IPOS copy
9906422-2
17 DEC 1999

MPEG VIDEO DECODER WITH INTEGRATED SCALING AND DISPLAY FUNCTIONS

Technical Field

The present invention is directed generally to digital video signal processing, and more particularly, to integrated decode systems, methods and articles of manufacture which allow selective scaling of video presentation by a predetermined reduction factor, while at the same time allowing for reduced external memory requirements for frame buffer storage.

Background of the Invention

The MPEG-2 standard describes an encoding method that results in substantial bandwidth reduction by a subjective lossy compression followed by a lossless compression. The encoded, compressed digital data is subsequently decompressed and decoded in an MPEG-2 compliant decoder. Video decoding in accordance with the MPEG-2 standard is described in detail in commonly assigned United States Letters Patent No. 5,576,765, entitled "Video Decoder", which is hereby incorporated herein by reference in its entirety.

Video decoders are typically embodied as general or special purpose processors and memory. For a conventional MPEG-2 decoder, two decoded reference frames are typically stored in memory at the same time. Thus, the cost of memory can often dominate the cost of the decode system. For example, an MPEG-2 video decoder might employ 2 MB or more of external memory, which generally comprises Dynamic Random

Access Memory (DRAM). External memory is used for various data areas, or buffers such as frame buffers.

In practice, the MPEG-2 video decoder is typically limited to 2 MB of external memory in order to minimize cost of the end product. The decoder must perform all of its functions within this limitation. For example, of particular importance is enabling output for both the European market which utilizes the PAL standard of 576 video scan lines and the U.S. market which utilizes the NTSC standard of 480 video scan lines. Even if there is no 2 MB of external memory limitation, it is advantageous to perform the video decode and display in as small a memory space as possible in order to give the remaining memory to other built-in features, such as on-screen graphics.

The MPEG-2 decompressed video data buffers, also called frame buffers, consume the largest part of external DRAM, therefore they are the prime candidate for memory reduction/compression. The frame buffers contain final pixel display and MPEG-reference data, and hence the reduction technique must also retain high video fidelity.

As the MPEG video decoder market becomes more and more competitive, there is a need for high level of feature integration at the lowest possible cost to achieve success in the marketplace. One such feature that, in the past, would have required circuitry external to the video decoder function is video scaling. The kind of scaling desired is to reduce the size of the display picture by a factor, such as 2 or 4, in both the horizontal and vertical axis.

In view of the above, and in order to establish commercial advantage, a novel design is desired wherein a

video scaling feature is built into the video decoder, such that advantageous use of existing decoder hardware can be applied to the processes required to produce a high quality scaled image. In one principal aspect, the present invention addresses this need.

Disclosure of the Invention

Briefly summarized, this invention comprises in one aspect a video decoding system which includes a video decoder for decoding an encoded stream of video data and a decimation unit coupled to the video decoder. The video decoder produces a decoded stream of video data and the decimation unit is adapted to scale the decoded stream of video data for display. The scaling occurs within the video decode system prior to storage of the decoded stream of video data in a frame buffer.

In another aspect, the invention comprises a digital video decoding system which includes a video decoder and a video scalar. The video decoder decodes an encoded stream of video data and produces therefrom a decoded stream of video data. The video scalar is coupled to the video decoder for scaling the decoded stream of video data prior to storage thereof in a frame buffer. The video decoding system includes a normal video mode and a scaled video mode. The video scalar scales the decoded stream of video data when the digital video decoding system is in the scaled video mode. The digital video decoding system further includes display mode switch logic for switching between the normal video mode and the scaled video mode, wherein the switching occurs without perceptual degradation of the display of the decoded stream of video data.

