VIDEO INTERFACE ARRANGED TO PROVIDE PIXEL DATA INDEPENDENT OF A LINK CHARACTER CLOCK

FIELD OF THE INVENTION

[0001] The invention relates to display devices. More specifically, the invention relates to digital display interface suitable for coupling video sources to video display devices.

BACKGROUND OF THE INVENTION

[0002] Currently, video display technology is divided into analog type display devices (such as cathode ray tubes) and digital type display devices (such as liquid crystal display, or LCD, plasma screens, etc.), each of which must be driven by specific input signals in order to successfully display an image. For example, a typical analog system includes an analog source (such as a personal computer, DVD player, etc.) coupled directly to a display device (sometimes referred to as a video sink) by way of a communication link. The communication link typically takes the form of a cable (such as an analog VGA cable in the case of a PC, otherwise referred to as VGA DB15 cable) well known to those of skill in the art. For example, the VGA DB15 cable includes 15 pins, each of which is arranged to carry a specific signal.

[0003] One of the advantages of the VGA DB15 cable is the ubiquitous nature of the cable, due to the large and ever-expanding installed base. As long as the analog systems described above predominate, there is little incentive to migrate away from any other cable form than the VGA DB15.

[0004] However, in recent years, the exploding growth of digital systems has made the use of digital capable cables such as Digital Visual Interface (DVI) cable more desirable. It is well known that DVI is a digital interface standard created by the Digital Display Working Group (DDWG). Data are transmitted using the transition minimized differential signaling (TMDS) protocol, providing a digital signal from the PC's graphics subsystem to the display. DVI handles bandwidths in excess of 160 MHz and thus supports UXGA and HDTV with a single set of links.

[0005] Today's display interconnect landscape includes the VGA (analog) and DVI (digital) for desktop display interconnect applications as well as LVDS (digital) for internal connectivity applications within laptops and other all-in-one devices. Graphics IC vendors, display controller IC vendors, monitor manufacturers and PC OEMs as well as desktop PC consumers, to one degree or another, must factor interface choice into their design, product definition, manufacturing, marketing and purchase decisions. For example, if a consumer purchases a PC with an analog VGA interface then the consumer must either purchase an analog monitor or a digital monitor in which the analog video signal provided by the VGA interface has been digitized by way of an inline analog to digital converter (ADC) or an ADC built into the particular monitor.

[0006] Therefore, it would be desirable to have a digital interface that is more cost effective that current interfaces (such as DVI) for coupling video sources and video displays. In some cases, the digital interface would also be backward compatible with analog video, such as VGA.

SUMMARY OF THE INVENTION

[0007] A packet based display interface arranged to couple a multimedia source device to a multimedia sink device is disclosed that includes a transmitter unit coupled to the source device arranged to receive a source packet data stream in accordance with a native stream rate, a receiver unit coupled to the sink device, and a linking unit coupling the transmitter unit and the receiver unit arranged to transfer the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other.

[0008] In another embodiment, a packet based method of coupling a multimedia source device to a multimedia sink device is disclosed. The method includes the operations of receiving source video data in accordance with a native video data rate and transferring the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other.

[0009] In another embodiment, computer program product for coupling a multimedia source device to a multimedia sink device is disclosed. The computer program product includes computer code for receiving source video data in accordance with a native video data rate, computer code for transferring the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other, and computer readable medium for storing the computer code.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a generalized representation of a cross platform display interface 100 in accordance with an embodiment of the invention.

Figs. 2A – 2C illustrates a video interface system that is used to connect a video source and a video display unit in accordance with a number of embodiments of the invention.

Fig. 3 shows exemplary main link rates in accordance with an embodiment of the invention.

Fig. 4A shows a main link data packet in accordance with an embodiment of the invention.

Fig. 4B shows a main link packet header in accordance with an embodiment of the invention.

Fig. 5A shows a system arranged to provide sub-packet enclosure and multiple-packet multiplexing in accordance with an embodiment of the invention.

Fig. 5B shows another implementation of the system shown in Fig. 5A.

Fig. 6 shows a high-level diagram of the multiplexed main link stream as an example of the stream shown in Fig. 5.

Fig. 7 show another example of a data stream in accordance with the invention.

Fig. 8 shows yet another example of a multiplexed data stream in accordance with an embodiment of the invention.

Fig. 9A shows a representative sub-packet in accordance with an embodiment of the invention.

Fig. 9B shows a representative main link data packet in accordance with an embodiment of the invention.

Fig. 10 shows an example of a selectively refreshed graphics image.

Fig. 11 shows an exemplary link training pattern in accordance with an embodiment of the invention.

Fig. 12 illustrates a logical layering of the system in accordance with an embodiment of the invention.

Fig. 13 shows an exemplary special character mapping using 8B/10B in accordance with an embodiment of the invention.

Fig. 14 shows an exemplary Manchester II encoding scheme in accordance with an embodiment of the invention.

Fig. 15 shows a representative auxiliary channel electrical sub layer in accordance with an embodiment of the invention.

Fig. 16 shows a representative main link electrical sub layer in accordance with an embodiment of the invention.

Fig. 17 shows a representative connector in accordance with an embodiment of the invention.

Fig. 18 shows a source state diagram in accordance with an embodiment of the invention.

Fig. 19 shows a display state diagram in accordance with an embodiment of the invention.

Figs. 20 – 24 illustrate various computer based implementations of the invention.

Fig. 25 shows a flowchart detailing a process for determining an operational mode of the interface in accordance with an embodiment of the invention.

Fig. 26 shows a flowchart detailing a process for providing a real time video image quality check in accordance with some aspects of the invention.

Fig. 27 shows a flowchart for a link set up process in accordance with an embodiment of the invention.

Fig. 28 shows a flowchart detailing a process for performing a training session in accordance with an embodiment of the invention.

Fig. 29 illustrates a computer system employed to implement the invention.

DETAILED DESCRIPTION OF SELECTED EMBODIMENTS

[0010] Reference will now be made in detail to a particular embodiment of the invention an example of which is illustrated in the accompanying drawings. While the invention will be described in conjunction with the particular embodiment, it will be understood that it is not intended to limit the invention to the described embodiment. To the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

[0011] The inventive interface is a point-to-point, packet-based, plug & play, serial digital display interface that is both open and scalable that is suitable for use with, but not limited to, desktop monitors as well as providing LCD connectivity within notebook/all-in-one PC's, and consumer electronics display devices including HDTV displays and the like. Unlike conventional display interfaces that transmit a single video raster plus timing signals such as Vsync, Hsync, DE, etc., the inventive interface provides a system of multi-stream packet transfer capable of transferring one or more packet streams simultaneously in the form of "virtual pipes" established within a physical link.

