded systems have used X-server and its windowing functions to create displays. While X-server was the de facto windowing solution for many years, the main drawback to its use in modern systems is its large open source code base which is quite costly to implement in safety critical systems.

Stacking window managers were also commonly used. While this solution allowed windows to overlap, it also required repainting of each window one by one, often deleting content from the display, or causing the repetitious response in Figure 1 when the program became unresponsive. Neither of these drawbacks are conducive to safety critical requirements.

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

Figure 1: Unresponsiveness with a Stacking Windows Manager

Contemporary embedded systems using a Real Time Operating System (RTOS) are now frequently using EGL – an interface created by the Khronos Group to connect rendering APIs and the platform's windowing solution - for context management with their OpenGL ES 2.0, OpenGL SC 1.0.1 and OpenGL SC 2.0 driver libraries. While EGL also provides synchronization of rendering and buffer binding for these graphics APIs, it leaves compositing to the application, and assumes that all composited rendering is done in a single application memory space.

What is a Compositor?

Today's modern applications rely on efficient windowing compositors to create streamlined displays. The compositor is a software component that renders the windows from each application onto the visible display. In this solution, each application has its own off-screen window buffer that it draws data into, and the compositor creates a composition of windows on the display using what the applications have written to the buffers. The compositor allows multiple windows to exist on the display simultaneously, allowing them to overlap, cropping them to fit, and controlling if they are visible. Compositors are also capable of performing additional 2D or 3D effects such as blending, rotation, scaling and fading.

Types of Composition

There are two different types of graphical composition: Composition into the hardware level, and composition into a framebuffer.

Hardware Level Composition

Hardware Level Composition involves using the layers of the display controller at the ratio of compositing one buffer per layer. The capabilities of this type of composition rest solely on the layers' capabilities and it uses several forms of transparency - Destination View Port, Source Chroma, and Source Alpha Bending – for its layers. When this method is implemented in an FPGA external to the

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

GPU, the Alpha channel is lost, which often results in a single specific (RGB) color to be used to define transparency, leading to blending artifacts. While the benefits of Hardware Level Composition include good performance, power conservation, and efficiency in dealing with a large number of updates, this method can be quite costly, can only display one buffer per layer, and requires additional bandwidth to display all windows.

Composition into the Framebuffer

This type of composition combines multiple elements from multiple applications and off-screen buffers into a single framebuffer. The framebuffer then renders the data to the displays (Figure 2). Unlike Hardware Level Composition, this method is not impeded by a lack of available layers for the number of display needed, or by a layers' capabilities to support certain actions. Composition into the Framebuffer requires only one layer to display all buffers. Although it requires the power of the CPU or GPU to operate, this solution conserves bandwidth when updates are less frequent.