In yet another aspect, a digital video decoding system is provided having a normal video mode and a scaled video mode. When in the normal video mode, full size frames are output for display on a video display coupled to the digital video decoding system, and when in the scaled video mode, scaled frames comprising a fractional size of the full size frames are output for display on the video display. A frame buffer is provided for temporarily storing the full size frames and the scaled frames after a decoding time thereof and prior to a display time, wherein there is a predefined latency between the decoding time and the display time. The predefined latency between the decoding time and the display time comprises a first latency when the digital video decoding system is in normal video mode and a second latency when the digital video decoding system is in scaled video mode.

In still another aspect, a frame buffer is provided for a digital video decoding system having video scaling capabilities. The frame buffer includes multiple defined memory areas for receiving I, P & B frames of a decoded stream of video data. The multiple defined memory areas comprise a first area and a second area for receiving full size I and P frames of the decoded stream of video data, as well as at least one third area for receiving scaled I, P & B frames of the decoded stream of video data commensurate with the first area and the second area receiving the full size I and P frames.

In a further aspect, the invention comprises a frame buffer for a digital video decoding system having video scaling capabilities. The frame buffer includes memory associated with the digital video decoding system. The memory is of a predefined size. The frame buffer further

includes control logic for partitioning the memory of the predefined size into three buffer areas when the digital video decoding system is in a normal video mode, wherein the three buffer areas receive full size I, P & B frames of a decoded stream of video data. The control logic is further adapted to partition the memory into five buffer areas when the digital video decoding system is in a scaled video mode. The five buffer areas comprise a first area and a second area for receiving full size I and P frames of the decoded stream of video data, and at least a third area, fourth area and fifth area for receiving scaled I, P & B frames of the decoded stream of video data.

Methods and articles of manufacture corresponding to the above-outlined systems and frame buffers are also described and claimed herein.

To restate, disclosed herein is a digital video decode system, method and article of manufacture which present an integrated scaling capability. The decoder is arranged such that it reduces the overall bandwidth to external memory when in the scaling mode. For example, the display picture can be reduced by a factor of 2 and/or 4 in both the horizontal and vertical axis. Advantageously, the integrated scaling function for the video decode system presented herein uses existing decoder hardware to produce a high quality scaled image.

By performing decimation/scaling at decode time, the total memory bandwidth requirement is reduced, making more memory bandwidth available to other features, such as on-screen graphics. Thus, scaling implemented in accordance with this invention requires less external memory (i.e., frame buffer memory) than would be required by a post-

processing approach, i.e., a display scaler would require four full size frame buffers. Further, in accordance with this invention, switching between non-scaling and scaling modes does not produce display artifacts. Scaling in accordance with this invention can also be employed with B frame memory reduction in full frame format, as well as with letterbox format.

Brief Description of the Drawings

The above-described objects, advantages and features of the present invention, as well as others, will be more readily understood from the following detailed description of certain preferred embodiments of the invention, when considered in conjunction with the accompanying drawings in which:

Fig. 1 shows an exemplary pair of groups of pictures (GOPs);

Fig. 2 shows an exemplary macroblock (MB) subdivision of a picture (4:2:0 format);

Fig. 3 depicts a block diagram of a video decoder;

Fig. 4 is block diagram of a video decoding system to employ the principles of the present invention;

Fig. 5 is a detailed embodiment of a video decoding system in accordance with the principles of the present invention;

Fig. 6 illustrates frame buffer subdivision in a normal mode and in a video scaling mode in accordance with the present invention;

Fig. 7a is a timing diagram showing delayed display timing in a video scaling mode in accordance with the principles of the present invention;

Fig. 7b illustrates one example of switching of the small frame buffers 2, 4 & 6 of Fig. 6 in accordance with the present invention;

Fig. 8 is a block diagram of one embodiment of a decimation unit in accordance with the principles of the present invention for the video decode system of Fig. 5;

Fig. 9 is a block diagram of one embodiment of display mode switch logic in accordance with the principles of the present invention for the video decode system of Fig. 5; and

Fig. 10 is a flowchart of one embodiment of processing implemented by the sync generator of Fig. 9 in accordance with the principles of the present invention.