[0012] For example, Fig. 1 shows a generalized representation of a cross platform packet based digital video display interface 100 in accordance with an embodiment of the invention. The interface 100 connects a transmitter 102 to a receiver 104 by way of a physical link 106 (also referred to as a pipe). In the described embodiment, a number of data streams 108 – 112 are received at the transmitter 102 that, if necessary, packetizes each into a corresponding number of data packets 114. These

data packets are then formed into corresponding data streams each of which are passed by way of an associated virtual pipe 116 – 120 to the receiver 104. It should be noted that the link rate (i.e., the data packet transfer rate) for each virtual link can be optimized for the particular data stream resulting in the physical link 106 carrying data streams each having an associated link rate (each of which could be different from each other depending upon the particular data stream). The data streams 110 – 114 can take any number of forms such as video, graphic, audio, etc.

[0013] Typically, when the source is a video source, the data streams 110 – 114 include various video signals that can have any number and type of well-known formats, such as composite video, serial digital, parallel digital, RGB, or consumer digital video. The video signal can be an analog video signal provided the source 102 includes some form of an analog video source such as for example, an analog television, still camera, analog VCR, DVD player, camcorder, laser disk player, TV tuner, set top box (with satellite DSS or cable signal) and the like. The source 102 can also include a digital image source such as for example a digital television (DTV), digital still camera, and the like. The digital video signal can be any number and type of well known digital formats such as, SMPTE 274M-1995 (1920 x 1080 resolution, progressive or interlaced scan), SMPTE 296M-1997 (1280 x 720 resolution, progressive scan), as well as standard 480 progressive scan video.

[0014] In the case where the source 102 provides an analog image signal, an analog-to-digital converter (A/D) converts an analog voltage or current signal into a discrete series of digitally encoded numbers (signal) forming in the process an appropriate digital image data word suitable for digital processing. Any of a wide variety of A/D converters can be used. By way of example, other A/D converters include, for

example those manufactured by: Philips, Texas Instrument, Analog Devices, Brooktree, and others.

[0015] For example, if the data stream 110 is an analog type signal, the an analog to digital converter (not shown) included in or coupled to the transmitter 102 will digitize the analog data which is then packetize by a packetizer that converts the digitized data stream 110 into a number of data packets 114 each of which will be transmitted to the receiver 104 by way of the virtual link 116. The receiver 104 will then reconstitute the data stream 110 by appropriately recombining the data packets 114 into their original format. It should be noted that the link rate is independent of the native stream rates. The only requirement is that the link bandwidth of the physical link 106 be higher than the aggregate bandwidth of data stream(s) to be transmitted . In the described embodiment, the incoming data (such as pixel data in the case of video data) is packed over the respective virtual link based upon a data mapping definition. In this way, the physical link 106 (or any of the constituent virtual links) does not, as does conventional interconnects such as DVI, carry one pixel data per link character clock.

[0016] In this way, the interface 100 provides a scaleable medium for the transport of not only video and graphics data, but also audio and other application data as may be required. In addition, the invention supports hot-plug event detection and automatically sets the physical link (or pipe) to its optimum transmission rate. The invention provides for a low pin count, purely digital display interconnect for all displays suitable for multiple platforms. Such platforms include host to display, laptop/all-in-one as well as HDTV and other consumer electronics applications.

[0017] In addition to providing video and graphics data, display timing information can be embedded in the digital stream providing essentially perfect and instant

display alignment, obviating the need for features like "Auto-Adjust" and the like. The packet based nature of the inventive interface provides scalability to support multiple, digital data streams such as multiple video/graphics streams and audio streams for multimedia applications. In addition, a universal serial bus (USB) transport for peripheral attachment and display control can be provided without the need for additional cabling.

[0018] Other embodiments of the inventive display interface will be discussed below.

[0019] Fig. 2 illustrates a system 200 based upon the system 100 shown in Fig.1 that is used to connect a video source 202 and a video display unit 204. In the illustrated embodiment, the video source 202 can include either or both a digital image (or digital video source) 206 and an analog image (or analog video source) 208. In the case of the digital image source 206, a digital data stream 210 is provided to the transmitter 102 whereas in the case of the analog video source 208, an A/D converter unit 212 coupled thereto, converts an analog data stream 213 to a corresponding digital data stream 214. The digital data stream 214 is then processed in much the same manner as the digital data stream 210 by the transmitter 102. The display unit 204 can be an analog type display or a digital type display or in some cases can process either analog or digital signals provided thereto. In any case, the display unit 204 includes a display interface 216 that interfaces the receiver 104 with a display 218 and a D/A converter unit 220 in the case of an analog type display. In the described embodiment, the video source 202 can take any number of forms (such as a personal desktop computer, digital or analog TV, set top box, etc.) whereas the video display unit 104 can take the form of a video display (such as an LCD type display, CRT type display, etc.).

[0020] Regardless of the type of video source or video sink, however, the various data streams are digitized (if necessary) and packetized prior to transmission over the physical link 106 which includes a uni-directional main link 222 for isochronous data streams and a bi-directional auxiliary channel 224 for link setup and other data traffic (such as various link management information, Universal serial bus (USB) data, etc.) between the video source 202 and the video display 204.

[0021] The main link 222 is thereby capable of simultaneously transmitting multiple isochronous data streams (such as multiple video/graphics streams and multi-channel audio streams). In the described embodiment, the main link 222 includes a number of different virtual channels, each capable of transferring isochronous data streams (such as uncompressed graphics/video and audio data) at multiple gigabits per second (Gbps). From a logical viewpoint, therefore, the main link 222 appears as a single physical pipe and within this single physical pipe, multiple virtual pipes can be established. In this way, logical data streams are not assigned to physical channels rather, each logical data stream is carried in its own logical pipe (i.e., virtual channel described above).

[0022] In the described embodiment, the speed, or transfer rate, of the main link 222 is adjustable to compensate for link conditions. For example, in one implementation, the speed of the main link 222 can be adjusted in a range approximated by a slowest speed of about 1.0 Gbps to about 2.5 Gbps per channel in approximately 0.4Gbps increments (see Fig. 3). At 2.5 Gbps per channel, the main link 222 can support SXGA 60Hz with a color depth of 18 bits per pixel over a single channel. It should be noted that a reduction in the number of channels reduces not only the cost of interconnect, but also reduces the power consumption which is an important consideration (and desirable) for power sensitive applications such as portable

devices and the like. However, by increasing the number of channels to four, the main link 222 can support WQSXGA (3200 x 2048 image resolution) with a color depth of 24-bits per pixel at 60Hz. or QSXGA (2560 x 2048) with a color depth of 18-bits per pixel at 60Hz, without data compression. Even at the lowest rate of 1.0 Gbps per channel, only two channels are required to support an uncompressed HDTV (i.e., 1080i or 720p) data stream.

