(12) UK Patent (19) GB (11) 2485613 (13) B

(45) Date of B Publication 13.08.2014

(54) Title of the Invention: Display control device

(51) INT CL: ***H04N 5/44*** (2011.01) ***H04N 19/423*** (2014.01) *H04N 19/625* (2014.01) *H04N 19/63* (2014.01)

(21) Application No: **1019766.3**

(22) Date of Filing: **22.11.2010**

(43) Date of A Publication **23.05.2012**

(56) Documents Cited:

EP 0847203 A2
EP 0782345 A1
WO 1998/026606 A1
US 6088395 A
US 6020924 A
US 20080298463 A1
US 20060130103 A1

(58) Field of Search:

As for published application 2485613 A viz:
INT CL **H04N**
Other: **WPI,EPODOC**
updated as appropriate

Additional Fields
Other: **No further FoS at B**

(72) Inventor(s):
William Stoye

(73) Proprietor(s):
DisplayLink (UK) Limited
(Incorporated in the United Kingdom)
Mount Pleasant House, Mount Pleasant,
CAMBRIDGE, CB3 0RN, United Kingdom

(74) Agent and/or Address for Service:
Mathys & Squire LLP
The Shard, 32 London Bridge Street, LONDON,
SE1 9SG, United Kingdom

GB 2485613 B

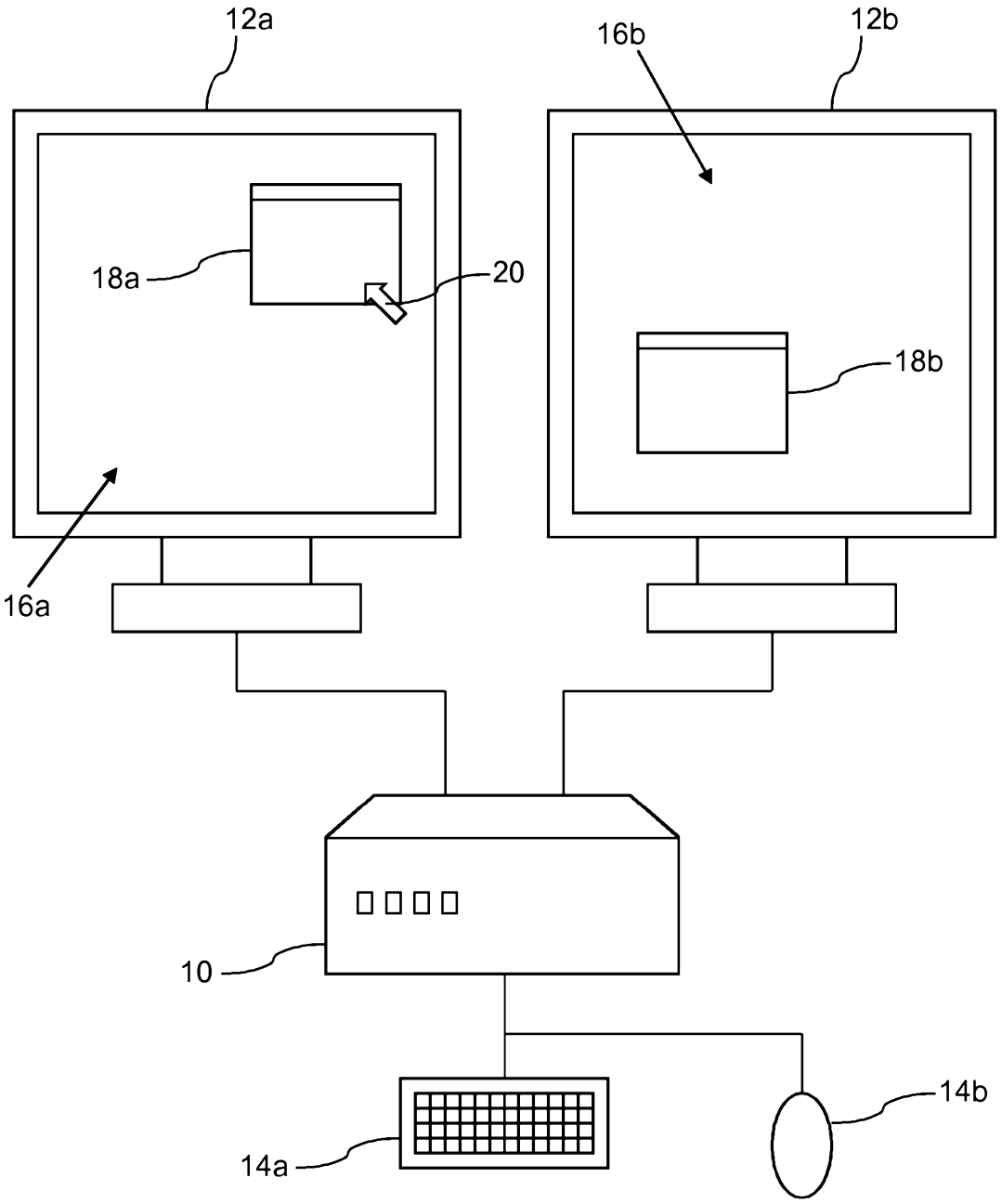


Fig. 1