Best Mode for Carrying Out the Invention

As the present invention may be applied in connection with an MPEG-2 decoder, in order to facilitate an understanding of the invention, some pertinent aspects of the MPEG-2 compression algorithm are first reviewed. It is to be noted, however, that the invention can also be applied

to other video coding algorithms which share some of the features of the MPEG-2 algorithm.

To begin with, it will be understood that the compression of a data object, such as a page of text, an image, a segment of speech, or a video sequence, can be thought of as a series of steps, including: 1) a decomposition of that object into a collection of tokens; 2) the representation of those tokens by binary strings which have minimal length in some sense; and 3) the concatenation of the strings in a well-defined order. Steps 2 and 3 are lossless, i.e., the original data is faithfully recoverable upon reversal, and Step 2 is known as entropy coding. Step 1 can be either lossless or lossy in general. Most video compression algorithms are lossy because of stringent bit-rate requirements. A successful lossy compression algorithm eliminates redundant and irrelevant information, allowing relatively large errors where they are not likely to be visually significant and carefully representing aspects of a sequence to which the human observer is very sensitive. The techniques employed in the MPEG-2 algorithm for Step 1 can be described as predictive/interpolative motion-compensated hybrid DCT/DPCM coding. Huffman coding, also known as variable length coding, is used in Step 2.

The MPEG-2 video standard specifies a coded representation of video for transmission as set forth in ISO-IEC JTC1/SC29/WG11, Generic Coding of Moving Pictures and Associated Audio Information: Video, International Standard, 1994. The algorithm is designed to operate on interlaced or non-interlaced component video. Each picture has three components: luminance (Y), red color difference (Cr), and blue color difference (Cb). The video data may be coded in 4:4:4 format, in which case there is one Cr and one

Cb sample for each Y sample, in 4:2:2 format, in which case there are half as many Cr and Cb samples as luminance samples in the horizontal direction, or in 4:2:0 format, in which case there are half as many Cr and Cb samples as luminance samples in both the horizontal and vertical directions.

An MPEG-2 data stream consists of a video stream and an audio stream which are packed, together with systems information and possibly other bitstreams, into a systems data stream that can be regarded as layered. Within the video layer of the MPEG-2 data stream, the compressed data is further layered. A description of the organization of the layers will aid in understanding the invention. These layers of the MPEG-2 Video Layered Structure are shown in Figs. 1 & 2. The layers pertain to the operation of the compression algorithm as well as the composition of a compressed bit stream. The highest layer is the Video Sequence Layer, containing control information and parameters for the entire sequence. At the next layer, a sequence is subdivided into sets of consecutive pictures, each known as a "Group of Pictures" (GOP). A general illustration of this layer is shown in Fig. 1. Decoding may begin at the start of any GOP, essentially independent of the preceding GOPs. There is no limit to the number of pictures which may be in a GOP, nor do there have to be equal numbers of pictures in all GOPs.

The third or Picture layer is a single picture. A general illustration of this layer is shown in Fig. 2. The luminance component of each picture is subdivided into 16 × 16 regions; the color difference components are subdivided into appropriately sized blocks spatially co-sited with the 16 × 16 luminance regions; for 4:4:4 video, the color

difference components are 16 x 16, for 4:2:2 video, the color difference components are 8 x 16, and for 4:2:0 video, the color difference components are 8 x 8. Taken together, these co-sited luminance region and color difference regions make up the fifth layer, known as a "macroblock" (MB). Macroblocks in a picture are numbered consecutively in lexicographic order, starting with Macroblock 1.

Between the Picture and MB layers is the fourth or "slice" layer. Each slice consists of some number of consecutive MB's. Finally, each MB consists of four 8 x 8 luminance blocks and 8, 4, or 2 (for 4:4:4, 4:2:2 and 4:2:0 video) chrominance blocks. The Sequence, GOP, Picture, and slice layers all have headers associated with them. The headers begin with byte-aligned Start Codes and contain information pertinent to the data contained in the corresponding layer.

