Title: DATA PROCESSING APPARATUS FOR TRANSMITTING/RECEIVING COMPRESSED DISPLAY DATA WITH IMPROVED ERROR ROBUSTNESS AND RELATED DATA PROCESSING METHOD

(57) Abstract: A data processing apparatus has a compressor and an output interface. The compressor performs compression upon an input display data to generate a compressed display data, wherein an error-resilient coding tool is involved in the compression. The output interface packs an output display data derived from the compressed display data into an output bitstream, and outputs the output bitstream via a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA).

FIG. 10
before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

Published:

— with international search report (Art. 21(3))
DATA PROCESSING APPARATUS FOR
TRANSMITTING/RECEIVING COMPRESSED DISPLAY DATA
WITH IMPROVED ERROR ROBUSTNESS AND RELATED
DATA PROCESSING METHOD

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. provisional application No. 61/954,667, filed on 03/18/2014 and incorporated herein by reference.

BACKGROUND

[0002] The disclosed embodiments of the present invention relate to transmitting display data over a display interface, and more particularly, to a data processing apparatus for transmitting/receiving compressed display data with improved error robustness and a related data processing method.

[0003] A display interface is disposed between a first chip and a second chip to transmit display data from the first chip to the second chip for further processing. For example, the first chip may be a host application processor, and the second chip may be a driver integrated circuit (IC). The display data may include image data, video data, graphic data, and/or OSD (on-screen display) data. Besides, the display data may be single view data for two-dimensional (2D) display or multiple view data for three-dimensional (3D) display. When a display panel supports a higher display resolution, 2D/3D display with higher resolution can be realized. Hence, the display data transmitted over the display interface would have a larger data size/data rate, which increases the power consumption of the display interface inevitably. If the host application processor and the driver IC are both located at a portable device (e.g., a smartphone) powered by a battery device, the battery life is shortened due to the increased power consumption of the display interface. Thus, there is a need for an innovative design which can effectively reduce the power consumption of the display interface.
SUMMARY

[0004] In accordance with exemplary embodiments of the present invention, a data processing apparatus for transmitting/receiving compressed display data with improved error robustness and a related data processing method are proposed.

[0005] According to a first aspect of the present invention, an exemplary data processing apparatus is disclosed. The exemplary data processing apparatus includes a compressor and an output interface. The compressor is arranged to perform compression upon an input display data to generate a compressed display data, wherein an error-resilient coding tool is involved in the compression. The output interface is arranged to pack an output display data derived from the compressed display data into an output bitstream, and output the output bitstream via a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA).

[0006] According to a second aspect of the present invention, an exemplary data processing apparatus is disclosed. The exemplary data processing apparatus includes an input interface and a de-compressor. The input interface is arranged to receive an input bitstream from a display interface, and un-pack the input bitstream into an input display data, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA). The de-compressor is arranged to perform decompression upon a compressed display data derived from the input display data to generate a de-compressed display data, wherein an error detection and an error concealment are involved in the decompression.

[0007] According to a third aspect of the present invention, an exemplary data processing method is disclosed. The exemplary data processing method includes: utilizing a compressor to perform compression upon an input display data to generate a compressed display data, wherein an error-resilient coding tool is involved in the compression; packing an output display data derived from the compressed display data
into an output bitstream; and outputting the output bitstream via a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA).

According to a fourth aspect of the present invention, an exemplary data processing method is disclosed. The exemplary data processing method includes: receiving an input bitstream from a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA); un-packing the input bitstream into an input display data; and utilizing a de-compressor to perform decompression upon a compressed display data derived from the input display data and generate a de-compressed display data, wherein an error detection and an error concealment are involved in the decompression.

These and other objectives of the present invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment that is illustrated in the various figures and drawings.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0010] FIG. 1 is a block diagram illustrating a data processing system according to an embodiment of the present invention.

[0011] FIG. 2 is a diagram illustrating a display data compression without compression-unit interleaving and a display data compression with compression-unit interleaving according to an embodiment of the present invention.

[0012] FIG. 3 is a diagram illustrating data partitioning of a compressed data of a compression unit in a frame according to an embodiment of the present invention.

[0013] FIG. 4 is a diagram illustrating a display data compression without constrained coding reference for prediction and a display data compression with constrained coding reference for prediction according to an embodiment of the present invention.

[0014] FIG. 5 is a diagram illustrating a display data compression without
constrained coding reference for rate control and a display data compression with constrained coding reference for rate control according to an embodiment of the present invention.

[0015] FIG. 6 is a diagram illustrating re-synchronization marker insertion applied to the compressed display data according to an embodiment of the present invention.

[0016] FIG. 7 is a diagram illustrating error correction code insertion applied to the compressed display data according to an embodiment of the present invention.

[0017] FIG. 8 is a diagram illustrating a compressed data copy inserted to the compressed display data according to an embodiment of the present invention.

[0018] FIG. 9 is a diagram illustrating another compressed data copy inserted to the compressed display data according to an embodiment of the present invention.

[0019] FIG. 10 is a flowchart illustrating an error-robust compression method according to an embodiment of the present invention.

[0020] FIG. 11 is a diagram illustrating a first example of the processing circuit shown in FIG. 1.

[0021] FIG. 12 is a diagram illustrating a second example of the processing circuit shown in FIG. 1.

[0022] FIG. 13 is a diagram illustrating a first example of the spatial error concealment technique applied to a compression unit according to an embodiment of the present invention.

[0023] FIG. 14 is a diagram illustrating a second example of the spatial error concealment technique applied to a compression unit according to an embodiment of the present invention.

[0024] FIG. 15 is a diagram illustrating an example of the temporal error concealment technique applied to a compression unit according to an embodiment of the present invention.

[0025] FIG. 16 is a flowchart illustrating an error detection and concealment method according to an embodiment of the present invention.