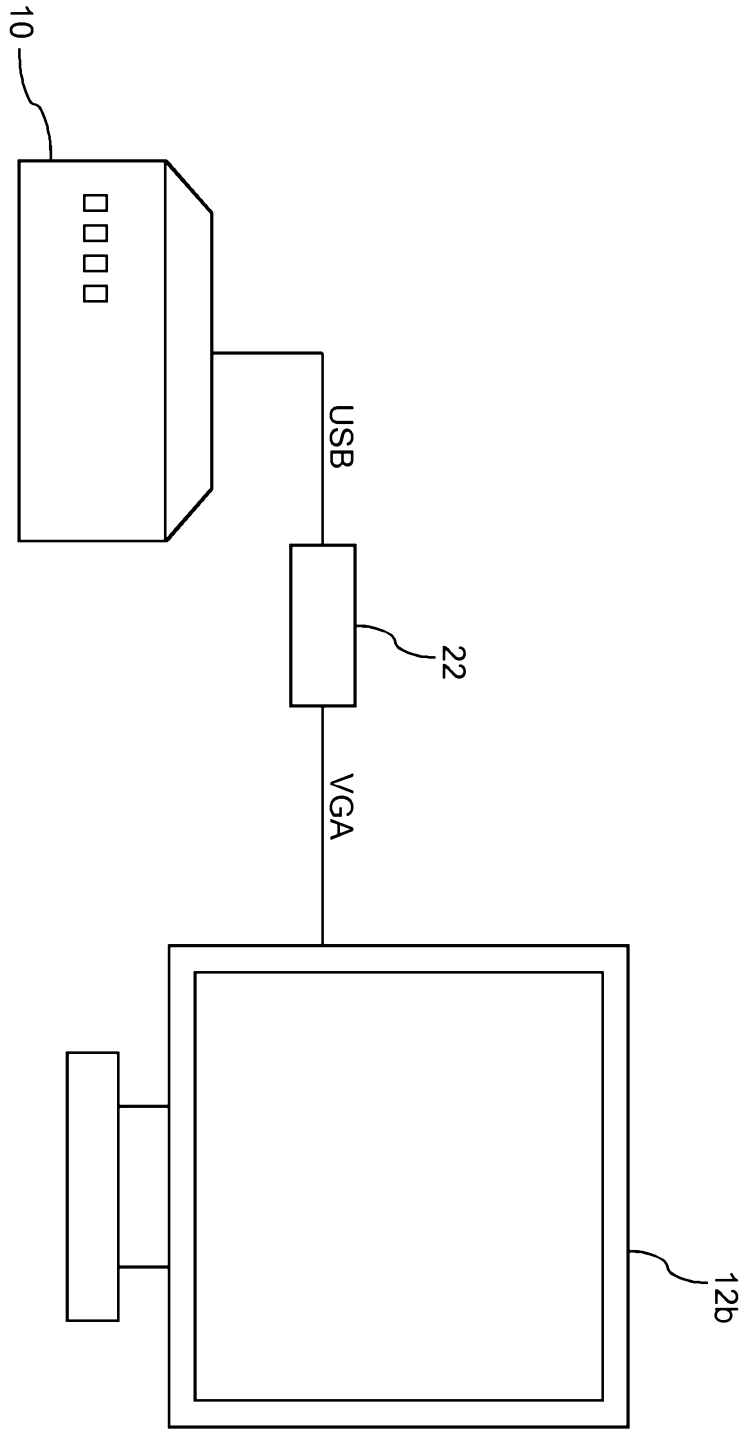


Fig. 2

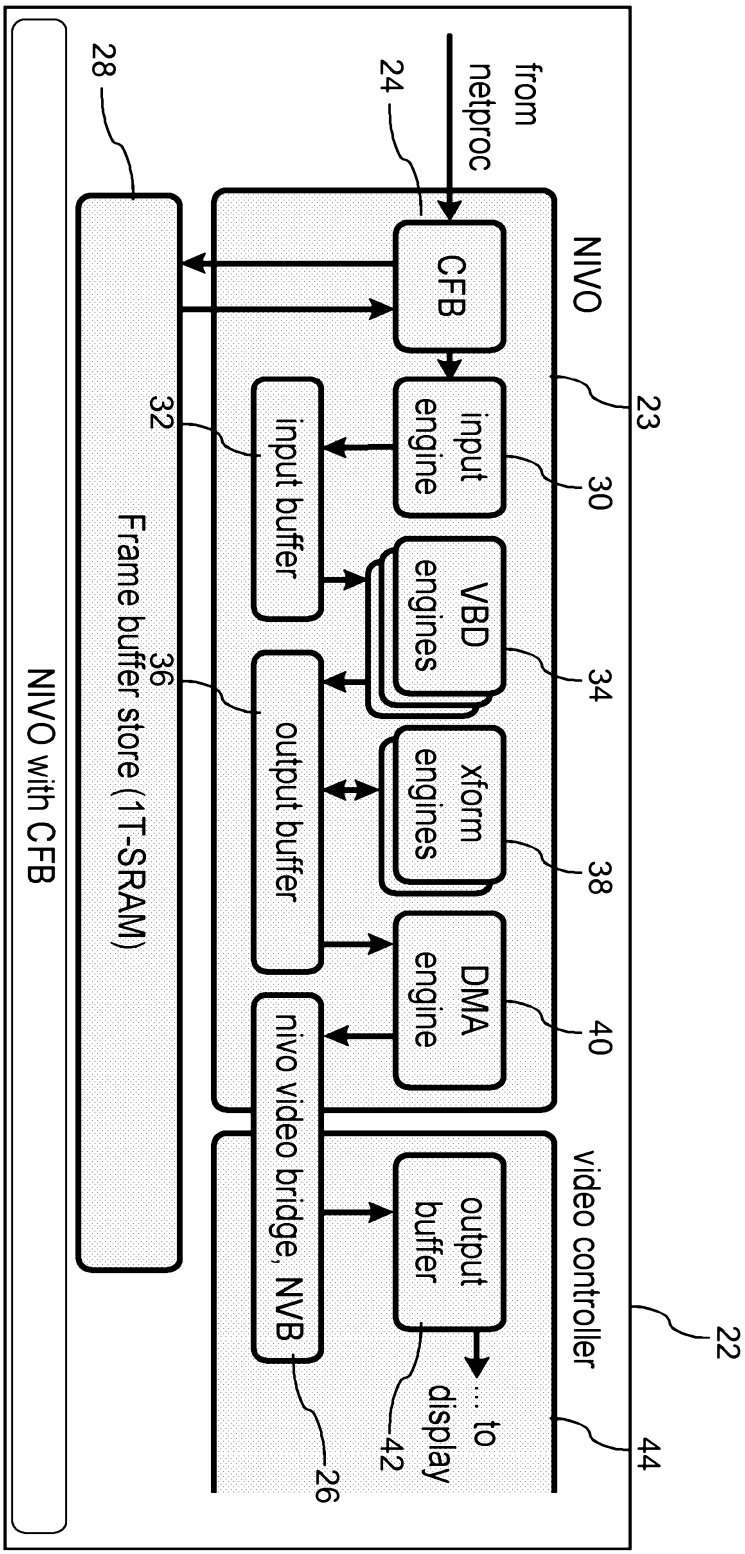

NIVO
23
from netproc
24
CFB
input engine
30
input buffer
32
VBD engines
34
output buffer
36
xform engines
38
DMA engine
40
nivo video bridge, NVB
26
Frame buffer store (1T-SRAM)
28
NIVO with CFB
video controller
22
output buffer
42
... to display
44

Fig. 3

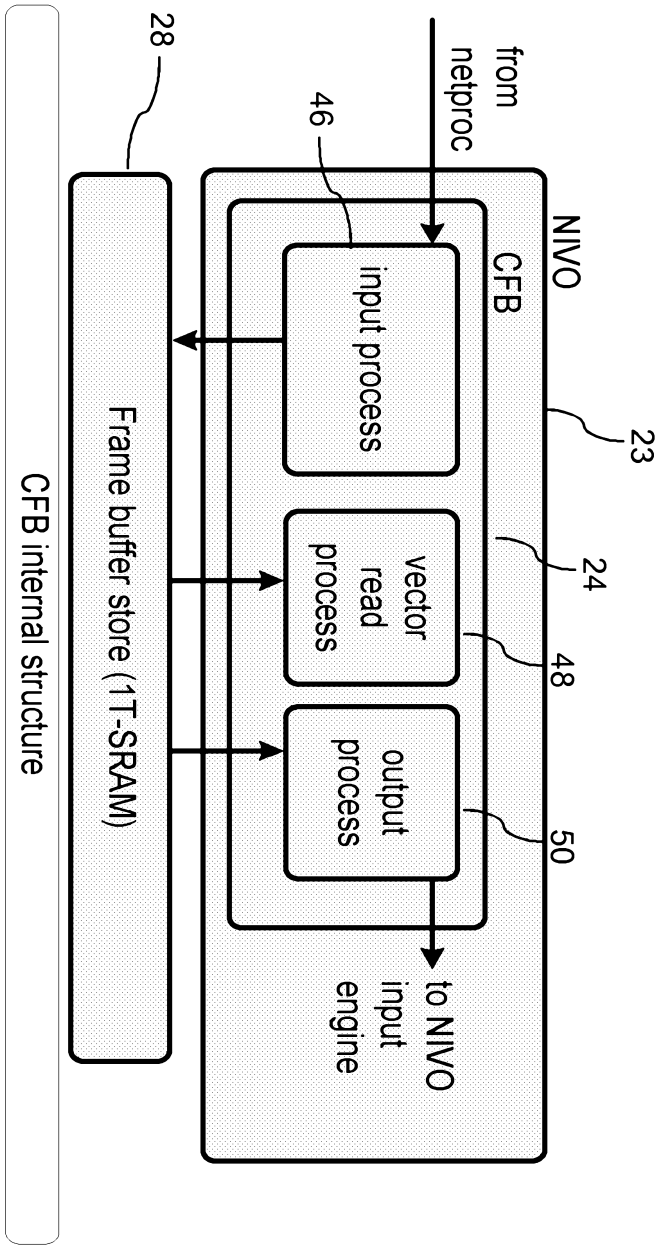

CFB internal structure
Frame buffer store (1T-SRAM)
28
NIVO
23
CFB
24
from netproc
46
input process
vector read process
48
output process
50
to NIVO input engine