A picture can be either field-structured or frame-structured. A frame-structured picture contains information to reconstruct an entire frame, i.e., the combination of one field containing the odd lines and the other field containing the even lines. A field-structured picture contains information to reconstruct one field. If the width of each luminance frame (in picture elements or pixels) is denoted as C and the height as R (C is for columns, R is for rows), a field-structured picture contains information for C x R/2 pixels.

The two fields in a frame are the top field and the bottom field. If we number the lines in a frame starting from 1, then the top field contains the odd lines (1, 3, 5, ..) and the bottom field contains the even lines (2, 4, 6,

..). Thus we may also call the top field the odd field, and the bottom field the even field.

A macroblock in a field-structured picture contains a 16 x 16 pixel segment from a single field. A macroblock in a frame-structured picture contains a 16 x 16 pixel segment from the frame that both fields compose; each macroblock contains a 16 x 8 region from each of the two fields.

Within a GOP, three types of pictures can appear. The distinguishing difference among the picture types is the compression method used. The first type, Intramode pictures or I-pictures, are compressed independently of any other picture. Although there is no fixed upper bound on the distance between I-pictures, it is expected that they will be interspersed frequently throughout a sequence to facilitate random access and other special modes of operation. Predictively motion-compensated pictures (P pictures) are reconstructed from the compressed data in that picture plus two reconstructed fields from previously displayed I or P pictures. Bidirectionally motion-compensated pictures (B pictures) are reconstructed from the compressed data in that picture plus two reconstructed fields from previously displayed I or P pictures and two reconstructed fields from I or P pictures that will be displayed in the future. Because reconstructed I or P pictures can be used to reconstruct other pictures, they are called reference pictures.

With the MPEG-2 standard, a frame can be coded either as a frame-structured picture or as two field-structured pictures. If a frame is coded as two field-structured pictures, then both fields can be coded as I pictures, the first field can be coded as an I picture and the second

field as a P picture, both fields can be coded as P pictures, or both fields can be coded as B pictures.

If a frame is coded as a frame-structured I picture, as two field-structured I pictures, or as a field-structured I picture followed by a field-structured P picture, we say that the frame is an I frame; it can be reconstructed without using picture data from previous frames. If a frame is coded as a frame-structured P picture or as two field-structured P pictures, we say that the frame is a P frame; it can be reconstructed from information in the current frame and the previously coded I or P frame. If a frame is coded as a frame-structured B picture or as two field-structured B pictures, we say that the frame is a B frame; it can be reconstructed from information in the current frame and the two previously coded I or P frames (i.e., the I or P frames that will appear before and after the B frame). We refer to I or P frames as reference frames.

A common compression technique is transform coding. In MPEG-2 and several other compression standards, the discrete cosine transform (DCT) is the transform of choice. The compression of an I-picture is achieved by the steps of 1) taking the DCT of blocks of pixels, 2) quantizing the DCT coefficients, and 3) Huffman coding the result. In MPEG-2, the DCT operation converts a block of n x n pixels into an n x n set of transform coefficients. Like several of the international compression standards, the MPEG-2 algorithm uses a DCT block size of 8 x 8. The DCT transformation by itself is a lossless operation, which can be inverted to within the precision of the computing device and the algorithm with which it is performed.

The second step, quantization of the DCT coefficients, is the primary source of lossiness in the MPEG-2 algorithm. Denoting the elements of the two-dimensional array of DCT coefficients by cmn, where m and n can range from 0 to 7, aside from truncation or rounding corrections, quantization is achieved by dividing each DCT coefficient cmn by wmn times QP, with wmn being a weighting factor and QP being the quantizer parameter. The weighting factor wmn allows coarser quantization to be applied to the less visually significant coefficients. The quantizer parameter QP is the primary means of trading off quality vs. bit-rate in MPEG-2. It is important to note that QP can vary from MB to MB within a picture.