**DETAILED DESCRIPTION**

[0026] Certain terms are used throughout the description and following claims to refer to particular components. As one skilled in the art will appreciate, manufacturers
may refer to a component by different names. This document does not intend to
distinguish between components that differ in name but not function. In the following
description and in the claims, the terms "include" and "comprise" are used in an
open-ended fashion, and thus should be interpreted to mean "include, but not limited
to ...". Also, the term "couple" is intended to mean either an indirect or direct electrical
collection. Accordingly, if one device is coupled to another device, that connection
may be through a direct electrical connection, or through an indirect electrical
connection via other devices and connections.

[0027] The concept of the present invention is to apply data compression upon
display data and then transmit compressed display data over a display interface. As the
data size/data rate of the compressed display data is smaller than that of the original
un-compressed display data, the power consumption of the display interface is reduced
correspondingly. However, bit error(s) may occur during data transmission between an
encoder side (e.g., an application processor) and a decoder side (e.g., a driver IC).
When data compression is applied to the display data, error pixels resulting from bit
error(s) introduced during compressed data transmission will be propagated and shown
on a display apparatus, where the size of an error propagation area is related to a
compression unit size. To improve error robustness of the display data compression,
the present invention further proposes adding at least one error-resilient coding tool to
a compressor and/or configuring a de-compressor to have an error detection and error
concealment capability. Further details will be described as below.

[0028] FIG. 1 is a block diagram illustrating a data processing system according
to an embodiment of the present invention. The data processing system 100 includes a
plurality of data processing apparatuses such as an application processor 102 and a
driver integrated circuit (IC) 104. The application processor 102 and the driver IC 104
may be implemented in different chips, and the application processor 102 communicates with the driver IC 104 via a display interface 103. In this embodiment,
the display interface 103 may be a display serial interface (DSI) standardized by a
Mobile Industry Processor Interface (MIPI) or an embedded display port (eDP)
standardized by a Video Electronics Standards Association (VESA).

[0029] The application processor 102 is coupled to the display interface 103, and
supports un-compressed data transmission and compressed data transmission. When
the application processor 102 is used to transmit un-compressed data to the driver IC
104, the application processor 102 generates an output display data D_OUT
according to a source display data D_IN1 provided by an external data source 105, where the output display data D_OUT1 is derived from un-compressed display data DI, and the un-compressed display data D1 is derived from the source display data D_IN1. When the application processor 102 is used to transmit compressed data to the driver IC 104, the application processor 102 generates the output display data D_OUT1 according to the source display data D_IN1 provided by the external data source 105, where the output display data D_OUT1 is derived from compressed display data DI’, the compressed display data DI’ is derived from an input display data D3, and the input display data D3 is derived from the un-compressed display data D1 or the source display data D_IN1. When the application processor 102 enables one of the un-compressed data transmission and the compressed data transmission, the application processor 102 further transmits the output display data D_OUT1 over the display interface 103. By way of example, but not limitation, the data source 105 may be a camera sensor, a memory card or a wireless receiver, and the source display data D_IN1 may include image data, video data, graphic data, and/or OSD data. Further, the source display data D_IN1 may be single view data for 2D display or multiple view data for 3D display.

[0030] As shown in FIG. 1, the application processor 102 includes a display controller 112, an output interface 114 and a processing circuit 116. The processing circuit 116 includes circuit elements required for processing the source display data D_IN1 to generate the output display data D_OUT1 (which may be un-compressed data for un-compressed data transmission over the display interface 103, or may be compressed data for compressed data transmission over the display interface 103). For example, the processing circuit 116 may have a compressor 117 and other circuitry 118, where the other circuitry 118 may have a display processor, a multiplexer, additional image/video processing element(s), etc. The display processor may perform image processing operations, including scaling, rotating, etc. For example, the display processor processes display data derived from the source display data D_IN1 to generate the un-compressed display data DI, where the source display data D_IN1 may be bypassed or processed by the additional image/video processing element(s) located before the display processor.

[0031] The compressor 117 is configured to perform display data compression. Hence, the compressor 117 performs compression upon the input display data D3 to generate the compressed display data DI’. In one exemplary design, the input display
data D3 may be the source display data D_INl provided by the data source 105. In another exemplary design, the input display data D3 may be the un-compressed display data D1 provided by the other circuitry 118.

[0032] The multiplexer of the other circuitry 118 receives the un-compressed display data D1 and the compressed display data D1', and selectively outputs the un-compressed display data D1 or the compressed display data D1' according to the operation mode of the application processor 102. For example, the display controller 112 controls the operation of the application processor 102. Hence, when the application processor 102 is operated under a compression mode, the multiplexer is controlled by the display controller 112 to output the compressed display data D1' to be the output display data D_OUTl; and when the application processor 102 is operated under a non-compression mode, the multiplexer is controlled by the display controller 112 to output the un-compressed display data D1 to be the output display data D_OUTl. As the present invention focuses on the error-robust display data compression, further description of the other circuit 118 is omitted here for brevity. The output interface 114 is arranged for packing/packetizing the output display data D_OUTl into an output bitstream according to the transmission protocol of the display interface 103, and transmits the output bitstream to the driver IC 104 via the display interface 103.

[0033] Concerning the error-robust display data compression, at least one error-resilient coding tool is involved in the compression performed by the compressor 117. For example, the compressor 117 may employ at least one of the proposed error-resilient coding tools, including compression-unit interleaving, data partitioning, constrained coding reference and redundancy information insertion, to achieve the error-robust display data compression. Further details of the proposed error-resilient coding tools are described as below.