[0023] In the described embodiment, a main link data rate is chosen whose bandwidth exceeds the aggregate bandwidth of the constituent virtual links. Data sent to the interface arrives at the transmitter at its native rate. A time-base recovery (TBR) unit 226 within the receiver 104 regenerates the stream's original native rate using time stamps embedded in the main link data packets, if necessary. It should be noted, however, that for appropriately configured digital display devices 232 shown in Fig. 2B, time base recovery is unnecessary since display data is be sent to the display driver electronics at the link character clock rate, thereby greatly reducing the number of channels required with a commensurate reduction in complexity and cost for the display. For example, Fig. 2C illustrates an exemplary LCD panel 232 configured in such a way that no time base recovery since display data is essentially pipelined to the various column drivers 234 that are used in combination with row drivers 236 to drive selected display elements 238 in the array 240.

[0024] Other embodiments describe a simple enumeration method for the link rate and the pixel/audio clock rate. It has been researched and understood that all the standard pixel/audio clock frequencies that exist today are a subset of the following master frequency:

$23.76\text{GHz} = 210 \times 33 \times 57 \times 111\ \text{Hz}$

This means that a pixel (or audio) clock rate can be expressed with four parameters, A, B, C, and D as:

$$\text{Pixel clock rate} = 2A * 3B \times 5C \times 11D$$

A = 4 bits, B = 2 bits, C = 3 bits, and D = 1 bit.

[0025] Even for a link whose link rate (which is the serial link bit rate / 10 for a link that uses 10-bit character such as 8B/10B characters) may be different from the pixel clock rate, there is a benefit in defining the link rate with these four parameters, A', B', C', and D': The benefit is the simplicity in regenerating pixel/audio clocks from a link clock. For example, let's say the link rate is set as A' = 6, B' = 3, C' = 7, and D' = 0 and the corresponding link rate is 135MHz. However, suppose the pixel clock rate is set as A = 8, B = 3, C = 6, and D = 0 (= 108MHz), then the pixel clock can be generated from link clock as pixel clock rate is equal to the link rate * 22 / 5 1 .

[0026] Referring back to those systems requiring time base recovery, the time-base recovery unit 226 may be implemented as a digital clock synthesizer. For an uncompressed video stream, the time stamp is stored in the packet header which as described in more detail below, is a 20-bit value. For a given stream, four of 20 bits are stored in each header successively (TS3-0, TS7-4, TS11-8, TS15-12, TS19-16). Native stream frequency (Freq_native) is obtained from link character clock frequency (Freq_link_char) as:

Eq(1) Freq_native = Freq_link_char * (TS19-0)/220

[0027] The transmitter 102 generates this time stamp by counting the number of native stream clocks in 220 cycles of the link character clock frequency period. The counter updates the value every 220 cycles of the link character clock. Since these two clocks are asynchronous with each other, the time stamp value will change by 1 over time. Between updates, the transmitter 102 will repeatedly send the same time stamp in the header of the given packet stream. A sudden change of the time stamp value (by more than 1 count) may be interpreted by the receiver as an indication of an unstable condition of the stream source.

[0028] It should be noted that, no time stamp is communicated for an audio stream. In this case, the source device informs the display device of the audio sample rate and number of bits per sample. By determining the audio rate based upon Eq(2) and the link character rate, the display device regenerates the original audio stream rate.

Eq(2) Audio rate = (audio sample rate) x (# bits per sample) x (# channels)

[0029] A main link data packet 400 shown in Fig. 4A includes a main link packet header 402 as shown in Fig. 4B that is formed of 16 bits where bits 3-0 are the Stream ID (SID) (indicating that maximum stream count is 16), bit 4 is the Time Stamp (TS) LSB. When bit 4 is equal to 1, this packet header has the least significant 4 bits of Time Stamp value (used only for uncompressed video stream). Bit 5 is a Video frame sequence bit which acts as the least significant bit of the frame counter which toggles from "0" to "1" or from "1" to "0" at the video frame boundary (used only for uncompressed video stream). Bits 7 and 6 are reserved whereas bits 8 through 10 are a 4-bit CRC (CRC) that checks errors for the previous eight bits. Bits 15-12 are Time Stamp/Stream ID Inversion. (TSP/SIDn) which for uncompressed video are used as four bits of 20-bit Time Stamp value.

[0030] One of the advantages of the inventive interface is the ability to multiplex different data streams each of which can be different formats as well as have certain main link data packets include a number of sub packets. For example, Fig. 5 shows a system 500 arranged to provide sub-packet enclosure and multiple-packet multiplexing in accordance with an embodiment of the invention. It should be noted that the system 500 is a particular embodiment of the system 200 shown in Fig. 2 and should therefore not be construed as limiting either the scope or intent of the invention. The system 500 includes a stream source multiplexer 502 included in the transmitter 102 used to combine a stream 1 supplemental data stream 504 with the data stream 210 to form a multiplexed data stream 506. The multiplexed data stream 506 is then forwarded to a link layer multiplexer 508 that combines any of a number of data streams to form a multiplexed main link stream 510 formed of a number of data packets 512 some of which may include any of a number of sub packets 514 enclosed therein. A link layer de-multiplexer 516 splits the multiplexed data stream 510 into its constituent data streams based on the stream IDs (SIDs) and associated sub packet headers while a stream sink de-multiplexer 518 further splits off the stream 1 supplemental data stream contained in the sub-packets.

[0031] Fig. 6 shows a high-level diagram of the multiplexed main link stream 600 as an example of the stream 510 shown in Fig. 5 when three streams are multiplexed over the main link 222. The three streams in this example are: UXGA graphics (Stream ID = 1), 1280x720p video (Stream ID = 2), and audio (Stream ID = 3). The small packet header size of main link packet 400 minimizes the packet overhead, which results in the very high link efficiency. The reason the packet header can be so small is that the packet attributes are communicated via the auxiliary channel 224 prior to the transmission of the packets over main link 222.

[0032] Generally speaking, the sub-packet enclosure is an effective scheme when the main packet stream is an uncompressed video since an uncompressed video data stream has data idle periods corresponding to the video-blanking period. Therefore, main link traffic formed of an uncompressed video stream will include series of Null special character packets during this period. By capitalizing on the ability to multiplex various data streams, certain implementations of the present invention use various methods to compensate for differences between the main link rate and the pixel data rate when the source stream is a video data stream. For example, as illustrated in Fig. 7, the pixel data rate is .5 Gb/sec, such that a bit of pixel data is transmitted every 2 ns. In this example, the link rate has been set to 1.25 Gb/sec, such that a bit of pixel data is transmitted each .8 ns. Here, transmitter 102 intersperses special characters between pixel data as illustrated in Fig. 8. Two special characters are disposed between a first bit of pixel data P1 and a second bit of pixel data P2. The special characters allow receiver 104 to distinguish each bit of pixel data. Interspersing the special characters between bits of pixel data also creates a steady stream of data that allows the link to maintain synchronization. In this example, the special characters are Null characters. No line buffer is needed for such methods, only a small FIFO, because the link rate is sufficiently fast. However, relatively more logic is required on the receiving side to reconstruct the video signal. The receiver needs to recognize when the special characters begin and end.