Fig. 4

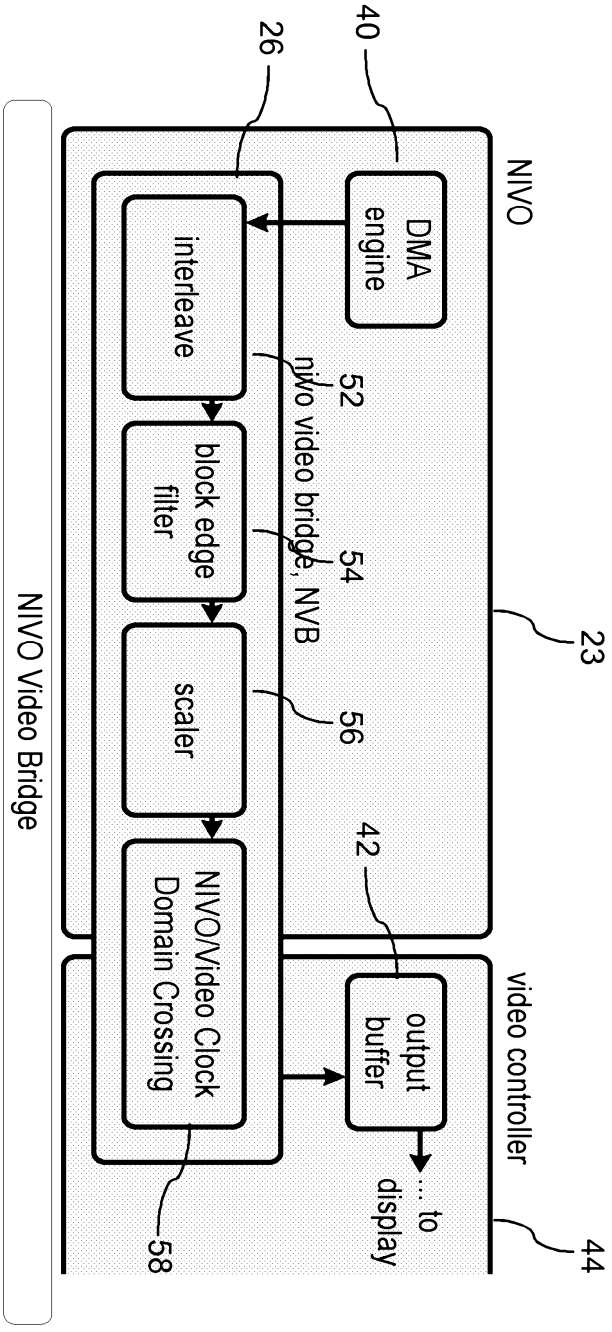

NIVO
DMA engine
interleave
block edge filter
scaler
NIVO/Video Clock Domain Crossing
nivo video bridge, NVB
video controller
output buffer
... to display
NIVO Video Bridge
40
26
52
54
56
42
58
23
44