[0034] FIG. 2 is a diagram illustrating a display data compression without compression-unit interleaving and a display data compression with compression-unit interleaving according to an embodiment of the present invention. The input display data D3 fed into the compressor 117 includes a display data of a frame IMG. The size of the frame IMG is WxH. That is, the number of pixel columns in the frame IMG is W, and the number of pixel rows in the frame IMG is H. A compression unit is a minimum encoding segment that can be independently decoded. For example, the size of each compression unit is Wx8. The sub-diagram (A) of FIG. 2 illustrates the display
data compression without compression-unit interleaving. A region R in the frame IMG is divided into two compression units 201 and 202, each having 8 consecutive pixel rows. Consider a case where bit errors are introduced to the compressed data of a specific compression unit during compressed data transmission over the display interface 103. When the compressed data with error bits is decoded at the driver IC 104, error propagation may occur in a continuous display area corresponding to the corrupted specific compression unit.

[0035] To mitigate the image quality degradation caused by the error propagation, the compressor 117 can be configured to perform the display data compression with compression-unit interleaving, as illustrated in sub-diagram (B) of FIG. 2. The compressor 117 uses compression-unit interleaving as one error-resilient coding tool. Hence, the compressor 117 partitions the region R in the frame IMG into a plurality of sub-regions S01-S08 and S11-S18. In this example, the size of each sub-region is 8x8. The compressor 117 selects interleaved sub-regions from the sub-regions S01-S08 and S11-S18 in the region R of the frame IMG to form one compression unit with the size of Wx8. Hence, one compression unit 203 is composed of sub-regions S01, S12, S03, S14, S05, S16, S07, and S18; and another compression unit 204 is composed of sub-regions S11, S02, S13, S04, S15, S06, S17, and S08. Consider a case where bit errors are introduced to the compressed data of a specific compression unit during compressed data transmission over the display interface 103. When the compressed data with error bits is decoded at the driver IC 104, error propagation is constrained in a discontinuous display area corresponding to the corrupted specific compression unit composed of interleaved sub-regions.

[0036] FIG. 3 is a diagram illustrating data partitioning of a compressed data of a compression unit in a frame according to an embodiment of the present invention. For example, a compression unit shown in sub-diagram (A) of FIG. 2 is compressed by the compressor 117 using data partitioning as one error-resilient coding tool. The compressor 117 is configured to partition and organize the compressed data of the compression unit into a plurality of data segments. The compressed data of the compression unit may have two parts, including a compression header and a compression payload. The data segments obtained by the data partitioning may include the compression header, sub-stream headers of the compression payload, and sub-stream payloads of the compression payload. For example, the compression header of the compression unit may include a picture parameter set (PPS) in VESA's
display stream compression. A sub-stream header may include a parameter K in Golomb-Rice coding. A sub-stream payload may include prefix and suffix parts in Golomb-Rice coding. In addition, the output interface 114 may add a transport header when packing the compressed display data D1’ into the output bitstream.

[0037] The data segments derived from partitioning and organizing the compressed data of the compression unit may include at least a first data segment with a first priority and a second data segment with a second priority. The compressor 117 configures error detection and correction capability of the first data segment and the second data segment based on the priority order of the first priority and the second priority. For example, the priority of the transport header of the output bitstream is higher than the priority of the compression header of the compression unit, the priority of the compression header of the compression unit is higher than the priority of the sub-stream header in the compression payload of the compression unit, and the priority of the sub-stream header in the compression payload of the compression unit is higher than the priority of the sub-stream payload in the compression payload of the compression unit. The compressor 115 configures an error correction code or a cyclic redundancy check (CRC) checksum to improve the error resilience, where the error detection and correction capability depends on the priority order. For example, the compressor 117 generates an error correction code for a data segment with a higher priority, but does not generate an error correction code for a data segment with a lower priority. For another example, the compressor 117 generates an error correction code with stronger error detection and correction capability for a data segment with a higher priority, and generates an error correction code with weaker error detection and correction capability for a data segment with a lower priority. However, these are for illustrative purposes only, and are not meant to be limitations of the present invention.

[0038] FIG. 4 is a diagram illustrating a display data compression without constrained coding reference for prediction and a display data compression with constrained coding reference for prediction according to an embodiment of the present invention. As mentioned above, a compression unit is a minimum encoding segment that can be independently decoded. In addition, one compression unit may be composed of a plurality of coding units each having a plurality of pixels. For example, each of the compression units 201 and 202 shown in sub-diagram (A) of FIG. 2 includes a plurality of coding units each having three pixels. As illustrated in sub-diagram (A) of FIG. 4, one coding unit CUIO in the compression unit 201 is
composed of pixels P0, P1, and P2 of the compression unit 201; and another coding
unit CU11 in the compression unit 201 is composed of pixels P3, P4, and P5 of the
compression unit 201. Further, one coding unit CU20 in the compression unit 202 is
composed of pixels P0, P1, and P2 of the compression unit 202; and another coding
unit CU21 in the compression unit 202 is composed of pixels P3, P4, and P5 of the
compression unit 202. When there is no constrained coding reference for prediction as
illustrated in sub-diagram (A) of FIG. 4, encoding of the coding unit CU11 in the
compression unit 201 may employ an intra compression-unit prediction provided by
the pixel data of the coding unit CU10 in the same compression unit 201, and/or
encoding of the coding unit CU20 in the compression unit 202 may employ an inter
coding-unit prediction provided by the pixel data of the coding unit CU10 in the
compression unit 201. Consider a case where bit errors are introduced to the
compressed data of the compression unit 201 (particularly, the compressed data of the
coding unit CU10) during compressed data transmission over the display interface 103.
When the compressed data with error bits is decoded at the driver IC 104, error
propagation may occur in a continuous display area corresponding to the compression
unit 201 (or both compression units 201 and 202) due to prediction dependency.

[0039] To mitigate the image quality degradation caused by the error propagation,
the compressor 117 can be configured to perform the display data compression with
constrained coding reference for prediction, as illustrated in sub-diagram (B) of FIG. 4.
Hence, the compressor 117 blocks an intra compression-unit prediction provided by
the coding unit CU10 in the compression unit 201 from being used for encoding the
coding unit CU11 in the same compression unit 201, and/or blocks an inter
compression-unit prediction provided by the coding unit CU10 in the compression unit
201 from being used for encoding the coding unit CU20 in the different compression
unit 202.