Following quantization, the DCT coefficient information for each MB is organized and coded, using a set of Huffman codes. As the details of this step are not essential to an understanding of the invention and are generally understood in the art, no further description is needed here.

Most video sequences exhibit a high degree of correlation between consecutive pictures. A useful method to remove this redundancy prior to coding a picture is "motion compensation". MPEG-2 provides tools for several methods of motion compensation.

The methods of motion compensation have the following in common. For each macroblock, one or more motion vectors are encoded in the bit stream. These motion vectors allow the decoder to reconstruct a macroblock, called the predictive macroblock. The encoder subtracts the "predictive" macroblock from the macroblock to be encoded to form the "difference" macroblock. The encoder uses tools to

compress the difference macroblock that are essentially similar to the tools used to compress an intra macroblock.

The type of a picture determines the methods of motion compensation that can be used. The encoder chooses from among these methods for each macroblock in the picture. If no motion compensation is used, the macroblock is intra (I). The encoder can make any macroblock intra. In a P or a B picture, forward (F) motion compensation can be used; in this case, the predictive macroblock is formed from data in the previous I or P frame. In a B picture, backward (B) motion compensation can also be used; in this case, the predictive macroblock is formed from data in the future I or P frame. In a B picture, forward/backward (FB) motion compensation can also be used; in this case, the predictive macroblock is formed from data in the previous I or P frame and the future I or P frame.

Because I and P pictures are used as references to reconstruct other pictures (B and P pictures) they are called reference pictures. Because two reference frames are needed to reconstruct B frames, MPEG-2 decoders typically store two decoded reference frames in memory.

Aside from the need to code side information relating to the MB mode used to code each MB and any motion vectors associated with that mode, the coding of motion-compensated macroblocks is very similar to that of intramode MBs. Although there is a small difference in the quantization, the model of division by wmn times QP still holds.

The MPEG-2 algorithm can be used with fixed bit-rate transmission media. However, the number of bits in each picture will not be exactly constant, due to the different

types of picture processing, as well as the inherent variation with time of the spatio-temporal complexity of the scene being coded. The MPEG-2 algorithm uses a buffer-based rate control strategy to put meaningful bounds on the variation allowed in the bit-rate. A Video Buffer Verifier (VBV) is devised in the form of a virtual buffer, whose sole task is to place bounds on the number of bits used to code each picture so that the overall bit-rate equals the target allocation and the short-term deviation from the target is bounded. This rate control scheme can be explained as follows. Consider a system consisting of a buffer followed by a hypothetical decoder. The buffer is filled at a constant bit-rate with compressed data in a bit stream from the storage medium. Both the buffer size and the bit-rate are parameters which are transmitted in the compressed bit stream. After an initial delay, which is also derived from information in the bit stream, the hypothetical decoder instantaneously removes from the buffer all of the data associated with the first picture. Thereafter, at intervals equal to the picture rate of the sequence, the decoder removes all data associated with the earliest picture in the buffer.

Fig. 3 shows a diagram of a conventional video decoder. The compressed data enters as signal 11 and is stored in the compressed data memory 12. The variable length decoder (VLD) 14 reads the compressed data as signal 13 and sends motion compensation information as signal 16 to the motion compensation (MC) unit 17 and quantized coefficients as signal 15 to the inverse quantization (IQ) unit 18. The motion compensation unit reads the reference data from the reference frame memory 20 as signal 19 to form the predicted macroblock, which is sent as the signal 22 to the adder 25. The inverse quantization unit computes the unquantized

coefficients, which are sent as signal 21 to the inverse transform (IDCT) unit 23. The inverse transform unit computes the reconstructed difference macroblock as the inverse transform of the unquantized coefficients. The reconstructed difference macroblock is sent as signal 24 to the adder 25, where it is added to the predicted macroblock. The adder 25 computes the reconstructed macroblock as the sum of the reconstructed difference macroblock and the predicted macroblock. The reconstructed macroblock is then sent as signal 26 to the demultiplexer 27, which stores the reconstructed macroblock as signal 29 to the reference memory if the macroblock comes from a reference picture or sends it out (to memory or display) as signal 28. Reference frames are sent out as signal 30 from the reference frame memory.