Fig. 5

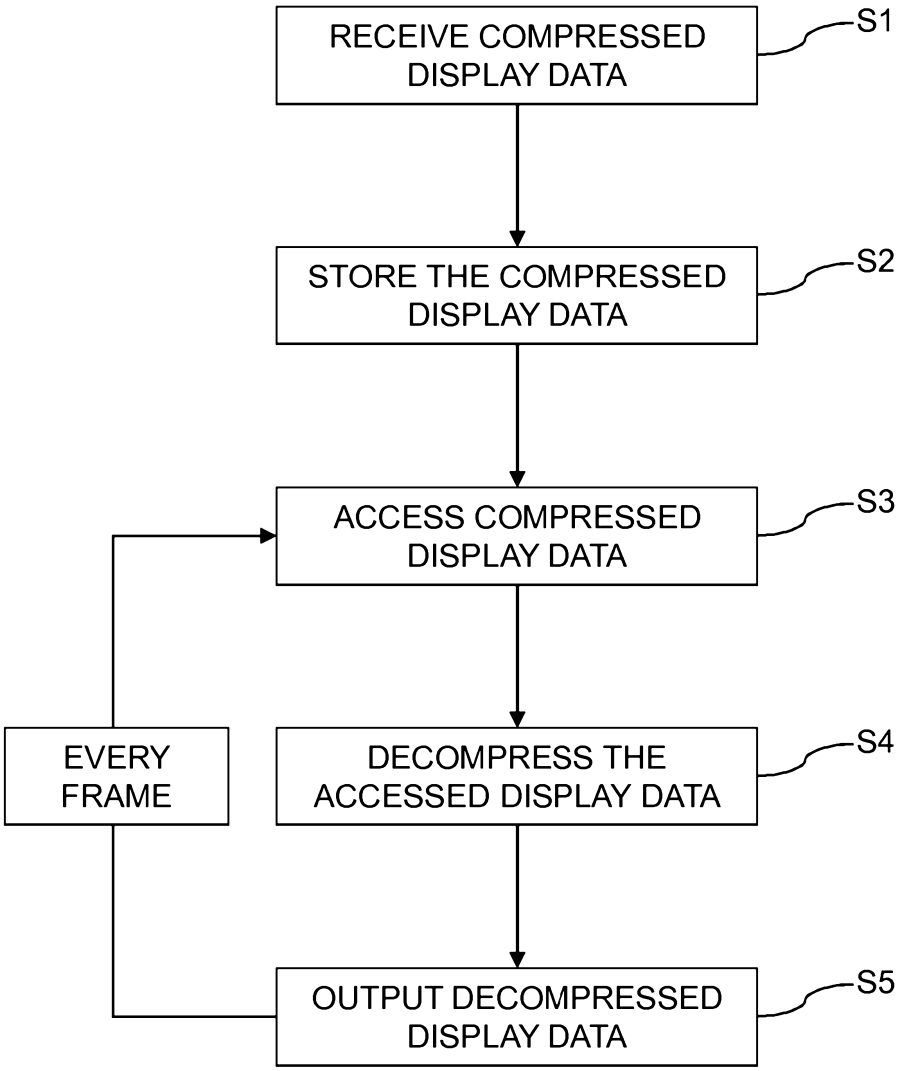

RECEIVE COMPRESSED DISPLAY DATA
S1
STORE THE COMPRESSED DISPLAY DATA
S2
ACCESS COMPRESSED DISPLAY DATA
S3
EVERY FRAME
DECOMPRESS THE ACCESSED DISPLAY DATA
S4
OUTPUT DECOMPRESSED DISPLAY DATA
S5

Fig. 6

DESCRIPTION

DISPLAY CONTROL DEVICE

This invention relates to a display control device and to a method of operating the display control device.

In desktop computing, it is now common to use more than one display device. Traditionally, a user would have a computer with a single display device attached, but now it is possible to have more than one display device attached to the computer, which increases the usable area for the worker. For example, International Patent Application Publication WO 2007/020408 discloses a display system which comprises a plurality of display devices, each displaying respectively an image, a data processing device connected to each display device and controlling the image displayed by each display device, and a user interface device connected to the data processing device. Connecting multiple display devices to a computer is a proven method for improving productivity.

The connection of an additional display device to a computer presents a number of problems. In general, a computer will be provided with only one video output such as a VGA-out connection. One method by which a second display device can be added to a computer is by adding an additional graphics card to the internal components of the computer. The additional graphics card will provide an additional video output which will allow the second display device to be connected to the computer and driven by that computer. However, this solution is relatively expensive and is not suitable for many non-technical users of computers.

An alternative method of connecting the second display device is to connect the additional display device to a USB socket on the computer, as all modern computers are provided with multiple USB sockets. This provides a simple connection topology, but requires additional hardware and software to be present, as in general, USB has a bandwidth that makes the provision of a

good quality video output a non-trivial task. It is also desirable that any additional hardware between the computer and the display device is kept as simple as possible. This means that when connecting an additional display device using a limited bandwidth technology such as USB, certain complex tasks can be difficult to achieve. In particular, a display control device needs to present that can receive the USB input and provide a suitable output to the display device.

As display technologies improve and the user's desire for display quality increases, the requirement placed upon the display control device that is receiving the incoming display data will correspondingly increase, as the amount of display data will increase proportionally. Since the whole point of the display control device is to provide a cost effective method of adding another display device, it is important that the costs of this device is kept to a minimum. It is obviously desirable that any new display control devices can handle higher definition video inputs and higher resolution display devices, so the ability to produce a high performance display control device at as lowa cost as possible is highly desirable.

It is therefore an object of the invention to improve upon the known art.

According to a first aspect of the present invention, there is provided a method of operating a display control device connected to receive compressed display data transported in a transport protocol via a general purpose data network from a source and to output decompressed display data in a display-specific data standard to a display device, the method comprising the steps of:

- receiving compressed display data from the source of the compressed display data via the general purpose data network,
- unpackaging the received compressed display data from the transport protocol in which it is transported,
- storing the unpackaged received compressed display data as it is received without first decompressing the received compressed display data, and
- for each frame refresh of the display device:
    - accessing the stored compressed display data,
    - decompressing the accessed display data, and
    - outputting the decompressed display data in the display-specific data standard to the display device.