[0040] FIG. 5 is a diagram illustrating a display data compression without
constrained coding reference for rate control and a display data compression with
constrained coding reference for rate control according to an embodiment of the
present invention. As mentioned above, a compression unit is a minimum encoding
segment that can be independently decoded; in addition, one compression unit may be
composed of a plurality of coding units each having a plurality of pixels. By way of
example, but not limitation, each compression unit in the same frame may be assigned
the same target bit budget. When there is no constrained coding reference for rate
control as illustrated in sub-diagram (A) of FIG. 5, a bit budget allocated to one coding unit may be dynamically adjusted by an intra compression-unit rate control or an inter compression-unit rate control. For example, a bit budget allocated to the coding unit CU1 in the compression unit 201 may be set based at least partly on a remaining bit budget left from encoding of the coding unit CUIO in the same compression unit 201, and/or a bit budget allocated to the coding unit CU20 in the compression unit 202 may be set based at least partly on a remaining bit budget left from encoding of the coding unit CUIO in the compression unit 201. Consider a case where bit errors are introduced to the compressed data of the compression unit 201 (particularly, the compressed data of the coding unit CUIO) during compressed data transmission over the display interface 103. When the compressed data with error bits is decoded at the driver IC 104, error propagation may occur in a continuous display area of the compression unit 201 (or both compression units 201 and 202) due to rate control dependency.

[0041] To mitigate the image quality degradation caused by the error propagation, the compressor 117 can be configured to perform the display data compression with constrained coding reference for rate control, as illustrated in sub-diagram (B) of FIG. 5. Hence, the compressor 117 blocks a remaining bit budget left from encoding of the coding unit CUIO in the compression unit 201 from being used for configuring a bit budget allocated to the coding unit CU1 in the same compression unit 201, and/or blocks a remaining bit budget left from encoding of the coding unit CUIO in the compression unit 201 from being used for configuring a bit budget allocated to the coding unit CU20 in the different compression unit 202.

[0042] When the error-resilient coding tool involved in the compression is the redundant information insertion, the redundant information may include a re-synchronization marker, an error correction code, and/or a copy of at least a portion of the compressed display data. FIG. 6 is a diagram illustrating re-synchronization marker insertion applied to the compressed display data according to an embodiment of the present invention. As mentioned above, the compression output of one compression unit in a frame may include a compression header and a compression payload. It should be noted that the re-synchronization marker may be a unique codeword different from all possible payload codewords and all possible header syntax patterns that may be transmitted over the display interface 103. In other words, the re-synchronization marker is uniquely identifiable in the compressed display data.
Dl' generated from compressing the input display data D3. The bit errors will not be propagated when encountering re-synchronization markers inserted in the compressed display data. For example, when a decoder side (e.g., driver IC 104) detects an error in the bitstream received from the display interface 103, it may discard bits until a re-synchronization marker is detected. When re-synchronization markers are inserted at proper locations, errors will be localized to small spatial regions in a frame.

[0043] FIG. 7 is a diagram illustrating error correction code insertion applied to the compressed display data according to an embodiment of the present invention. The compressor 117 calculates at least one error correction code (e.g., at least one CRC checksum) according to the compressed display data Dl', and adds the at least one error correction code (e.g., at least one CRC checksum) to the compressed display data Dl'. In the example shown in FIG. 7, one error correction code (e.g., one CRC checksum) is inserted after the compressed display data Dl', thereby improving the error detection and correction capability in the decoder side (e.g., driver IC 104).

[0044] FIG. 8 is a diagram illustrating a compressed data copy inserted to the compressed display data according to an embodiment of the present invention. The compressor 117 generates at least one copy of at least a portion (i.e., part or all) of the compressed display data Dl' to the output interface 114 for transmission. As mentioned above, the compression output of one compression unit in a frame may include a compression header 801 and a compression payload 803. In the example shown in FIG. 8, the compressor 117 copies the compression header 801 to generate a redundant compression header 802 to the output interface 114, such that the bitstream data of one compression unit will have two compression headers 801 and 802. When one compression header (e.g., 801) is corrupted during the compressed data transmission over the display interface 103, the other compression header (e.g., 802) can be used in the decoder side (e.g., driver IC 104).

[0045] FIG. 9 is a diagram illustrating another compressed data copy inserted to the compressed display data according to an embodiment of the present invention. The compressor 117 generates at least one copy of at least a portion (i.e., part or all) of the compressed display data Dl' to the output interface 114 for transmission. As mentioned above, the compression output of one compression unit in a frame may include a compression header 903 and a compression payload 901. In the example shown in FIG. 9, the compressor 117 copies a portion of the compression payload 901 to generate a redundant compression payload 902 to the output interface 114, such that
the bitstream data of one compression unit will have a full compression payload 901 and a partial compression payload 902. When a portion of the full compression payload 901 is corrupted during the compressed data transmission over the display interface 103, the additional partial compression payload 902 can be used in the decoder side (e.g., driver IC 104).