[0033] An alternative to the interspersing method is to alternate consecutive bits of pixel data with special characters, such as null values. For example, P1 through P4 could be fed into a line buffer included in the transmitter 104, then one or more null values could be fed into the buffer until more pixel data are available. Such implementations require a relatively larger buffer space than the interspersing

methods described above. In many such implementations, the time required to fill the line buffer will exceed the time required to transmit the data after the line buffer is full, due to the relatively high link speeds.

[0034] As discussed with reference to Fig. 5A, one of the advantages of the inventive interface is the ability to not only multiplex various data streams, but also the enclosing of any of a number of sub packets within a particular main link data packet. Fig. 9A shows a representative sub-packet 900 in accordance with an embodiment of the invention. The sub-packet 900 includes a sub-packet header 902 that in the described embodiment is 2 bytes and is accompanied by SPS (Sub-Packet Start) special character. If the main link data packet in which the sub-packet 900 is enclosed contains a packet payload in addition to the sub-packet 900, the end of the sub-packet 900 must be marked by SPE (Sub-Packet End) special character. Otherwise, the end of the main packet (as indicated by ensuing COM character in the example shown in Fig. 9B) marks the end of both the sub-packet 902 and the main packet into which it is enclosed. However, a sub-packet does not need to end with SPE when its enclosing main packet has no payload. Fig 9B shows an exemplary sub-packet format within a main link packet in accordance with an embodiment of the invention. It should be noted that the definition of the header field and sub-packet payload is dependent on the specific application profile that uses the sub-packet 902.

[0035] A particularly useful example of sub-packet enclosure usage is selective refresh of an uncompressed graphics image 1000 illustrated in Fig. 10. The attributes of the entire frame 1002 (Horizontal/Vertical Total, Image Width/Height, etc.) will be communicated via the auxiliary channel 224 since those attributes stay constant as long as the stream remains valid. In selective refresh operation, only a portion 1004 of the image 1000 is updated per video frame. The four X-Y coordinates of the updated

rectangle(s) (i.e., the portion 1004) must be transmitted every frame since the values of the rectangle coordinates changes from frame to frame. Another example is the transmission of color look-up table (CLUT) data for required for 256-color graphic data where the 8-bit pixel data is an entry to the 256-entry CLUT and the content of the CLUT must be dynamically updated.

[0036] The single bi-directional auxiliary channel 224 provides a conduit to for various support functions useful for link set up and supporting main link operations as well as to carry auxiliary application data such as USB traffic. For example, with the auxiliary channel 224, a display device can inform the source device of events such as sync loss, dropped packets and the results of training sessions (described below). For example, if a particular training session fails, the transmitter 102 adjusts the main link rate based upon pre-selected or determined results of the failed training session. In this way, the closed loop created by combining an adjustable, high speed main link with a relatively slow and very reliable auxiliary channel allows for robust operation over a variety of link conditions. It should be noted that in some cases (an example of which is shown in Fig. 5B), a logical bi-directional auxiliary channel 520 can be established using a portion 522 of the bandwidth of the main link 222 to transfer data from the source device 202 to the sink device 204 and a uni-directional back channel 524 from the sink device 204 to the source device 202. In some applications, use of this logical bi-directional auxiliary channel may be more desirable than using a half-duplex bi-directional channel described in Fig. 5A.

[0037] Prior to starting the transmission of actual packet data streams the transmitter 102 establishes a stable link through a link training session that is analogous in concept to the link setup of the modem. During link training, the main link transmitter 102 sends a pre-defined training pattern so that the receiver 104 can determine

whether it can achieve a solid bit/character lock. In the described embodiment, training related handshaking between the transmitter 102 and the receiver 104 is carried on the auxiliary channel. An example of a link training pattern is shown in Fig. 11 in accordance with an embodiment of the invention. As illustrated, during the training session, a phase 1 represents the shortest run length while phase 2 is the longest that are used by the receiver to optimize an equalizer. In phase 3, both bit lock and character lock are achieved as long as the link quality is reasonable. Typically, the training period is about 10 ms, in which time, approximately 107 bits of data are sent. If the receiver 104 does not achieve solid lock, it informs the transmitter 102 via the auxiliary channel 224 and the transmitter 102 reduces the link rate and repeats the training session.

[0038] In addition to providing a training session conduit, the auxiliary channel 224 can be also used to carry main link packet stream descriptions thereby greatly reducing the overhead of packet transmissions on the main link 222. Furthermore, the auxiliary channel 224 can be configured to carry Extended Display Identification Data (EDID) information replacing the Display Data Channel (DDC) found on all monitors (EDID is a VESA standard data format that contains basic information about a monitor and its capabilities, including vendor information, maximum image size, color characteristics, factory pre-set timings, frequency range limits, and character strings for the monitor name and serial number. The information is stored in the display and is used to communicate with the system through the DDC which sites between the monitor and the PC graphics adapter. The system uses this information for configuration purposes, so the monitor and system can work together). In what is referred to as an extended protocol mode, the auxiliary channel

can carry both asynchronous and isochronous packets as required to support additional data types such as keyboard, mouse and microphone.

[0039] Fig. 12 illustrates a logical layering 1200 of the system 200 in accordance with an embodiment of the invention. It should be noted that while the exact implementation may vary depending upon application, generally, a source (such as the video source 202) is formed of a source physical layer 1202 that includes transmitter hardware, a source link layer 1204 that includes multiplexing hardware and state machine (or firmware), and a data stream source 1206 such as Audio/Visual/Graphics hardware and associated software. Similarly, a display device includes a physical layer 1208 (including various receiver hardware), a sink link layer 1210 that includes de-multiplexing hardware and state machine (or firmware) and a stream sink 1212 that includes display/timing controller hardware and optional firmware. A source application profile layer 1214 defines the format with which the source communicates with the link layer 1204 and similarly, a sink application profile layer 1216 defines the format with which the sink 1212 communicates with the sink link layer 1210.

[0040] The various layers will now be discussed in more detail.

SOURCE DEVICE PHYSICAL LAYER

[0041] In the described embodiment, the source device physical layer 1202 includes an electrical sub layer 1202-1 and a logical sub layer 1202-2. The electrical sub layer 1202-1 includes all circuitry for interface initialization/operation such as hot plug/unplug detection circuit, drivers/receivers/termination resistors, parallel-to-serial/serial-to-parallel conversions, and spread-spectrum-capable PLL's . The logical sub layer 1202-2 includes circuitry for, packetizing/de-packetizing, data scrambling/de-scrambling, pattern generation for link training, time-base recovery

circuits, and data encoding/decoding such as 8B/10B (as specified in ANSI X3.230-1994, clause 11) that provides 256 link data characters and twelve control characters (an example of which is shown as Fig. 13) for the main link 222 and Manchester II for the auxiliary channel 224 (see Fig. 14).