Various techniques have been proposed for reducing memory requirements of a decode system by storing decoded video data in compressed form. One such technique is described in the above-incorporated patent application entitled "Multi-Format Reduced Memory MPEG-2 Compliant Decoder," Serial No. 08/958,632. This co-pending application relates to a method for reducing memory requirements for frame buffer storage for an MPEG-2 decoder, and to editing or modifying the video output, e.g., from a 4:3 form factor of television to a 16:9 format of motion pictures. A significant aspect of the technique is decimation of the B-coded pictures within hardware of the video decoder. This technique includes first motion compensating the P-coded and B-coded pictures, followed by decimating the B-coded pictures during the decode phase, and increasing the size of scan line fetches for the I-coded and P-coded pictures to enable their horizontal decimation during the display phase.

A decode system, generally denoted 40, to employ the present invention is depicted in **Fig. 4**. System 40 includes a bus interface 44 which couples the decode system 40 to a memory bus 42. MPEG encoded video data is fetched from PCI bus 42 by a DMA controller 46 which writes the data to a video First-In/First-Out (FIFO) buffer 48. The DMA controller also fetches on-screen display and/or audio data from bus 42 for writing to an OSD/audio FIFO 50. A memory controller 52 will place video data into a correct memory buffer within dynamic random access memory (DRAM) 53. MPEG compressed video data is then retrieved by the video decoder 54 from DRAM 53 and decoded as described above in connection with **Fig. 3**. Conventionally, the decoded video data is then stored back into the frame buffers of DRAM 53 for subsequent use as already described. When a reference frame is needed, or when video data is to be output from the decode system, stored data in DRAM 53 is retrieved by the MEM controller and forwarded for output via a display & OSD interface 58. Audio data, also retrieved by the memory controller 52, is output through an audio interface 60.

As discussed initially herein, this invention addresses the need for a decoding system with integrated scaling capability which can scale the size of an MPEG-2 video presentation by a predetermined reduction factor. At the same time, the invention preferably allows for reduction in external memory requirements for full-frame buffer storage as well, e.g., using the above-described B frame memory reduction technique. As the MPEG-2 video decoder market becomes more and more competitive, the need for high level of feature integration at the lowest possible cost is important to achieving success in the marketplace. The present invention acknowledges this by providing a scaling mode to reduce the size of a display picture by a predefined

factor, such as 2 and/or 4 in both the horizontal and vertical axis.

Fig. 5 depicts one embodiment of a video decode system in accordance with the principles of the present invention. This video decode system includes external memory 53, which in the embodiment shown comprises SDRAM frame buffer storage. Memory 53 interfaces with a memory control unit 52. Memory control unit 52 receives decoded video data from a video decoder 54 and provides video data for display through video display unit 90. In accordance with the principles of the present invention, video decode system 65 of **Fig. 5** includes numerous features which implement a video scaling mode capability.

For example, decimation unit 82 is modified to include both a normal video decimation mode and a video scaling mode. Frame buffers 53 are modified to accommodate storage of decoded video data in either full-frame format or a combination of full-frame format and scaled video format. Display mode switch logic 96 is provided within video display unit 90 to facilitate seamless switching between normal video mode and scaled video mode. Frame buffer pointer control 86 is modified to provide the correct frame buffer pointers based on the novel partitioning of the frame buffers when in normal video mode and when in scaled video mode. Further, as described in the above-incorporated, co-filed United States patent application, a flicker reduction mechanism is preferably integrated within video display unit 90, and in particular within vertical upsample logic 94. Each of these features is described in detail below, with the exception of flicker reduction which is described in the co-filed application.

