EMBEDDED DISPLAY INTERFACES

White Paper

KOZIO®
Fastest path to reliable hardware

ABSTRACT

This is part one of a multi-part white paper series on effectively testing embedded display systems. Effectively testing embedded display systems requires a clear understanding of the various interfaces which may exist in the system. This paper details the hardware interconnections used across the spectrum of these systems, in preparation for subsequent papers which describe specific test methodologies for displays.

Covered in this paper are the basics of embedded display technologies and various display interface standards: VGA; Parallel (DPI); High Density Serial Links including Display Serial Interface (DSI), Flatlink (SDI), FPD-Link and Remote Frame Buffer Interface (RFB or DBI); Analog Television Interfaces such as Component Video, S-Video and Composite Video; and the Digital Television/Monitor Interfaces DVI and HDMI.

Contents

KOZIO® Fastest path to reliable hardware

INTRODUCTION

Embedded processors are beginning to include a wide variety of interfaces to display systems. These are driven by cell phone applications and mobile devices, but are also used in many other environments because a display provides a more powerful and flexible user interface than simple LEDs or character displays. This series of papers will review the various display interface standards, describe some of the challenges in testing a display and its interface to an embedded processor, and explore the capabilities of the Kozio kDiagnostics® platform to debug and test these interfaces.

DISPLAY BASICS

In general, the display subsystem of an embedded system is designed to transfer the data from a block of internal processor memory called a Frame Buffer (FB), which software running on the main Processor updates to change the image being displayed. The display data consists of some number of bits for each pixel of the displayed area. In most cases there are specific bits which describe the intensity of the red, green and blue (RGB) components of each pixel, although other formats such as YUV/YCrCb (luminance, red chrominance, blue chrominance) are occasionally used.

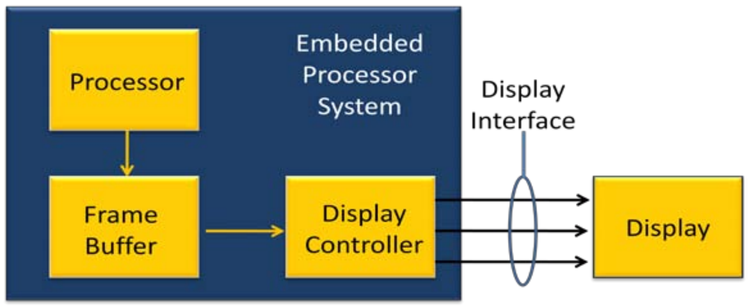

Figure 1 - Internal Display Controller Architecture

A logic block generally referred to as a Display Controller fetches data from the Frame Buffer, formats it according to the desired display interface, and transmits it to the Display. Figure 1 shows a basic system, where the Display Controller is internal to the Embedded Processor and communicates directly to the Display.

In many systems the Embedded Processor includes a Display Interface Controller, which creates an intermediate communication structure which is separate from the Display Interface. In this case the Display Controller is external to the Embedded processor, and is often included within the Display itself. This external Display Controller often includes its own Frame Buffer. Systems of this type are shown in Figure 2. Note that it is also possible to connect to an external Display Controller via a standard bus such as PCI.

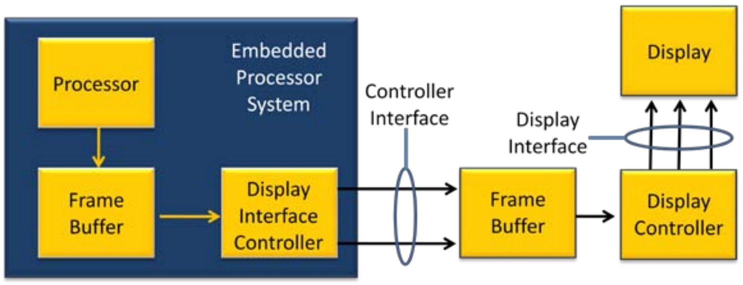

Figure 2 - External Display Controller Architecture

The intensity of each pixel on a display, whether it is a CRT, LCD or other type, generally must be periodically refreshed. The architecture of this function was developed in the days of CRTs, but has remained quite consistent as shown in Figure 3. The refresh is usually done by "raster scanning", which starts at the first pixel (typically the upper left hand corner), generates an intensity for that pixel, and then moves horizontally through all pixels of the first scan line. At that point a "horizontal synchronization" or HSYNC signal occurs, which causes the refresh to move to the beginning of the second line, and so on. This process continues until all lines have been refreshed, at which point a "vertical synchronization" or VSYNC signal occurs. This causes the refresh to return to the first pixel, and the process repeats.