[0042] It should be noted that the 8B/10B encoding algorithm is described, for example, in U.S. Patent Number 4,486,739, which is hereby incorporated by reference. As known by those of skill in the art, the 8B/10B code is a block code that encodes 8-bit data blocks into 10-bit code words for serial transmission. In addition, the 8B/10B transmission code converts a byte wide data stream of random 1s and 0s into a DC balanced stream of 1s and 0s with a maximum run length of 5. Such codes provide sufficient signal transitions to enable reliable clock recovery by a receiver, such as transceiver 110. Moreover, a DC balanced data stream proves to be advantageous for fiber optic and electromagnetic wire connections. The average number of 1s and 0s in the serial stream is be maintained at equal or nearly equal levels. The 8B/10B transmission code constrains the disparity between the number of 1s and 0s to be –2, 0, or 2 across 6 and 4 bit block boundaries. The coding scheme also implements additional codes for signaling, called command codes.

[0043] It should be noted that in order to avoid the repetitive bit patterns exhibited by uncompressed display data (and hence, to reduce EMI), data transmitted over main link 222 is first scrambled before 8B/10B encoding. All data except training packets and special characters will be scrambled. The scrambling function is implemented with Linear Feedback Shift Registers (LFSRs). When data encryption is enabled, the initial value of an LFSR seed is dependent on an encryption key set. If it is data scrambling without encryption, the initial value will be fixed.

[0044] Since data stream attributes are transmitted over the auxiliary channel 224, the main link packet headers serve as stream identification numbers thereby greatly reducing overhead and maximizing link bandwidth. It should also be noted that neither the main link 222 nor the auxiliary link 224 has separate clock signal lines. In this way, the receivers on main link 222 and auxiliary link 224 sample the data and extract the clock from the incoming data stream. Fast phase locking for any phase lock loop (PLLs) circuit in the receiver electrical sub layer is important for since the auxiliary channel 224 is half-duplex bi-directional and the direction of the traffic changes frequently. Accordingly, the PLL on the auxiliary channel receiver phase locks in as few as 16 data periods thanks to the frequent and uniform signal transitions of Manchester II (MII) code

[0045] At link set up time, the data rate of main link 222 is negotiated using the handshake over auxiliary channel 224. During this process, known sets of training packets are sent over the main link 222 at the highest link speed. Success or failure is communicated back to the transmitter 102 via the auxiliary channel 224. If the training fails, main link speed is reduced and training is repeated until successful. In this way, the source physical layer 1102 is made more resistant to cable problems and therefore more suitable for external host to monitor applications. However, unlike conventional display interfaces, the main channel link data rate is decoupled from the pixel clock rate. A link data rate is set so that link bandwidth exceeds the aggregate bandwidth of the transmitted streams.

SOURCE DEVICE LINK LAYER

[0046] The source link layer 1204 handles the link initialization and management. For example, upon receiving a hot plug detect event generated upon monitor power-up or connection of the monitor cable from the source physical layer 1202, the source

device link layer 1204 evaluates the capabilities of the receiver via interchange over the auxiliary channel 224 to determine a maximum main link data rate as determined by a training session, the number of time-base recovery units on the receiver, available buffer size on both ends, availability of USB extensions and then notifies the stream source 1206 of an associated hot plug event. In addition, upon request from the stream source 1206, the source link layer 1204 reads the display capability (EDID or equivalent). During a normal operation, the source link layer 1204 sends the stream attributes to the receiver 104 via the auxiliary channel 224, notifies the stream source 1204 whether the main link 222 has enough resource for handling the requested data streams, notifies the stream source 1204 of link failure events such as sync loss and buffer overflow, and sends MCCS commands submitted by the stream source 1204 to the receiver via the auxiliary channel 224. All communications between the source link layer 1204 and the stream source/sink use the formats defined in the application profile layer 1214.

APPLICATION PROFILE LAYER (SOURCE AND SINK)

[0047] In general, the Application Profile Layer defines formats with which a stream source (or sink) will interface with the associated link layer. The formats defined by the application profile layer are divided into the following categories, Application independent formats (Link Message for Link Status inquiry) and Application dependent formats (main link data mapping, time-base recovery equation for the receiver, and sink capability/stream attribute messages sub-packet formats, if applicable). The Application Profile Layer supports the following color formats 24-bit RGB, 16-bit RG2565, 18-bit RGB, 30-bit RGB, 256-color RGB (CLUT based), 16-bit, CbCr422, 20-bit YCbCr422, and 24-bit YCbCr444.

[0048] For example, the display device application profile layer (APL) 1214 is essentially an application-programming interface (API) describing the format for Stream Source/Sink communication over the main link 222 that includes a presentation format for data sent to or received from the interface 100. Since some aspects of the APL 1214 (such as the power management command format) are baseline monitor functions, they are common to all uses of the interface 100. Whereas other non-baseline monitor functions, such as such as data mapping format and stream attribute format, are unique to an application or a type of isochronous stream that is to be transmitted. Regardless of the application, the stream source 1204 queries the source link layer 1214 to ascertain whether the main link 222 is capable of handling the pending data stream(s) prior to the start any packet stream transmission on the main link 222.

[0049] When it is determined that the main link 222 is capable of supporting the pending packet stream(s), the stream source 1206 sends stream attributes to the source link layer 1214 that is then transmitted to the receiver over the auxiliary channel 224. These attributes are the information used by the receiver to identify the packets of a particular stream, to recover the original data from the stream and to format it back to the stream's native data rate. The attributes of the data stream are application dependent.

[0050] In those cases where the desired bandwidth is not available on the main link 222, the stream source 1214 may take corrective action by, for example, reducing the image refresh rate or color depth.

DISPLAY DEVICE PHYSICAL LAYER

[0051] The display device physical layer 1216 isolates the display device link layer 1210 and the display device APL 1216 from the signaling technology used for link data transmission/reception. The main link 222 and the auxiliary channel 224 have their own physical layers, each consisting of a logical sub layer and an electrical sub layer that includes the connector specification. For example, the half-duplex, bi-directional auxiliary channel 224 has both a transmitter and a receiver at each end of the link as shown in Fig 15. An auxiliary link transmitter 1502 is provided with link characters by a logical sub layer 1208-1 that are then serialized serialized and transmitted to a corresponding auxiliary link receiver 1504. The receiver 1504, in turn, receives serialized link character from the auxiliary link 224 and de-serializes the data at a link character clock rate. It should be noted that the major functions of the source logical sub layers include signal encoding, packetizing, data scrambling (for EMI reduction), and training pattern generation for the transmitter port. While for the receiver port, the major functions of the receiver logical sub layer includes signal decoding, de-packetizing, data de-scrambling, and time-base recovery.