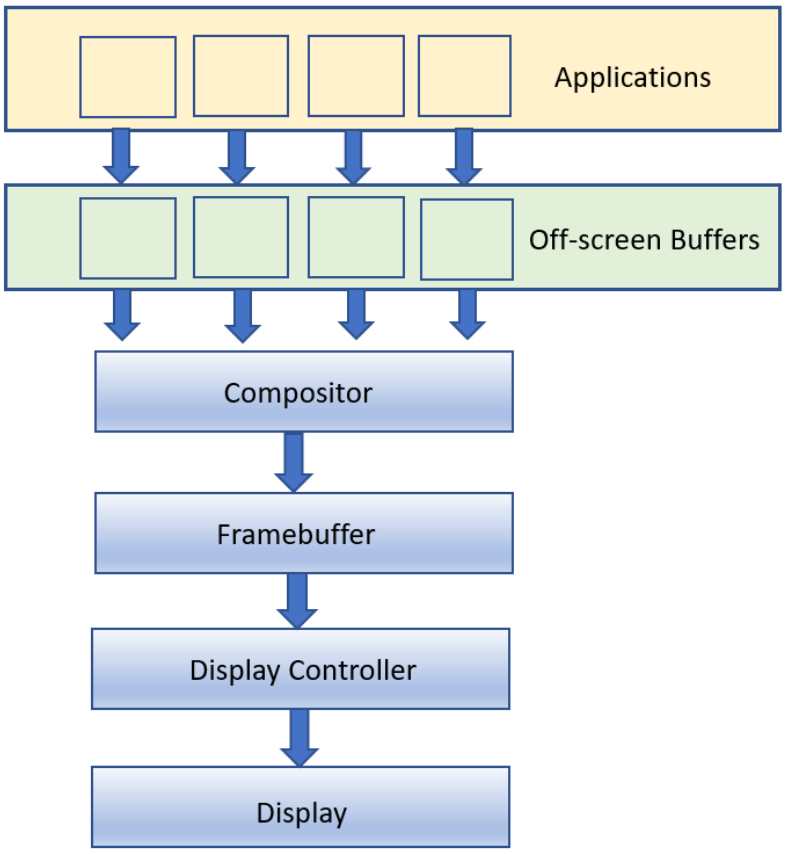

Figure 2: Composition into the Framebuffer: Path of Information from Applications to Display

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

EGL_EXT_Compositor and FACE Alignment

The FACE Consortium published the EGL_EXT_compositor extension with the Khronos Group on Feb 3, 2017 to add compositing capability to EGL. The extension minimizes application effort and allows for the composition of multiple windows within a multi-partition EGL system (Figure 3). The extension allows for the following capabilities[1]:

- Creation of a primary EGLContext and window for each display;
- Creation of additional windows using non-displayable surfaces;
- Composition of all non-displayable windows to a single display by way of a handle within the primary EGLContext to each off-screen window

There is one primary context created for the display while other EGLContexts are referred to as secondary contexts. The EGL_EXT_compositor is beneficial as it allows for off-screen asynchronous updates and also keeps information secure by preventing any non-primary contexts and surfaces from rendering to the display[2]. It allows (but does not specify the need for) management and control of GPU allocation to specific contexts including how much GPU memory a specific application is allowed to use. Each application draws into its own off-screen window, providing a level of security that ensures one application won't be overwritten with other applications' video data, or provide back-channel access though a shared framebuffer. The compositing application controls visibility of applications and prevents one application from drawing over another application unless system designer allows it. This extension ensures that OpenGL Graphics cannot be rendered to the display without express instructions from the compositing application, nor can they interfere with any other rendering contexts.

[1] Retrieved Aug 16, 2017 from https://www.khronos.org/registry/EGL/extensions/EXT/EGL_EXT_compositor.txt
[2] Retrieved Aug 16, 2017 from https://www.khronos.org/registry/EGL/extensions/EXT/EGL_EXT_compositor.txt