According to a second aspect of the present invention, there is provided a display control device configured to receive compressed display data transported

28 01 14

in a transport protocol via a general purpose data network from a source and to output decompressed display data in a display-specific data standard to a display device, the display control device comprising a frame buffer store, and one or more components arranged to perform the method described above.

Owing to the invention, it is possible to provide a display control device that will have a reduced memory requirement and therefore a reduced cost with respect to a display control device of the same capability using an uncompressed memory buffer. Although it might seem counter-intuitive to decompress the stored display data every frame refresh, rather than store the display data in an uncompressed form as is conventional, the increase in complexity in the decompression unit of the display control device is far outweighed by the savings in memory. In particular, in many high-definition display systems, the need for an off-chip memory is removed, and this realises significant power and operational savings. All of the compressed display data can be stored on chip, without the need for a separate RAM for the display buffer.

Preferably, the transport protocol is either USB or Ethernet and the display-specific data standard is either VGA or DVI. This provides a simplified solution to the connection topology and to the process of receiving the display data and writing it into memory.

Advantageously, the step of storing the received compressed display data comprises storing compressed display data for two consecutive frames. To improve the working of the display control device with respect to its operation in a high-definition environment, it is advantageous to delay the display by a single frame and store two entire frames (in their compressed format) in the frame buffer. It is desirable to have enough storage for two or even three complete frames so as to avoid the visible artefact known as tearing. The update of the physical display panel at a refresh rate (say, 60Hz) is not synchronised with the sending of new data to the display. If there is only exactly enough memory for the display, and the device is only half-way

28 01 14

through updating the memory, then the refresh hardware may have no choice but to display half of the old picture and half of the new. If this happens repeatedly in a video then it is highly noticeable (and annoying) to the human viewer.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:-

Figure 1 is a schematic diagram of a display system,

Figure 2 is a schematic diagram of components of the display system,

Figure 3 is a schematic diagram of a display control device of the display system,

Figure 4 is a schematic diagram of a compressed frame buffer of the display control device,

Figure 5 is a schematic diagram of a decompression-video bridge of the display control device, and

Figure 6 is a flowchart of a method of operating the display control device.

A display system is shown in Figure 1. The system comprises a processing device 10, display devices 12 and user interface devices 14. The user interface devices are a keyboard 14a and a mouse 14b. The system shown in Figure 1 is a standard desktop computer, with an additional display device 12b, which is composed of discrete components that are locally located but could equally be a device such as a laptop computer or suitably enabled handheld device such as a mobile phone or pda (personal digital assistant) all using an additional display 12b. Similarly, the system may comprise part of a networked or mainframe computing system, in which case the processing device 10 may be located remotely from the user input devices 14 and the display devices 12, or indeed may have its function distributed amongst separate devices.

The display devices 12 show images 16, and the display of the images 16 is controlled by the processing device 10. One or more applications are

running on the processing device 10 and these are represented to the user by corresponding application windows 18, with which the user can interact in a conventional manner. A cursor 20 is shown, and the user can control the movement of the cursor 20 about the images 16 shown on the display device 12 using the computer mouse 14b, again in a totally conventional manner. The user can perform actions with respect to any running application via the user interface device 14 and these actions result in corresponding changes in the images 16, displayed by the display device 12.

The operating system run by the processing device 10 uses virtual desktops to manage the multiple display devices 12. Each physical display device 12 is represented by a frame buffer that contains everything currently shown on that display device 12. The operating system is configured to arrange these frame buffers into a single virtual desktop. When these frame buffers are arranged in the virtual desktop 22 in the same relative positions in which the physical display devices 12 are relatively placed, then the operating system can draw objects on all the display devices 12 in a natural way. The virtual desktop is a combination of the respective images 16a and 16b being shown by the display devices 12. If the user moves the mouse 14a such that the cursor 20 moves right off the edge of one display device 12a, then the cursor 20 appears on the left of the display device 12b to the right. Similarly a window 18 spread across several display devices 12 appears properly lined up between the display devices 12.

More detail of the connection of the secondary display device 12b to the processing device 10 is shown in Figure 2. The processing device 10 connects to the secondary display device 12b via a display control device 22. The display control device 22 is connected to the processing device 10 via a standard USB connection, and appears to the processing device 10 as a USB connected device. Any communications between the processing device 10 and the display control device 22 are carried out under the control of a USB driver specifically for the display control device 22. Such devices allow the connection of the secondary display device 12b to the processing device 10 without the need for any hardware changes to the processing device 10.

The display control device 22 connects to the display device 12b via a standard VGA connection, and the display device 12b is a conventional display device 12 which requires no adjustment to operate in the display system shown in Figure 2. As far as the display device 12b is concerned, it could be connected directly to the graphics card of a processing device; it is unaware that the graphical data displayed by the display device 12b has actually been first sent via a USB connection to an intermediate component, the display control device 22. Multiple additional display devices 12 can be connected to the processing device 10 in this way, as long as suitable USB slots are available on the processing device 10.