[0046] FIG. 10 is a flowchart illustrating an error-robust compression method according to an embodiment of the present invention. The error-robust compression method may be employed by the application processor 102 having the compressor 117. In step 1002, the display controller 112 of the application processor 102 queries the driver IC 104 to determine if there are bit error(s) detected by the driver IC 104. For example, the bit error(s) may occur during compressed data transmission over the display interface 103. In step 1004, the display controller 112 of the application processor 102 further queries the driver IC 104 to know the de-compression capability of the driver IC 104. When there are bit errors occurring during compressed data transmission over the display interface 103, the compressor 117 can selectively enable the error-robust compression based on the de-compression capability of the driver IC 104. When the de-compression capability of the driver IC 104 indicates that the compression-unit interleaving is supported, the compressor 117 may enable the compression-unit interleaving while performing compression upon the input display data D3 (Step 1006). When the de-compression capability of the driver IC 104 indicates that the data partitioning is supported, the compressor 117 may enable the data partitioning while performing compression upon the input display data D3 (Step 1008). When the de-compression capability of the driver IC 104 indicates that the constrained coding reference is supported, the compressor 117 may enable the constrained coding reference while performing compression upon the input display data D3 (Step 1010). When the de-compression capability of the driver IC 104 indicates that the redundant information insertion is supported, the compressor 117 may enable the redundant information insertion while performing compression upon the input display data D3 (Step 1012). In step 1014, the compressor 117 performs compression upon the input display data D3 to generate the compressed display data D1', and transmits the compressed display data D1' to the display interface 103 through the output interface 114. It should be noted that step 1002 is optional. That is, step 1002 may be omitted, depending upon actual design consideration.

[0047] Please refer to FIG. 1 again. Regarding the driver IC 104, it communicates
with the application processor 102 via the display interface 103. In this embodiment, the driver IC 104 is coupled to the display interface 103, and supports un-compressed data reception and compressed data reception. When the application processor 102 transmits the output display data D_OUT1 (which is derived from the un-compressed data D1) to the driver IC 104, the driver IC 104 is operated under a non-decompression mode to obtain an un-compressed data D2 and drive a display panel 106 according to the un-compressed display data D2. By way of example, the display panel 106 may be implemented using any 2D/3D display device. When the application processor 102 transmits the output display data D_OUT1 (which is derived from the compressed data D1') to the driver IC 104, the driver IC 104 is operated under a de-compression mode to obtain a de-compressed display data D4 and drive the display panel 106 according to the de-compressed display data D4.

[0048] As shown in FIG. 1, the driver IC 104 includes a driver IC controller 122, an input interface 124 and a processing circuit 126. The input interface 124 is arranged for receiving an input bitstream from the display interface 103, and un-packing/un-packetizing the input bitstream into an input display data D_IN2 according to the transmission protocol of the display interface 103. The processing circuit 126 may include circuit elements required for driving the display panel 106 according to a video mode or an image/command mode. For example, the processing circuit 126 may have a de-compressor 127 and other circuitry 128, and the other circuitry 128 may have a display buffer, multiplexers, etc. The de-compressor 127 is used for performing de-compression upon a compressed display data D2' derived from the input display data D_IN2. The display buffer is arranged for storing a display data to provide a buffered display data under the image/command mode, wherein the display data stored into the display buffer may be an un-compressed display data, a compressed display data or a de-compressed display data, depending upon actual design consideration/requirement. The multiplexers control interconnections of the de-compressor 127, the display buffer and the display panel 106. As the present invention focuses on the error detection and error concealment performed by the de-compressor 127, further description of the other circuitry 128 is omitted here for brevity.

[0049] The de-compressor 127 performs decompression upon the compressed display data D2' derived from the input display data D_IN2 to generate the de-compressed display data D4, where an error detection and an error concealment are
involved in the decompression. The de-compressor 127 may detect bit error(s) in the
compressed display data D2' through a bitstream-level detection, or detect bit error(s)
in the de-compressed display data D4 through a pixel-level detection.

[0050] In one exemplary design of the bitstream-level detection, the
de-compressor 127 detects error(s) in the compressed display data D2' by checking at
least one error correction code (e.g., at least one CRC checksum) of the compressed
display data D2'. For example, as shown in FIG. 7, an error correction code at the end
of a compressed display data transmitted from the application processor 102 to the
driver IC 104 may be used by the de-compressor 127 to verify correctness of the
received compressed display data. In another exemplary design of the bitstream-level
detection, the de-compressor 127 detects error(s) in the compressed display data D2'
by checking occurrence of any illegal syntax (e.g., illegal codeword) in the
compressed display data D2'.

[0051] Concerning the pixel-level detection, the de-compressor 127 detects
error(s) in the de-compressed display data D4 by checking smoothness of at least one
boundary, wherein the at least one boundary is between two de-compressed
compression units, or is between two de-compressed coding units in the same
compression unit. When a compressed compression unit (or a compressed coding unit)
has bit error(s), the error propagation will degrade the image quality of a

25 corresponding de-compressed compression unit (or de-compressed coding unit). The
boundary between a corrupted compression unit (or corrupted coding unit) and a
neighboring compression unit (or neighboring coding unit) will have unnatural image
characteristics. Hence, when a smoothness level at a boundary between two
de-compressed compression units (or two de-compressed coding units in the same
compression unit) is lower than a threshold, the de-compressor 127 may determine

30 that at least one of the de-compressed compression units (or de-compressed coding
units) has error(s). When the smoothness level at the boundary between two
de-compressed compression units (or two de-compressed coding units in the same
compression unit) is not lower than the threshold, the de-compressor 127 may
determine that both of the de-compressed compression units (or de-compressed coding
units) are error-free.

[0052] After the error detection performed by the de-compressor 127 detects
occurrence of error(s), the de-compressor 127 is operative to perform an error
concealment operation to conceal corrupted de-compressed compression units to avoid
image quality degradation.

[0053] FIG. 11 is a diagram illustrating a first example of the processing circuit 126 shown in FIG. 1. In this example, the other circuitry 128 may have a switch 1102, a display buffer 1104, and a multiplexer (MUX) 1106. When the driver IC 104 is operated in the video mode for driving the display panel 106, the compressed display data D2' is not stored into the display buffer 1104, and is directly de-compressed by the de-compressor 127 to generate the de-compressed display data D4 to a first input port PI of the MUX 1106, and the MUX 1106 outputs the de-compressed display data D4 received at the first input port PI to the display panel 106. 