Operationally, an MPEG input video source is fed through memory control unit 52 as coded MPEG-2 video data to the input of video decoder 54. Decoder 54 includes a Huffman decoder 72, Inverse Quantizer 74, Inverse DCT 76, Motion Compensation 78 and adder 80, which function as described above in connection with the video decoder of Fig. 3. An internal processor 70 oversees the video decode process and, in accordance with the principles of the present invention, receives a signal from a host system whenever the host desires to switch the video display between, for example, normal video display and scaled video display. This signal is referred to in Fig. 5 as a "host controlled format change" signal. In response to host format changes, control signals are sent from internal processor 70 to Huffman decoder 72, Inverse Quantizer 74, Motion Compensation 78, as well as to upsample logic 94, display fetch unit 92 and display mode switch logic 96 within video display 90. Again, these control signals direct the video decode system in accordance with the principles of the present invention (and as described below) to switch the display output between, for example, normal video mode and scaled video mode.

Full size macroblocks of decoded video data are sequentially output from video decoder 54 to decimation unit 82 where, in one embodiment, the full size macroblocks undergo one of two types of compression. First, if full size video is desired, then decimation of the B-coded pictures only is still preferably performed as described in the above-incorporated application entitled: "Multi-Format Reduced Memory MPEG-2 Compliant Decoder". In this normal video mode, decimation is a process of reducing the amount of data by interpolating or averaging combined values to get an interpolated pixel value. Interpolation reduces the

number of pixels, and therefore, less external memory is required in the overall system. In a second mode, decimation unit 82 performs picture scaling in accordance with the principles of this invention. By way of example, the type of scaling employed may reduce the overall size of the display picture by a factor of 2 or 4 in both the horizontal and vertical axis.

Along with providing decimation unit 82 with a stream of decoded full-size macroblocks, video decoder also sends a "motion compensation unit block complete" signal on line 83, which lets decimation unit 82 know when a macroblock has been completely decoded. Similarly, decimation unit 82 provides a "decimator busy" signal on line 85 to motion compensation unit 78 of video decoder 54. This "decimator busy" signal informs the motion compensation unit when the decimation unit is busy and when the unit has completed its operations, after which the motion compensation unit can proceed to the next macroblock.

Motion compensation unit 78 of video decoder 54 provides read video addresses directly to memory control unit 52, and write video addresses to decimation unit 82 for writing of decoded video data (full size) and/or scaled macroblocks to external memory 53. In parallel with the read video address and write video address, pointers are provided by frame buffer pointer control 86 to the memory control unit. These pointers define which frame buffer areas within SDRAM 53 are to be accessed by a given read video address or write video address in accordance with the partitionings of the frame buffer memory space pursuant to this invention (as described further below). These pointers are referred to in Fig. 5 as current pointer and current small pointer, with current pointer comprising a pointer for

a full size macroblock, and current small pointer comprising a pointer for a scaled macroblock.

Decimation unit 82 receives the decoded full-size macroblocks, buffers the information internally and if scaling mode is activated, performs scaling as described below. In a normal mode, decimation unit 82 outputs decoded video data full-size macroblocks to memory control unit 52 for storage in frame buffers 53. When in scaling mode, decimation unit 82 scales the full-size macroblocks and outputs scaled macroblocks to memory control unit 52 for storage in frame buffers 53.

Frame buffer pointer control 86 is significant and controls rotation of the frame buffers, i.e., frame buffer assignments, when in normal video mode and video scaling mode in accordance with the principles of the present invention (described further below).