The display control device 22 is external to the processing device 10 and is not a graphics card. It is a dedicated piece of hardware that receives graphical data via the USB connection from the processing device 10 and transforms that graphics data into a VGA format that will be understood by the display device 12b. In topological terms USB and VGA are only examples of data standards that can be used to connect the additional display device 12b to the processing device 10. The general principle is that a general-purpose data network (such as USB or Ethernet) connects the processing device 10 to the display control device 22 and a display-specific data standard (such as VGA or DVI) is used on the connection from the display control device 22 to the display device 12b.

More detail of the display control device 22 is shown in Figure 3. This Figure shows the flow of display data in the presence of a compressed frame buffer 24. In broad terms within the display control device 22 there are three separate devices; a NIVO 23, a video controller 44 and a frame buffer store 28. NIVO stands for network in, video out and is essentially a video decoder that receives compressed video over the USB connection and turns the compressed video into pixel data that can be used to refresh the display device 12. The video controller 44 is performing the physical refresh of the display device 12.

There are two important components, the CFB 24 itself and a NVB 26 (a Nivo Video Bridge). The display data from the data processing device 10 is

stored in a frame buffer store 28 in compressed form, rather than being forwarded directly to the decompression components, as it arrives. No decompression is performed on the display data before it is stored. For each frame refresh of the display device 12, the display data is fetched by the CFB 24 and forwarded through the decompression components.

The decompression part of the display control device 22 comprises a series of components, being an input engine 30, an input buffer 32, VBD engines 34, an output buffer 36, xform engines 38 and a DMA engine 40, which connects to the NVB 26. The VBD engines 34 are entropy decoders, the xform engines 38 are performing tile transforms and the DMA engine 40 performs colour transforms. The original display data was compressed by being transformed, quantized and then entropy coded and these components of the NIVO 23 are effectively performing the reverse of these functions to decompress the display data back into pixel data.

The NVB 26 converts pixel tiles into pixel scan lines as required by video refresh, handles any timing differences between previous decode and video refresh requirements, and then passes the resulting pixels to an output buffer 42 of the video controller 44, as they are required. This stage may include some image improvement such as scaling or block edge filtering. The operation of the display control device 22 may give the impression that much more work is being done because the decode is performed for every refresh, but compared to a device that uses a large off-chip DRAM to store the uncompressed display data, a great deal of off-chip DRAM traffic is avoided and so the impact on power consumption is to reduce it rather than increase.

Figure 4 shows the internal components of the control component, the CFB 24 in more detail. As data arrives from the data processing device 10 it is received at an input process 46 and stored in the frame buffer store 28. This part of the display control device 22 can be embodied as an ASIC using on-chip 1T-SRAM for the frame buffer store 28.

The format of the display data sent by the data processing device 10 is controlled by a host device driver that is responsible for store management in the compressed frame store 28. Each atom of display data (such as a tile

group) sent from the host is prefixed with an address in the CFB store 28 at which the atom is to be saved. The host then sends a vector of pointers to the atoms, and a command to send those atoms for display. The vector of pointers is retained in the frame buffer store 28 along with the display data, so that if necessary the vector can be replayed for each refresh frame. For output one state machine reads the vector of pointers, and another state machine follows each pointer in turn to find an atom to be forwarded for display.

Figure 5 shows the internal components of the Nivo Video Bridge 26 in more detail. The exact processing to be performed may vary depending on the final device specification. The output of the DMA engine 40 provides pixels in tile group order, while the video controller 44 requires them in image scan order. The interleave block 52 converts between these. The interleave block 52 requires storage equal to <height of tile group> × <width of widest display mode> in pixels. For full high-definition, with 16x1 tile groups this comes to 48KB. No double buffering is required. The block edge filter 54 (if present) requires enough SRAM for a single scan-line of data (6KB for full high-definition). This is discussed in more detail below. The scaler 56 (if present) requires enough SRAM for four scan-lines of data (24KB for full high-definition). Again, this is discussed in more detail below.

The decompression/video clock domain crossing requires a small FIFO. The FIFO size may have to be increased for the worst-case lumpiness of NIVO output (in particular, to handle any gaps in NIVO output, for instance caused by PROC16 scheduling operations). The storage required are not significant. It may be advantageous to move the scaler 56 and/or the BEF 54 into the video clock domain 58. This would typically allow them to clock slower (using less power) while still meeting the requirements of the device. However, the ordering of interleave block 52, BEF 54 and scaler 56 must be maintained.

The block edge filter 54 is not an essential part of the design but can be used in order to get better picture quality at a chosen compression level. At around 4bpp the worst artefact that the compressed display data introduces is that in smoothly shaded areas of the image a faint grid of 4x4 pixels is visible. The provision of a very simple 2-pixel BEF 54 applied on a 4x4 grid removes

this effect. Likewise, the upscaler 56 is not an essential part of the design but would give more freedom when attaching computers or portable devices to displays.