[0054] When the driver IC 104 is operated in the image/command mode to drive the display panel 106, the compressed display data D2' is stored into the display buffer 1104, the de-compressor 127 de-compresses the compressed display data D2' read from the display buffer 1104 to generate the de-compressed display data D4 to a second input port P2 of the MUX 1106, and the MUX 1106 outputs the de-compressed display data D4 received at the second input port P2 to the display panel 106. In addition, the switch 1102 is controlled by the error detection performed by the de-compressor 127. For example, when the de-compressor 127 detects error(s) during processing of a current frame, the de-compressor 127 may control the switch 1102 to disconnect the display buffer 1104 from a compressed data input, such that the remaining compressed data of the current frame is not stored into the display buffer 1104. When the compressed data of a next frame (e.g., re-transmitted compressed data of the current frame) is available at the compressed data input, the de-compressor 127 may control the switch 1102 to connect the display buffer 1104 to the compressed data input, thereby allowing the compressed data of the next frame (e.g., re-transmitted compressed data of the current frame) to overwrite data of the corrupted frame in the display buffer 1104. However, this is for illustrative purposes only, and is not meant to be a limitation of the present invention. The switch 1102 shown in FIG. 11 may be omitted, depending upon actual design consideration. 

[0055] FIG. 12 is a diagram illustrating a second example of the processing circuit 126 shown in FIG. 1. The major difference between the example shown in FIG. 11 and the example shown in FIG. 12 is the location of the display buffer 1104. In the example shown in FIG. 12, the display buffer 1104 may serve as a frame buffer to store the de-compressed display data D4 generated from the de-compressor 127 in the image/command mode. Specifically, when the driver IC 104 is operated in the
image/command mode to drive the display panel 106, the compressed display data D2' is fed into the de-compressor 127, the de-compressor 127 generates the de-compressed display data D4 to the display buffer 1104, and the MUX 1106 outputs the de-compressed display data D4 read from the display buffer 1104 to the display panel 106. In addition, the switch 1102 is controlled by the error detection performed by the de-compressor 127. For example, when the de-compressor 127 detects error(s) during processing of a current frame, the de-compressor 127 may control the switch 1102 to disconnect the de-compressor 127 from a compressed data input, such that the remaining compressed data of the current frame is not transmitted to the de-compressor 127. When the compressed data of a next frame (e.g., re-transmitted compressed data of the current frame) is available at the compressed data input, the de-compressor 127 may control the switch 1102 to connect the de-compressor 127 to the compressed data input, thereby allowing the compressed data of the next frame (e.g., re-transmitted compressed data of the current frame) to be fed into the de-compressor 127. However, this is for illustrative purposes only, and is not meant to be a limitation of the present invention. The switch 1102 shown in FIG. 12 may be omitted, depending upon actual design consideration.

[0056] Regarding the examples shown in FIG. 11 and FIG. 12, the de-compression 127 may be configured to employ a spatial error concealment technique or a temporal error concealment technique to conceal a corrupted de-compressed compression unit to avoid image quality degradation. Further details of the spatial error concealment technique and the temporal error concealment technique applied to a compression unit are described as below.

[0057] FIG. 13 is a diagram illustrating a first example of the spatial error concealment technique applied to a compression unit according to an embodiment of the present invention. One frame FN has a plurality of compression units (e.g., 1301 and 1302) included therein. In this example, the error detection performed by the de-compressor 127 indicates that the compression unit 1301 is error-free, and further indicates that the compression unit 1302 is corrupted due to bit error(s) detected. The spatial error concealment employed by the de-compressor 127 conceals the corrupted compression unit 1302 by a spatially-neighboring error-free compression unit (e.g., error-free compression unit 1301) in the same frame FN.

[0058] FIG. 14 is a diagram illustrating a second example of the spatial error concealment technique applied to a compression unit according to an embodiment of
the present invention. One frame FN has a plurality of compression units (e.g., 1401 and 1402) included therein. In this example, compression-unit interleaving is employed by the compressor 117 at the application processor 102. Hence, each of the compression units 1401 and 1402 is composed of interleaved sub-regions. The error detection performed by the de-compressor 127 indicates that the compression unit 1401 is error-free, and further indicates that the compression unit 1402 is corrupted due to bit error(s) detected. The spatial error concealment employed by the de-compressor 127 conceals the corrupted compression unit 1402 by an interpolated compression unit, wherein the interpolated compression unit is derived from interpolating one or more spatially-neighboring error-free compression units in the same frame FN. For example, the interpolated compression unit is generated by interpolating interleaved sub-regions included in the error-free compression unit 1401.

[0059] FIG. 15 is a diagram illustrating an example of the temporal error concealment technique applied to a compression unit according to an embodiment of the present invention. There are successive frames (e.g., FN-1 and FN) in the time domain, where the frame FN-1 has a plurality of compression units (e.g., 1501 and 1502) included therein, and the frame FN has a plurality of compression units (e.g., 1503 and 1504) included therein. It should be noted that the compression units 1501 and 1503 are co-located in different frames FN-1 and FN, and the compression units 1502 and 1504 are co-located in different frames FN-1 and FN. In this example, the error detection performed by the de-compressor 127 indicates that the compression unit 1501 in the previous frame FN-1 is error-free, and further indicates that the compression unit 1503 in the current frame FN is corrupted due to bit error(s) detected. The temporal error concealment employed by the de-compressor 127 conceals the corrupted compression unit 1503 in the current frame FN by a temporally-neighboring error-free compression unit (e.g., the co-located error-free compression unit 1501) in the previous frame FN-1.