AUXILIARY CHANNEL

[0052] The major functions of auxiliary channel logical sub layer include data encoding and decoding, framing/de-framing of data and there are two options in auxiliary channel protocol: standalone protocol (limited to link setup/management functions in a point-to-point topology) is a lightweight protocol that can be managed by the Link Layer state-machine or firmware and extended protocol that supports other data types such as USB traffic and topologies such as daisy-chained sink devices. It should be noted that the data encoding and decoding scheme is identical regardless of the protocol whereas framing of data differs between the two.

[0053] Still referring to Fig. 15, the auxiliary channel electrical sub layer contains the transmitter 1502 and the receiver 1504. The transmitter 1502 is provided with link characters by the logical sub layer, which it serializes and transmits out. The receiver 1504 receives serialized link character from the link layer and subsequently de-serializes it at link character clock rate. The positive and negative signals of auxiliary channel 224 are terminated to ground via 50-ohm termination resistors at each end of the link as shown. In the described implementation, the drive current is programmable depending on the link condition and ranges from approximately 8mA to approximately 24mA resulting in a range of Vdifferential_pp of approximately 400mV to approximately 1.2V. In electrical idle modes, neither the positive nor the negative signal is driven. When starting transmission from the electrical idle state, the SYNC pattern must be transmitted and the link reestablished. In the described embodiment, the SYNC pattern consists of toggling a auxiliary channel differential pair signals at clock rate 28 times followed by four 1's in Manchester II code. The auxiliary channel master in the source device detects hot-plug and hot-unplug events by periodically driving or measuring the positive and negative signals of auxiliary channel 224.

MAIN LINK

[0054] In the described embodiment, the main link 222 supports discrete, variable link rates that are integer multiples of the local crystal frequency (see Fig 3 for a representative set of link rates consonant with a local crystal frequency of 24-MHz). As shown in Fig. 16, the main link 222 (being an unidirectional channel) has only a transmitter 1602 at the source device and only a receiver 1604 at the display device.

[0055] As shown, the cable 1604 takes the form includes a set of twisted pair wires, one for each of the Red (R), Green(G), and Blue(B) video signals provides in a

typical RGB color based video system (such as PAL based TV systems). As known by those of skill in the art, twisted pair cable is a type of cable that consists of two independently insulated wires twisted around one another. One wire carries the signal while the other wire is grounded and absorbs signal interference. It should be noted that in some other systems, the signals could also be component based signals (Pb, Pr, Y) used for NTSC video TV systems. Within the cable, each twisted pair is individually shielded. Two pins for +12V power and ground are provided. The characteristics impedance of each differential pair is 100 ohms +/- 20%. The entire cable is also shielded. This outer shield and individual shields are shorted to the connector shells on both ends. The connector shells are shorted to ground in a source device. A connector 1700 as shown in Fig. 17 has 13 pins in one row having a pinout that is identical both for the connector on the source device end and that on the display device end. The source device supplies the power.

[0056] The main link 222 is terminated on both ends and since the main link 222 is AC coupled, the termination voltage can be anywhere between 0V (ground) to +3.6V. In the described implementation, the drive current is programmable depending on the link condition and ranges from approximately 8mA to approximately 24mA resulting in a range of Vdifferential_pp of approximately 400mV to approximately 1.2V. The minimum voltage swing is selected for each connection using a training pattern. An electrical idle state is provided for power management modes. In electrical idle, neither the positive nor the negative signals are driven. When starting a transmission from electrical idle state, the transmitter must conduct a training session in order re-establish the link with the receiver.

STATE DIAGRAMS

[0057] The invention will now be described in terms of state diagrams shown in Figs. 18 and 19 described below. Accordingly, Fig.18 shows the source state diagram described below. At an off state 1802, the system is off such that the source is disabled. If the source is enabled, then the system transitions to a standby state 1804 suitable for power saving and receiver detection. In order to detect whether or not the receiver is present (i.e., hot plug/play), the auxiliary channel is periodically pulsed (such as for 1us every 10ms) and a measure of a voltage drop across the termination resistors during the driving is measured. If it is determined that a receiver is present based upon the measured voltage drop, then the system transitions to a detected receiver state 1806 indicating that a receiver has been detected, i.e, a hot plug event has been detected. If, however, there is no receiver detected, then the receiver detection is continued until such time, if ever, a receiver is detected or a timeout has elapsed. It should be noted that in some cases the source device may choose to go to "OFF" state from which no further display detection is attempted.

[0058] If at the state 1806 a display hot unplug event is detected, then the system transitions back to the standby state 1804. Otherwise the source drives the auxiliary channel with a positive and negative signal to wake up receiver and the receiver's subsequent response, if any, is checked. If there is no response received, then the receiver has not woken up and source remains in the state 1806. If, however, a signal is received from the display, then the display has woken up and the source is ready read the receiver link capabilities (such as max link rate, buffer size, and number of time-base recovery units) and the system transitions to a main link initialization state 1808 and is ready to commence a training start notification phase.

[0059] At this point, a training session is started by sending a training pattern over the main link at a set link rate and checks an associated training status. The receiver sets a pass/fail bit for each of three phases and the transmitter will proceed to the next phase upon detection of pass only such that when a pass is detected, the main link is ready at that link rate. At this point, the interface transitions to a normal operation state 1510, otherwise, the link rate is reduced and the training session is repeated. During the normal operation state 1810, the source continues to periodically monitor a link status index, which if fails, a hot unplug event is detected and the system transitions to the standby state 1804 and waits for a hot plug detection event. If, however, a sync loss is detected, then the system transitions to state 1808 for a main link re-initiation event.

[0060] Fig 19 shows the display state diagram 1900 described below. At a state 1902, no voltage is detected, the display goes to an OFF state. At a standby mode state 1904, both main link receiver and auxiliary channel slave are in electrical idle, a voltage drop across the termination resistors of auxiliary channel slave port are monitored for a predetermined voltage. If the voltage is detected, then the auxiliary channel slave port is turned on indicating a hot plug event and the system moves to a display state 1906, otherwise, the display remains in the standby state 1904. At the state 1906 (main link initialization phase), if a display is detected, then the auxiliary slave port is fully turned on, and the transmitter responds to a receiver link capability read command and the display state transitions to 1908, otherwise, if there is no activity on the auxiliary channel for more than a predetermined period of time then the auxiliary channel slave port is put into the to standby state 1904.

[0061] During a training start notification phase, the display responds to the training initiation by the transmitter by adjusting the equalizer using training patterns,

updating the result for each phase. If the training fails, then wait for another training session and if the training passes, then go to normal operation state 1910. If there is no activity on the auxiliary channel or on the main link (for training) for more than a predetermined (10ms, for example), the auxiliary channel slave port is set to the standby state 1904.

[0062] Figs. 20 – 24 show particular implementations of the cross platform display interface.