By using a display control device 22 with a frame buffer store 28 that stores compressed display data rather than uncompressed display data it is possible to reduce system cost as there are no DDR chips, no DDR pins, a simpler/cheaper PCB, a reduced physical size, reduced system power consumption, all of which is ideal for low-end systems. When using four bit/pixel double-buffered, or eight bit/pixel for a static display, it is enough to provide extremely good image quality with a device size using 2Mbit of 1T-SRAM, which is about 0.71mm2 on 65nm. This is a hard macrocell including control logic. 2MBbyte would be 5.68mm2. This is enough to provide full high-definition 4bit/pixel for a double-buffered display or up to 8 bit/pixel for a static display.

There will be required some additional store for control data/code, 128KB is enough for either USB or Ethernet-based input. 1T-SRAM is faster than DDR and the bandwidth required by NIVO/Video is greatly reduced, so a smaller ARC cache is probably sufficient. The decompression part of the device becomes a little bigger in order to support this mode of operation. In order to turn tile groups into video scan-lines an SRAM containing <tile group height> × <max width of display in pixels> × <store for a pixel> is required. For full high-definition using 16x1 tile groups this comes to 48KB. The block edge filter requires another 6KB. The scaler requires another 24KB.

Figure 6 is a flowchart summarising the fundamental steps taken by the display control device 22 in delivering the display data for output. The method of operating the display control device 22 comprises, firstly step S1, which comprises receiving compressed display data, and secondly step S2, which comprises storing the received compressed display data. In the preferred embodiment, the received compressed display data comprises compressed frame data, each frame comprised of a plurality of rectangular tiles encoded with a wavelet transform. It is not material to the display control device 22 how

the display data has been compressed, it could any known public scheme or via a proprietary compression algorithm.

The method continues with steps S3 to S5, for each frame refresh of the display device 12. These steps are, step S3 accessing stored compressed display data, step S4 decompressing the accessed display data, and step S5 outputting the decompressed display data. For each frame refresh, not all of the compressed display data need be accessed. If only a portion of the usable display area has been updated, then only the relevant tiles stored within the frame buffer store 28 will be accessed, decompressed and outputted for display on the display device 12. Every frame, so twenty-five times per second (or whatever frame rate is being used), the frame buffer store 28 is accessed for the required compressed display data.

The step S2 of storing the received compressed display data preferably comprises storing the compressed display data without first decompressing the display data. The display control device 22 simply operates by receiving the compressed display data and writing that display data, in its compressed form, directly into the frame buffer store 28. No intermediate processing steps are carried out on the compressed display data. This display data is now available for decompression and display, every frame refresh of the display device 12. Preferably, a single chip within the display control device 22 carries out all of the receipt, storage and decompression of the received compressed display data, using on-chip SRAM for the storage of the compressed display data.

CLAIMS

1. A method of operating a display control device connected to receive compressed display data transported in a transport protocol via a general purpose data network from a source and to output decompressed display data in a display-specific data standard to a display device, the method comprising the steps of:

- receiving compressed display data from the source of the compressed display data via the general purpose data network,
- unpackaging the received compressed display data from the transport protocol in which it is transported,
- storing the unpackaged received compressed display data as it is received without first decompressing the received compressed display data, and
- for each frame refresh of the display device:
    - accessing the stored compressed display data,
    - decompressing the accessed display data, and
    - outputting the decompressed display data in the display-specific data standard to the display device.

28 01 14

2. A method according to claim 1, wherein the transport protocol is USB protocol.

3. A method according to claim 1, wherein the transport protocol is Ethernet protocol.

4. A method according to any preceding claim, wherein the step of storing the received compressed display data comprises storing compressed display data for two consecutive frames.

5. A method according to claim 4, wherein the decompressed display data is delayed by a single frame compared to the received compressed display data.

6. A method according to any preceding claim, wherein the received compressed display data comprises compressed frame data, each frame comprised of a plurality of rectangular tiles encoded with a wavelet transform.

7. A method according to any one of claims 1 to 5, wherein the received compressed display data comprises compressed frame data, each frame comprised of a plurality of rectangular tiles encoded with a discrete cosine transform.

8. A method according to any preceding claim, wherein the display-specific data standard is VGA.

9. A method according to any one of claims 1 to 7, wherein the display-specific data standard is DVI.

28 01 14

10. A display control device configured to receive compressed display data transported in a transport protocol via a general purpose data network from a source and to output decompressed display data in a display-specific data standard to a display device, the display control device comprising:

- a frame buffer store, and
- one or more components arranged to perform the method of any preceding claim.

11. A display system comprising a computer for outputting compressed display data in a transport protocol, a display device for receiving decompressed display data in a display-specific data standard, and displaying the decompressed display data on a screen, and a display control device according to claim 10 coupled via a general purpose data network to the computer for receiving the compressed display data in the transport protocol via the general purpose data network from the computer and coupled to the display device via a display-specific data standard connection for outputting

the decompressed display data to the display device in the display-specific data standard.

12. A method of operating a display control device substantially as hereinbefore described with reference to the drawings.

13. A display control device substantially as hereinbefore described with reference to the drawings.

14. A display system substantially as hereinbefore described with reference to the drawings.

28 01 14