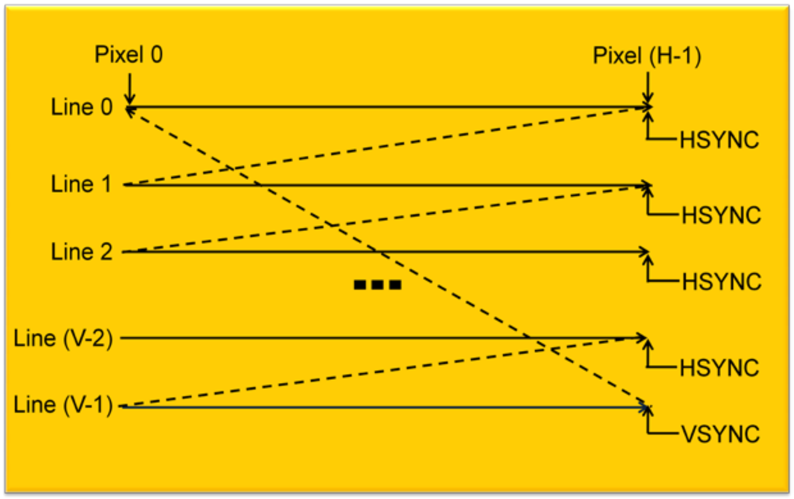

Figure 3 - Basic Display Timing

The physical nature of the display generally dictates that the response to HSYNC or VSYNC is not instantaneous, and thus nothing is refreshed for a time after each signal is received. These times are referred to as "blanking periods". They are shown as the dashed lines In Figure 3.

DISPLAY INTERFACE STANDARDS

A number of display interface standards have been developed over the years, evolving to adapt to requirements for interconnect speed, color depth, number of interconnections, etc.

VGA

The primary display interface standard for many years has been VGA, an analog standard which is optimized for connecting a CRT display over a cable to a computing system. The VGA interface, shown in Figure 4, consists of three analog intensities for red, green and blue, and two digital signals which carry HSYNC and VSYNC. Four additional digital signals carry either four display identification signals, or the clock and data signals of DDC which is a serial protocol identical to I^2C used to provide bidirectional communication to a display. This is a well-known, low cost system, but difficulties in maintaining image quality as the clock frequencies have increased (due to larger pixel resolutions and color depths) have led to the development of a number of digital interfaces.

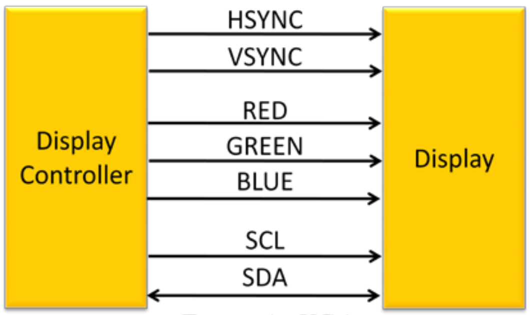

Figure 4 - VGA

Parallel (DPI)

The first digital interface to be developed was a parallel set of unidirectional digital interconnections, always including a pixel clock PCLK, horizontal and vertical synchronization signals HSYNC and VSYNC, and a set of data bits typically separated into red, green and blue signals. The basic connection is shown in Figure 5. In many applications a data enable signal DE is also included which indicates when displayable data is being transferred. This is a very simple interface to create, supports very high aggregate data rates, and is by far the most commonly used interface in printed circuit applications. This has been standardized by the Mobile Industry Processor Interface (MIPI) organization as the DPI (Display Pixel Interface) standard.

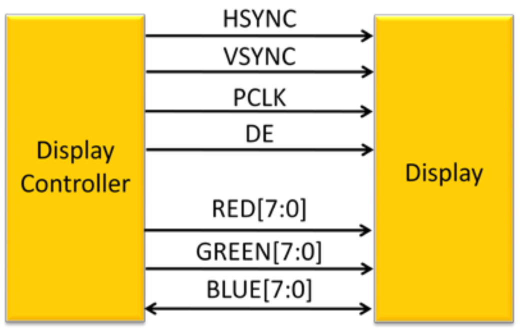

Figure 5 - Display Pixel Interface (DPI)

DPI is rarely used between physical boxes due to the high number of interconnections required (28 in the above example), and provides no mechanism for communication from the display to the processor. Because every pixel is transferred to the display for every frame, DPI also uses a significant amount of power and can generate challenging levels of Electromagnetic Interference (EMI). In many cases, the Display Interface is DPI but other protocols are used as the Controller Interface.

Almost all embedded processors which support display outputs provide a DPI interface, most often at 24 bits per pixel but occasionally at 16 or 18 bits per pixel. Since DPI is a unidirectional interface, most systems also include another standard interface, such as I^{2}C or SPI, to exchange configuration information with the Display.