[0060] FIG. 16 is a flowchart illustrating an error detection and concealment method according to an embodiment of the present invention. The error detection and concealment method may be employed by the driver IC 104 having the de-compressor 127. In step 1602, the driver IC 104 receives an input bitstream from the display interface 103. In step 1604, the de-compressor 127 performs the error detection to check if there are error(s) detected in the compressed display data D27de-compressed display data D4. If there is no error detected, the flow proceeds with step 1614.
However, if there are error(s) detected, the flow proceeds with step 1606. In step 1606, a display driving mode of the driver IC 104 is checked. If the driver IC 104 is operated in a video mode to drive the display panel 106, the flow proceeds with step 1608. If the driver IC 104 is operated in an image/command mode to drive the display panel 106, the flow proceeds with one of steps 1610 and 1612. In each of steps 1608 and 1610, the de-compressor 127 employs a spatial error concealment technique to conceal any corrupted de-compressed compression unit. In one exemplary design, the de-compressor 127 has a line buffer. Hence, the de-compressor 127 can use the line buffer to achieve spatial error concealment under the video mode. In addition, the de-compressor 127 can use the display buffer to achieve spatial error concealment under the image/command mode. In step 1612, the de-compressor 127 employs a temporal error concealment technique to conceal any corrupted de-compressed compression unit. In one exemplary design, the de-compressor 127 can use the display buffer to achieve temporal error concealment under the image/command mode. In step 1614, the de-compressor 127 generates the de-compressed display data D4 to the display panel 106. Hence, the driver IC 104 drives the display panel 106 according to the de-compressed display data D4.

[0061] Those skilled in the art will readily observe that numerous modifications and alterations of the device and method may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.
CLAIMS

1. A data processing apparatus comprising:
   a compressor, arranged to perform compression upon an input display data to
   generate a compressed display data, wherein an error-resilient coding tool is involved in
   the compression; and
   an output interface, arranged to pack an output display data derived from the
   compressed display data into an output bitstream, and output the output bitstream via a
   display interface, wherein the display interface is one of a display serial interface (DSI)
   standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display
   port (eDP) standardized by a Video Electronics Standards Association (VESA).

2. The data processing apparatus of claim 1, wherein the input display data
   comprises a display data of a frame, and the error-resilient coding tool comprises:
   partitioning a region in the frame into a plurality of sub-regions; and
   selecting interleaved sub-regions from the sub-regions to form one compression unit.

3. The data processing apparatus of claim 1, wherein the compressed display data
   comprises a compression data of a compression unit in a frame, and the error-resilient
   coding tool comprises:
   partitioning and organizing the compressed data of the compression unit into a
   plurality of data segments, wherein the data segments include at least a first data segment
   with a first priority and a second data segment with a second priority; and
   configuring error detection and correction capability of the first data segment and
   the second data segment based on a priority order of the first priority and the second
   priority.

4. The data processing apparatus of claim 1, wherein the input display data
   comprises a display data of a frame, and the error-resilient coding tool comprises:
   blocking an intra compression-unit prediction provided by a first coding unit in a
   compression unit of the frame from being used for encoding a second coding unit in the
   compression unit, wherein the second coding unit is different from the first coding unit.

5. The data processing apparatus of claim 1, wherein the input display data
   comprises a display data of a frame, and the error-resilient coding tool comprises:
   blocking an inter compression-unit prediction provided by a coding unit in a first
   compression unit of the frame from being used for encoding a coding unit in a second
compression unit of the frame, wherein the first compression unit is different from the second compression unit.

6. The data processing apparatus of claim 1, wherein the input display data comprises a display data of a frame, and the error-resilient coding tool comprises:

- blocking a remaining bit budget left from encoding of a first coding unit in a compression unit of the frame from being used for configuring a bit budget allocated to a second coding unit in the compression unit, wherein the second coding unit is different from the first coding unit.

7. The data processing apparatus of claim 1, wherein the input display data comprises a display data of a frame, and the error-resilient coding tool comprises:

- blocking a remaining bit budget left from encoding of a coding unit in a first compression unit of the frame from being used for configuring a bit budget allocated to a coding unit in a second compression unit of the frame, wherein the first compression unit is different from the second compression unit.

8. The data processing apparatus of claim 1, wherein the error-resilient coding tool comprises:

- inserting at least one re-synchronization marker into the compressed display data, wherein the at least one re-synchronization marker is uniquely identifiable in the compressed display data.

9. The data processing apparatus of claim 1, wherein the error-resilient coding tool comprises:

- calculating at least one error correction code according to the compressed display data; and

- adding the at least one error correction code to the compressed display data.

10. The data processing apparatus of claim 1, wherein the error-resilient coding tool comprises:

- generating a copy of at least a portion of the compressed display data to the output interface.

11. A data processing apparatus comprising:

- an input interface, arranged to receive an input bitstream from a display interface, and un-pack the input bitstream into an input display data, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA); and
a de-compressor, arranged to perform decompression upon a compressed display data derived from the input display data to generate a de-compressed display data, wherein an error detection and an error concealment are involved in the decompression.

12. The data processing apparatus of claim 11, wherein the error detection comprises:

detecting error(s) in the compressed display data by checking at least one error correction code of the compressed display data.

13. The data processing apparatus of claim 11, wherein the error detection comprises:

detecting error(s) in the compressed display data by checking occurrence of at least one illegal syntax in the compressed display data.

14. The data processing apparatus of claim 11, wherein the error detection comprises:

detecting error(s) in the de-compressed display data by checking smoothness of at least one boundary, wherein the at least one boundary is between two de-compressed compression units, or is between two de-compressed coding units in a same compression unit.

15. The data processing apparatus of claim 11, wherein the de-compressed display data comprises a de-compressed data of a frame, and the error concealment comprises:

concealing a first de-compressed compression unit with error(s) by a second de-compressed compression unit, wherein the first de-compressed compression unit and the second de-compressed compression unit are in the frame.

16. The data processing apparatus of claim 11, wherein the de-compressed display data comprises a de-compressed data of a frame, and the error concealment comprises:

concealing a first de-compressed compression unit with error(s) by an interpolated compression unit, wherein the interpolated compression unit is derived from interpolating at least one second de-compressed compression unit, and the first de-compressed compression unit and the at least one second de-compressed compression unit are in the frame.