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

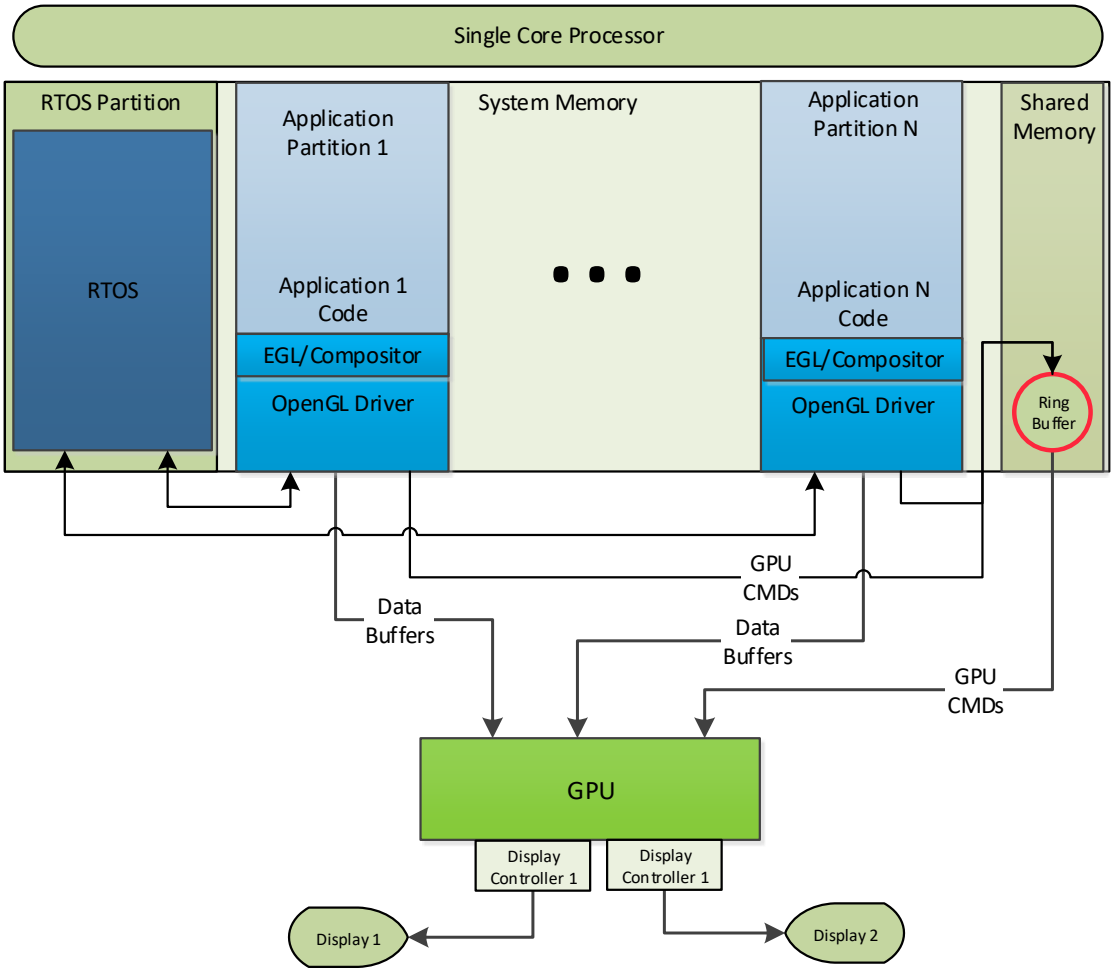

Figure 3: EGL_EXT_compositor Workflow

Another benefit of the EGL_EXT_compositor is that it provides standard windowing APIs for FACE alignment. For example, an EGL rendering context and window surfaces can be setup for use with the FACE Portable Components Segment, without the need to call platform specific APIs. This allows OpenGL applications using EGL to either own the GPU or be used with the FACE Graphics Services, enabling portability between platforms and the sharing of displays with other Graphics Services Units of Conformance. The EGL_EXT_compositor also provides a way to statically define the window management system and removes driver- and OS-specific calls for accessing the framebuffer. For FACE-alignment, applications cannot make specific callouts that aren't specified in the FACE Technical Standard, which for graphics means the OpenGL and EGL standards. The EGL_EXT_compositor uses EGL to setup windows statically, defining a windowing system which uses the EGL API only, with no 3rd party APIs (such as X-11, Windows, or Android).

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

Use Cases: Compositors and Safety Critical Applications

Safety critical applications require determinism and high reliability in graphics capabilities to meet Design Assurance Level (DAL) guidelines, and the use of a compositor can meet those requirements. In the realm of automotive safety criticality, Automotive Safety Integrity Level (ASIL) guidelines allow for there to be any combination of overlapping windows, so long as they do not violate safety restrictions such as overlapping on safety critical display region(s). For example, an automotive display may consist of several displays such as a speedometer, tachometer, etc. that need to be safety certified at ASIL C. A second ASIL D application that requires safety certifiability (airbag sensor, indicator lights, etc.) can also exist with the first application, but can be rendered into another area of memory. The compositor extension will ensure that between the 2 applications, the ASIL D application is always rendered and appears in a specific location on the screen (Figure 4). This not only allows display of the full stack on the screen, but also allows displays to be generated in smaller areas of the screen. In other words, a compositor application ensures that applications rendered to different locations in memory are managed properly on the physical display.