High Density Serial Links

As form factors shrink, particularly for handheld devices, the large number of connections and high power dissipation in DPI make it unattractive in terms of package pin counts, PC board traces and interconnect cable signals. To overcome these limitations, several interface standards have been developed which use a small number of differential serial connections running at higher clock rates.

Displays themselves typically have DPI interfaces, so in most cases the high density interface is converted to DPI with an additional component, which may be integrated into the display assembly or be located on a system board as shown in Figure 2. This component also typically contains a local Frame Buffer to hold the displayed image, so that the full screen refresh rate does not need to be sustained over the serial interface.

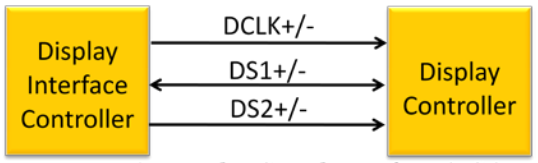

Figure 6 - Display Serial Interface (DSI)

Display Serial Interface (DSI)

DSI is another MIPI standard which is a Controller Interface using one high speed clock line and a few (typically 1 to 4) high speed data lines, as shown in Figure 6. One of the data lines is bidirectional and may be used to read data from the display device. Display data is generally sent in a burst which contains an entire frame, which is transmitted whenever the processor FB changes, or at a periodic rate which may be slower than the actual refresh rate of the display. Rectangular block transfers are also supported, and the DSI interface provides low speed read/write access to internal registers within the Display Controller.

DSI is architected to support multiple Display Controllers connected via the same physical DSI interconnect, but bandwidth limitations often make this impractical. The OMAP 36xx and 44xx processors from Texas Instruments include DSI interfaces. A number of Display Controller chips, such as the eDISCO series from Toshiba, support DSI inputs.

Flatlink (SDI)

Flatlink, also known as the Serial Digital Interface or SDI, was developed by Texas Instruments. It multiplies the pixel clock by 7 and serializes the parallel data and synchronization signals to 3 LVDS differential data signals as shown in Figure 7. Bidirectional transfers are not supported. The OMAP 34xx processors from Texas Instruments include a Flatlink interface.

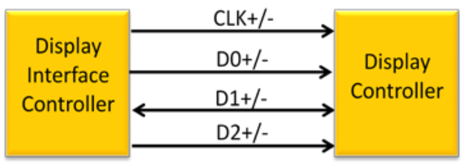

Figure 7 - Flatlink (SDI)

FPD-Link

FPD-Link was developed by National Semiconductor and is very similar to SDI. It serializes the DPI parallel clock, data and synchronization lines to 3 or 4 LVDS differential signals running at 7x the pixel clock frequency, which is physically the same as SDI as shown in Figure 7. FPD-Link is a modification of the older LVDS Display Interface (LDI), also from National. Bidirectional transfers are not supported.

Remote Frame Buffer Interface (RFB or DBI)

The previous approaches all generally serialize the parallel data stream, but maintain the synchronous nature of the data and require the same aggregate data bandwidth as the parallel interface. The Remote Frame Buffer Protocol (RFB), which has been standardized by MIPI as the Display Bus Interface (DBI), explicitly takes advantage of the existence of a FB in the display controller and transfers commands to update the FB image only when necessary. DBI, as shown in Figure 8, is somewhere between a purely parallel DPI interface and a high density DSI interface, with a more bus-oriented approach typically including read and write enable signals, chip selects, an address/data signal, and several data lines (usually 8 to 16) used for bidirectional transfers of address and data. The OMAP35xx processor family from TI includes an RFB interface.

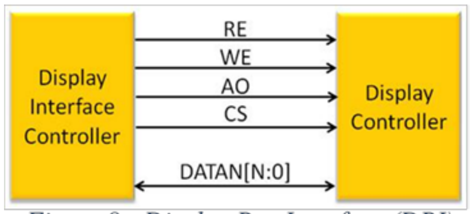

Figure 8 - Display Bus Interface (DBI)

Analog Television Interfaces

Many embedded processors include integrated interfaces for connecting directly to analog televisions. There are three main interface formats.

Component Video

Component video provides three separate analog signals – either red, green and blue or luminance (Y), red chrominance (Cr) and blue chrominance (Cb), on three RCA connectors. This interface is shown in Figure 9.

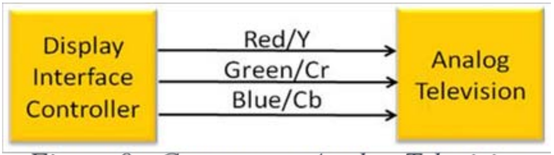

Figure 9 - Component Analog Television

S-Video

S-Video (for Separate Video) video utilizes two analog signals, a luminance signal as in Component Video and a single chrominance signal, on a single 4-pin connector. This interface is shown in Figure 10. The OMAP 34xx and 35xx embedded processors from TI support the S-Video interface.