17. The data processing apparatus of claim 11, wherein the de-compressed display data comprises a de-compressed data of a first frame and a de-compressed data of a second frame, and the error concealment comprises:

concealing a first de-compressed compression unit with error(s) by a second de-compressed compression unit, wherein the first de-compressed compression unit is in
one of the first frame and the second frame, and the second de-compressed compression unit is in another of the first frame and the second frame.

18. A data processing method comprising:

utilizing a compressor to perform compression upon an input display data to generate a compressed display data, wherein an error-resilient coding tool is involved in the compression;

packing an output display data derived from the compressed display data into an output bitstream; and

outputting the output bitstream via a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA).

19. A data processing method comprising:

receiving an input bitstream from a display interface, wherein the display interface is one of a display serial interface (DSI) standardized by a Mobile Industry Processor Interface (MIPI) and an embedded display port (eDP) standardized by a Video Electronics Standards Association (VESA);

un-packing the input bitstream into an input display data; and

utilizing a de-compressor to perform decompression upon a compressed display data derived from the input display data to generate a de-compressed display data, wherein an error detection and an error concealment are involved in the decompression.
FIG. 2
Bitstream data of one compression unit

Transport header

Compressi on header

Compression payload

Sub-stream header

Sub-stream payload

Sub-stream header

Sub-stream payload

Sub-stream header

Sub-stream payload

Sub-stream header

Sub-stream payload

Priority order: \( 1 > 2 > 3 > 4 \)

FIG. 3
FIG. 4
FIG. 5
FIG. 6

FIG. 7
Bitstream data of one compression unit

801
Compression header

803
Compression payload

802
Compression header
(Reudndant)

FIG. 8

Bitstream data of one compression unit

901
Compression header

903
Compression payload

902
Compression payload
(Reudndant)

FIG. 9
Start

Query if there are bit error(s) detected by decoder (e.g., driver IC) 1002

Query de-compression capability of decoder 1004

Apply compression-unit interleaving if decoder can support 1006

Apply data partitioning if decoder can support 1008

Apply constrained coding reference if decoder can support 1010

Apply redundant information insertion if decoder can support 1012

Performs compression upon input display data to generate compressed display data, and transmit compressed display data over display interface 1014

End

FIG. 10
Start

Receive input bitstream from encoder (e.g., application processor) via display interface

Error(s) detected?

No

Video mode

Conceal corrupted compression unit by using spatial error concealment technique

Yes

Display driving mode?

Image/command mode

Conceal corrupted compression unit by using spatial error concealment technique

Conceal corrupted compression unit by using temporal error concealment technique

Generate de-compressed display data, and drive display panel according to de-compressed display data

End

FIG. 16
**INTERNATIONAL SEARCH REPORT**

**International application No.**
PCT/CN2015/074430

<table>
<thead>
<tr>
<th>A. CLASSIFICATION OF SUBJECT MATTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>H04N 7/12(2006.01)i; H04N 19/00(2014.01)i; G06F 13/00(2006.01)i</td>
</tr>
</tbody>
</table>

According to International Patent Classification (IPC) or to both national classification and IPC

<table>
<thead>
<tr>
<th>B. FIELDS SEARCHED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum documentation searched (classification system followed by classification symbols)</td>
</tr>
<tr>
<td>H04N G06F</td>
</tr>
</tbody>
</table>

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

- CNKI, CNPAT, WPI, EPODOC: output, display+, video, image, compress+, encod+, pack+, etc.; compress+, decod+, un?pack+, interface, error?, region?, area?, block?, MB, MIPI, DSI, eDP, VESA

<table>
<thead>
<tr>
<th>C. DOCUMENTS CONSIDERED TO BE RELEVANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category*</td>
</tr>
<tr>
<td>PX</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

- Special categories of cited documents:
  - “A” document defining the general state of the art which is not considered to be of particular relevance
  - “E” earlier application or patent but published on or after the international filing date
  - “Q” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  - “O” document referring to an oral disclosure, use, exhibition or other means
  - “D” document published prior to the international filing date but later than the priority date claimed

- “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- “&” document member of the same patent family

Date of the actual completion of the international search

17 July 2015

Date of mailing of the international search report

05 August 2015

Name and mailing address of the ISA/CN

STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China

Authorized officer

QLYing

Facsimile No. (86-10)62019451

Telephone No. (86-10)62413416

Form PCT/ISA/210 (second sheet) (July 2009)
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date (day/month/year)</th>
<th>Patent family member(s)</th>
<th>Publication date (day/month/year)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>US 2014098119 A1</td>
<td>10 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w o 2014056391 A1</td>
<td>17 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>u s 2014098110 A1</td>
<td>10 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w o 2014056375 A1</td>
<td>17 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>w o 2014056407 A1</td>
<td>17 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>u s 2014098893 A1</td>
<td>10 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>u s 2014098111 A1</td>
<td>10 April 2014</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104704551 A</td>
<td>10 June 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104704550 A</td>
<td>10 June 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104704533 A</td>
<td>10 June 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104704442 A</td>
<td>10 June 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104737224 A</td>
<td>24 June 2015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 104737223 A</td>
<td>24 June 2015</td>
</tr>
<tr>
<td></td>
<td>22 December 2011</td>
<td>WO 2011163138 A1</td>
<td>29 December 2011</td>
</tr>
<tr>
<td>US 2011310980 A1</td>
<td></td>
<td>CN 102681812 A</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 101110958 A</td>
<td>23 January 2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>u s 2008063078 A1</td>
<td>13 March 2008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2008028541 A</td>
<td>07 February 2008</td>
</tr>
</tbody>
</table>

Form PCT/ISA/210 (patent family annex) (July 2009)