[0063] Fig. 20 shows a PC motherboard 2000 having an on-board graphics engine 2002 that incorporates a transmitter 2004 in accordance with the invention. It should be noted that the transmitter 2004 is a particular example of the transmitter 102 shown in Fig. 1. In the described embodiment, the transmitter 2004 is coupled to an connector 2006 (along the lines of the connector 1700) mounted on the motherboard 2000 which in turn is connected to a display device 2008 by way of a twisted pair cable 2010 couples a display device 2010.

[0064] As known in the art, PCI Express (developed by Intel Corporation of Santa Clara, CA) is a high-bandwidth, low pin count, serial, interconnect technology that also maintains software compatibility with existing PCI infrastructure. In this configuration, the PCI Express port is augmented to become compliant with the requirements of the cross platform interface which can directly drive a display device either using a motherboard mounted connector as shown.

[0065] In situations where it is not practical to mount the connector on the motherboard, the signals can be routed through the SDVO slot of the PCI Express motherboard and brought to the back of the PC using a passive card connector as shown in Fig. 21. As is the case with the current generation of add-in graphics cards, a add-in graphics card can supplant the onboard graphics engine as shown in Fig. 23.

[0066] In the case of notebook applications, the transmitter on the motherboard graphics engine would drive through internal cabling, an integrated receiver/TCON which would drive the panel directly. For the most cost effective implementation, the receiver/TCON would be mounted on the panel thereby reducing the number of interconnect wires to 8 or 10 as shown in Fig. 24

[0067] All of the above examples assume integrated transmitters. However, is it quite feasible to implement as a standalone transmitter integrating into PCI and PCI Express environments through the AGP or SDVO slots, respectively. A standalone transmitter will enable output streams without any change in graphics hardware or software.

FLOWCHART EMBODIMENTS

[0068] The methodology of the invention will now be described in terms of a number of flowcharts each describing a particular process for enabling the invention. Specifically, FIGS 25 – 29 describe a number of interrelated processes that when used singly or in any combination described aspects of the invention.

[0069] Fig. 25 shows a flowchart detailing a process 2500 for determining an operational mode of the interface 100 in accordance with an embodiment of the invention. In this process, the operational mode will only be set to a digital mode if the video source and the display device are both digital. Otherwise, the operational mode will be set to analog mode. It should be noted that "analog mode" in this context can include both conventional VGA mode as well as enhanced analog mode having differential analog video with embedded alignment signal and bi-directional sideband. This enhanced analog mode will be described below.

[0070] In step 2502, a video source is interrogated to determine whether the video source supports analog or digital data. If the video source supports only analog data,

the operational mode of coupling device 100 will be set to analog (step 2508), then the process will end (step 2512).

[0071] If the video source can output digital data, the process continues to step 2506. The display device is then interrogated to determine whether the display device is configured to receive digital data. If the display device supports only analog data, the operational mode of coupling device will be set to analog (step 2508), then the process will end (step 2512). Otherwise, the operational mode of the coupling device is set to digital (step 2510). For example, a processor may control switches within the coupling device to set the mode to digital. In general, the coupling device is configured to operate in a fully digital mode only when both the video source and the video sink are operating in a corresponding digital mode.

[0072] Fig. 26 shows a flowchart detailing a process 2600 for providing a real time video image quality check in accordance with some aspects of the invention. In this example, all determinations of process 2600 are made by a processor coupled to the display interface.

[0073] In step 2600, a video signal is received from a video source. Next, a signal quality test pattern is provided by the video source associated with the received video signal (step 2602). In step 2604, a determination of a bit error rate is made, based upon the quality test pattern. Then, a determination is made of whether the bit error rate is greater than a threshold value (step 2606). If the bit error rate is determined to not be greater than the threshold value, then a determination is made (step 2614) of whether or not there are more video frames. If it is determined that there are more video frames, then the process returns to step 2600. Otherwise, the process ends.

[0074] However, if the bit error rate is determined to be greater than the threshold value in step 2606, a determination is made (step 2608) as to whether the bit rate is

greater than a minimum bit rate. If the bit rate is greater than a minimum bit rate, then the bit rate is lowered (step 2610) and the process returns to step 2606. If the bit rate is not greater than the minimum bit rate, then the mode is changed to analog mode (step 2612) and the process ends.

[0075] Fig. 27 shows a flowchart for a link set up process 2700 in accordance with an embodiment of the invention. The process 2700 begins at 2702 by the receiving of a hot plug detection event notification. At 2704 a main link inquiry is made by way of an associated auxiliary channel to determine a maximum data rate, a number of time base recovery units included in a receiver, and available buffer size. Next, at 2706, the maximum link data rate is verified by way of a training session and at 2708, a data stream source is notified of the hot plug event. At 2710, the capabilities of the display (using EDID, for example) are determined by way of the auxiliary channel and the display responds to the inquires at 2712 which, in turn, results a collaboration of the main link training session at 2714.

[0076] Next, at 2716, the stream source sends stream attributes to the receiver by way of the auxiliary channel and at 2718, the stream sources are further notified whether the main link is capable of supporting the requested number of data streams at 2720. At 2722, the various data packets are formed by adding associated packet headers and the multiplexing of a number of source streams is scheduled at 2724. At 2726 a determination is made whether or not the link status is OK. When the link status is not OK, then the source(s) are notified of a link failure event at 2728, otherwise, the link data streams are reconstructed into the native streams based upon the various packet headers at 2730. At 2732, the reconstructed native data streams are then passed to the display device.

[0077] Fig. 28 shows a flowchart detailing a process 2800 for performing a training session in accordance with an embodiment of the invention. It should be noted that the training session process 2800 is one implementation of the operation 2506 described in Fig. 25. A training session is started at 2802 by sending a training pattern over the main link at a set link rate to the receiver. A typical link training pattern is shown in Fig. 11 in accordance with an embodiment of the invention. As illustrated, during the training session, a phase 1 represents the shortest run length while phase 2 is the longest. The receiver is to use these two phases to optimize the equalizer. In phase 3, both bit lock and character lock are achieved as long as the link quality is reasonable. At 2804, the receiver checks an associated training status and based upon the training status check, the receiver sets a pass/fail bit for each of three phases and the transmitter at 2806. At each phase, the receiver will proceed to the next phase upon detection of pass only and at 2810 and if the receiver does not detect a pass then the receiver reduces the link rate and repeats the training session. The main link is ready at that link rate at which a pass is detected at 2812.