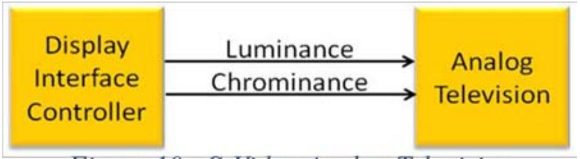

Figure 10 - S-Video Analog Television

Composite Video

Composite video combines the luminance and chrominance signals of S-Video into a single analog signal, on a single RCA connector, as shown in Figure 11. The OMAP 34xx, 35xx, 36xx and 44xx processors from TI include a Composite Video connection.

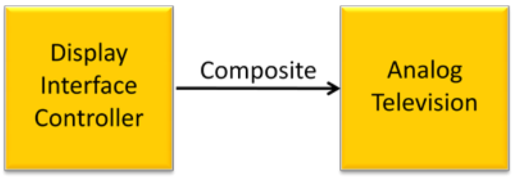

Figure 11 - Composite Analog Television

Digital Television/Monitor Interfaces

As television and monitor resolutions increased, the need emerged for a higher quality digital alternative which could be carried over cables a few meters in length. Two main standards have been developed, primarily by Silicon Image, specifically for television applications: the Digital Video Interface (DVI) and a later enhancement called the High Definition Multimedia Interface (HDMI). Interface parts typically translate DPI to DVI and/or HDMI, and do not contain a Frame Buffer.

DVI

DVI includes a differential pixel clock, 3 high speed (10x pixel clock) differential data lines, a bidirectional DDC interface similar to I^{2}C and some status signals - see Figure 12. DVI transfers the full video at serial rates in excess of 2 Gb/s and uses a relative large cable connector.

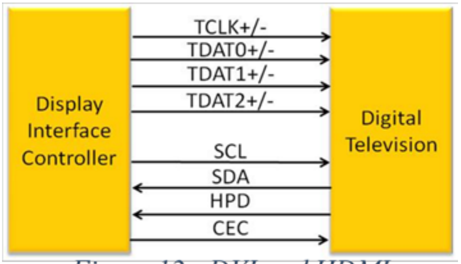

Figure 12 - DVI and HDMI

HDMI

HDMI has the same physical and video interfaces as DVI (see Figure 11), but utilizes the video blanking periods to transmit audio and HDCP security information. It also uses a much smaller connector than DVI and is the digital interface utilized in all digital televisions today. HDMI controllers can drive DVI Displays using a cable adapter, but the audio capability is not available. The OMAP44x embedded processors from Texas Instruments provide an integrated HDMI interface.

TESTING AN EMBEDDED DISPLAY SUBSYSTEM

An embedded display subsystem has three to five components: the display controller within the embedded processor, the display, the display interface, and in many cases an external controller component and its connection to the processor.

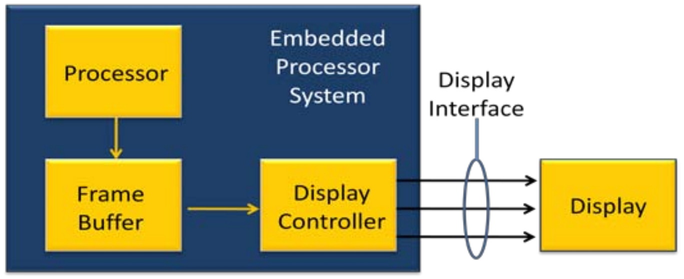

Figure 1 - Internal Display Controller Architecture

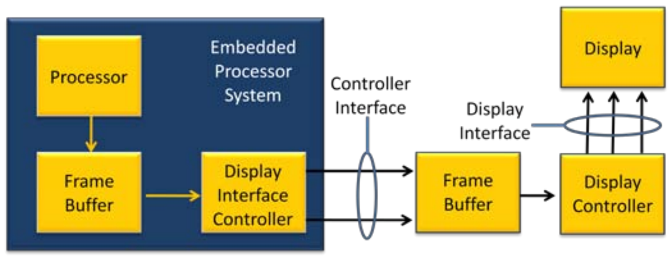

Figure 2 - External Display Controller Architecture

Figures 1 and 2 show the various configurations. The process for testing such a subsystem depends heavily on the display interface, and must consider each of the components. Since there is no easy automated way to determine what is being shown on a display, in most cases the test process involves displaying a set of particular images and having an operator verify that the image is correctly displayed.

Testing all of the various components requires several specific test functions. The subsequent papers in this series will show how the Kozio kDiagnostics environment greatly simplifies the process of generating tests for these elements and debugging failures detected by those tests.

CONTACT INFORMATION

Kozio, Inc.
www.kozio.com
+1.303.776.1356
Email Inquiries: *sales@kozio.com*