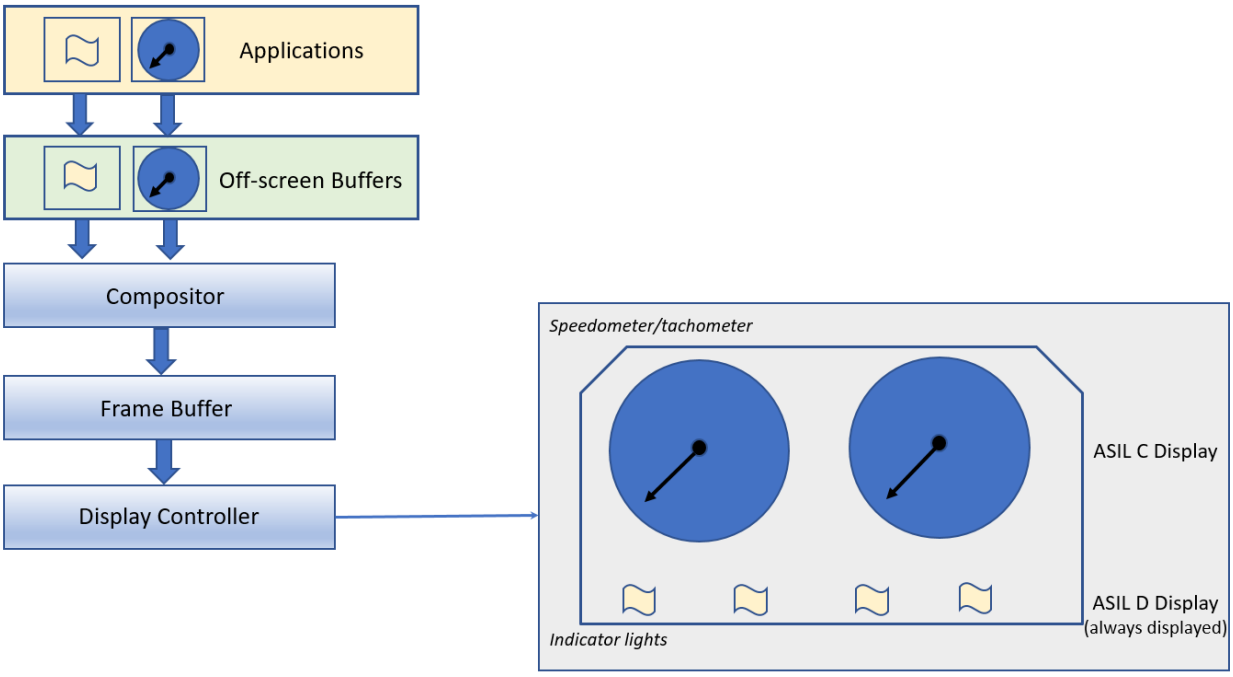

Figure 4: Automotive Mixed DAL Example

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

Another example of ensuring secured display is that of credit card digital pin keypads on retail purchasing devices. Digital pin keypads cannot be displayed unless it is assured that the display will not be covered by another keypad display. The windowing compositor can provide the guarantee that the digital keypad cannot be stacked with another keypad.

Figure 5 demonstrates single DAL level applications using the EGL_EXT_compositor with a compositing application. Large area displays (LAD) are become more popular in airborne applications, expanding from traditional 6 inch displays up to 22 or 24 inch displays. This requires the displays to securely and safely split the screen space to display a variety of applications. For example, the Primary Flight Instruments could appear on the right, while the left display holds the map, crew checklists, etc. The windowing compositor can ensure between one and six functions are successfully displayed together.

Figure 5: LAD Single DAL level applications using an EGL-EXT-compositor

In a hypervisor environment, an aircraft crew station with DAL D requirements may have both Linux OS environments and RTOS environments presenting applications into a windowing compositor. This configuration allows common Linux applications as well as flight specific applications – crew info, instrumentation, etc. - on to the same display. Instead of giving the display to one hypervisor guest, a

Core Avionics & Industrial Inc.
400 North Tampa Street
Suite 2850
Tampa, Florida 33602

T: 888-330-5376
F: 866-485-3199
www.coreavi.com

single display can be shared by multiple hypervisor guests with minimal interactions and communication between them.

Sharing a single display surface in a hypervisor system comes with many challenges. One major issue is how to pass data between different guest operating systems within the hypervisor. The EGL_EXT_compositor extension in a hypervisor environment allows the application to render its data to off-screen windows. Handles to the off-screen windows are passed to another application which does the compositing, and may be in a different guest OS or may be in the same guest OS. The benefit of this solution is that the rendering applications do not need extra logic added to communicate with the compositing application in a way that makes the software specific to a unique hypervisor instantiation.

Where to Find Additional Information on the EGL_EXT_compositor

CoreAVI is the only graphics driver supplier supporting standard windowing APIs for FACE-aligned window composition through the EGL_EXT_compositor.

CoreAVI provides high Technology Readiness Level (TRL) OpenGL SC 1.0.1 and OpenGL SC 2.0 drivers with a SecureCore option and HyperCore GPU virtualization manager 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.

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