[0078] Fig. 29 illustrates a computer system 2900 employed to implement the invention. Computer system 2900 is only an example of a graphics system in which the present invention can be implemented. Computer system 2900 includes central processing unit (CPU) 1510, random access memory (RAM) 2920, read only memory (ROM) 2925, one or more peripherals 2930, graphics controller 2960, primary storage devices 2940 and 2950, and digital display unit 2970. As is well known in the art, ROM acts to transfer data and instructions uni-directionally to the CPUs 2910, while RAM is used typically to transfer data and instructions in a bi-directional manner. CPUs 2910 may generally include any number of processors. Both primary storage devices 2940 and 2950 may include any suitable computer-readable media. A

secondary storage medium 880, which is typically a mass memory device, is also coupled bi-directionally to CPUs 2910 and provides additional data storage capacity. The mass memory device 880 is a computer-readable medium that may be used to store programs including computer code, data, and the like. Typically, mass memory device 880 is a storage medium such as a hard disk or a tape which generally slower than primary storage devices 2940, 2950. Mass memory storage device 880 may take the form of a magnetic or paper tape reader or some other well-known device. It will be appreciated that the information retained within the mass memory device 880, may, in appropriate cases, be incorporated in standard fashion as part of RAM 2920 as virtual memory.

[0079] CPUs 2910 are also coupled to one or more input/output devices 890 that may include, but are not limited to, devices such as video monitors, track balls, mice, keyboards, microphones, touch-sensitive displays, transducer card readers, magnetic or paper tape readers, tablets, styluses, voice or handwriting recognizers, or other well-known input devices such as, of course, other computers. Finally, CPUs 2910 optionally may be coupled to a computer or telecommunications network, e.g., an Internet network or an intranet network, using a network connection as shown generally at 2995. With such a network connection, it is contemplated that the CPUs 2910 might receive information from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using CPUs 2910, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave. The above-described devices and materials will be familiar to those of skill in the computer hardware and software arts.

[0080] Graphics controller 2960 generates analog image data and a corresponding reference signal, and provides both to digital display unit 2970. The analog image data can be generated, for example, based on pixel data received from CPU 2910 or from an external encode (not shown). In one embodiment, the analog image data is provided in RGB format and the reference signal includes the VSYNC and HSYNC signals well known in the art. However, it should be understood that the present invention can be implemented with analog image, data and/or reference signals in other formats. For example, analog image data can include video signal data also with a corresponding time reference signal.

[0081] Although only a few embodiments of the present invention have been described, it should be understood that the present invention may be embodied in many other specific forms without departing from the spirit or the scope of the present invention. The present examples are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0082] While this invention has been described in terms of a preferred embodiment, there are alterations, permutations, and equivalents that fall within the scope of this invention. It should also be noted that there are many alternative ways of implementing both the process and apparatus of the present invention. It is therefore intended that the invention be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

In the claims

1. A packet based display interface arranged to couple a multimedia source device to a multimedia sink device, comprising:

a transmitter unit coupled to the source device arranged to receive a source video data stream in accordance with a native video data rate;

a receiver unit coupled to the sink device; and

a linking unit coupling the transmitter unit and the receiver unit arranged to transfer the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other .

2. A packet based display interface as recited in claim 1, wherein the multimedia data packet stream is one of a number of multimedia data packet streams each having an associated adjustable data stream link rate that is independent of the native stream rate.

3. A display interface as recited in claim 1, wherein the link unit further comprises:

a unidirectional main link arranged to carry the multimedia data packets from the transmitter unit to the receiver unit; and

a bi-directional auxiliary channel arranged to transfer information between the transmitter unit and the receiver unit and vice versa.

4. A display interface as recited in claim 3, wherein the bi-directional auxiliary channel is formed of a uni-directional back channel configured to carry

information from the sink device to the source device and a uni-directional forward channel included as part of the main channel for carrying information from the source device to the sink device in concert with the back channel.

5. A display interface as recited in claim 2, wherein the main link unit further comprises:

a number of virtual links each being associated with a particular one of the multi media data packet streams wherein each of said virtual links has an associated virtual link bandwidth and a virtual link rate.

6. A display interface as recited in claim 5, wherein a main link bandwidth is at least equal to an aggregate of the virtual link bandwidths.

7. A display interface as recited in 1, wherein the source data stream is packetized over the respective virtual link based upon a mapping definition.

8. A method of coupling a multimedia source device to a multimedia sink device, comprising:

receiving source video data in accordance with a native video data rate;

transferring the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other.

9. A method as recited in claim 8, wherein the multimedia data packet stream is one of a number of multimedia data packet streams each having an

associated adjustable data stream link rate that is independent of the native stream rate.

10. A method as recited in claim 8, wherein the link unit further comprises:

a unidirectional main link arranged to carry the multimedia data packets from the transmitter unit to the receiver unit; and

a bi-directional auxiliary channel arranged to transfer information between the transmitter unit and the receiver unit and vice versa.

11. A method as recited in claim 10, wherein the bi-directional auxiliary channel is formed of a uni-directional back channel configured to carry information from the sink device to the source device and a uni-directional forward channel included as part of the main channel for carrying information from the source device to the sink device in concert with the back channel.

12. A method as recited in claim 8, wherein the main link unit further comprises:

a number of virtual links each being associated with a particular one of the multi media data packet streams wherein each of said virtual links has an associated virtual link bandwidth and a virtual link rate.

13. A method as recited in claim 12, wherein a main link bandwidth is at least equal to an aggregate of the virtual link bandwidths.

14. A method as recited in claim 8, wherein the source data stream is packetized over the respective virtual link based upon a mapping definition.

15. Computer program product for coupling a multimedia source device to a multimedia sink device, comprising:

computer code for receiving source video data in accordance with a native video data rate;

computer code for transferring the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other; and

computer readable medium for storing the computer code.

16. Computer program product as recited in claim 15, wherein the multimedia data packet stream is one of a number of multimedia data packet streams each having an associated adjustable data stream link rate that is independent of the native stream rate.

17. Computer program product as recited in claim 15, wherein the link unit further comprises:

a unidirectional main link arranged to carry the multimedia data packets from the transmitter unit to the receiver unit; and

a bi-directional auxiliary channel arranged to transfer information between the transmitter unit and the receiver unit and vice versa.

18. Computer program product as recited in claim 17, wherein the bi-directional auxiliary channel is formed of a uni-directional back channel configured to carry information from the sink device to the source device and a uni-directional forward channel included as part of the main channel for carrying information from the source device to the sink device in concert with the back channel.

ABSTRACT OF THE DISCLOSURE

VIDEO INTERFACE ARRANGED TO PROVIDE PIXEL DATA INDEPENDENT OF A LINK CHARACTER CLOCK

[0083] A packet based display interface arranged to couple a multimedia source device to a multimedia sink device is disclosed that includes a transmitter unit coupled to the source device arranged to receive a source packet data stream in accordance with a native stream rate, a receiver unit coupled to the sink device, and a linking unit coupling the transmitter unit and the receiver unit arranged to transfer the video data in the form of a number of main link characters at a link character clock rate that is independent of the native stream rate such that video data and the link character clock are asynchronous to each other .

Fig. 2A