As described in the above-incorporated applications, decimation unit 82 also functions as part of video display unit 90 when retrieving data for display. Specifically, decoded video data comprising full-size scan lines is retrieved from frame buffer storage 53 and fed through decimation unit 82 for B-frame re-expansion of pictures as explained in the above-incorporated patent application entitled "Multi-Format Reduced Memory Video Decoder With Adjustable Polyphase Expansion Filter." This is done so that consistency is maintained for the video within a group of pictures, and thus reduced resolution of any one picture is not perceptible. After re-expansion, the full-size scan lines are provided to display output interface 98.

Alternatively, when in video scaling mode, decoded video comprising scaled scan lines is retrieved from frame buffer storage 53 and fed directly to scan line video buffers 84. The scan lines are divided between luminance and chrominance data and both a current scan line and a prior scan line are fed from scan line video buffers 84 to vertical and horizontal upsample logic 94. Upsample controls are received from display fetch unit 92, which coordinates letterbox formatting, SIF upsampling, 4:2:0 to 4:2:2 upsampling, and flicker reduction (in accordance with the principles of the above-incorporated, co-filed patent application).

Display fetch unit 92 provides the read video address for retrieval of scan lines from frame buffer storage 53. A "current pointer, current small pointer" synchronization (sync) signal for display is received by memory control unit 52 from display mode switch logic 96 of video display unit 90. As noted above, the current pointer, current small pointer signal points to the particular frame buffer area from which scan lines are to be retrieved, while the read video address signal designates the particular scan lines to be retrieved within that frame buffer area.

Display mode switch logic 96 is provided in accordance with the principles of the present invention in order to ensure seamless switching between, for example, scaled video mode and normal video mode. Logic 96 receives as input a control signal from internal processor 70 of video decoder 54, as well as a vertical synchronization (VSYNC) signal (from display output interface 98) and a B picture "MPEG-2 repeat field" signal from Huffman decoder 72 of video decoder 54. VSYNC is an external synchronization signal that indicates the start of a new display field. Output

from display mode switch logic 96, in addition to the current pointer, current small pointer sync for the display, is a "display format sync for display" signal fed to display fetch unit 92, as well as a "display format sync for decode" signal fed to the decode logic of decimation unit 82. Display mode switch logic 96 also outputs a "block video" signal to display output interface 98 which is employed, in accordance with the principles of the present invention, to block one display frame to keep noise from the display when switching between display modes. Video data is received at the display output interface from upsample logic 94. The decimation unit, frame buffer partitioning, frame buffer pointer control and display mode switch logic, all implemented in accordance with the principles of this invention, are described in greater detail below with reference to **Figs. 6-10**.

First, the frame buffers. The frame buffers are used to store the constructed pictures for display, as well as for prediction of subsequent pictures. Since a B picture is not used for prediction, its frame buffer is available for use after the picture has been displayed. For I or P pictures, the frame buffer needs to be held after display, particularly for predicting B pictures.

Fig. 6 depicts frame buffer allocation for both normal video mode and scaled video mode in accordance with the principles of this invention. In normal mode, there are three frame buffers to support the decoding and display processes. Frame buffer 0 and frame buffer 1 are allocated for I and P pictures, while frame buffer 2 is assigned to B pictures. The frame buffers are tagged by buffer pointers, i.e., the current pointers from frame buffer pointer control 86 of **Fig. 5**.

In the scaled video mode, at least five frame buffers are employed. Frame buffer 0 and frame buffer 1 again accommodate full size I and P picture video. The at least three other buffers, which are labeled frame buffer 2, frame buffer 4 and frame buffer 6 in the example shown, are tagged by small pointers generated by the frame buffer pointer control. These small buffers are mainly used for display purposes when in scaled video mode. The buffers are small size in order to fit the video scalings. When decoding an I or P picture, the constructed picture is stored into either buffer 0 or buffer 1 depending upon whichever is available. At the same time, a scaled down version of the same picture is stored into one of the smaller buffers, i.e., frame buffer 2, frame buffer 4 or frame buffer 6. The full size video is then used for prediction, while the small sized video in the small frame buffers is used for display of the scaled down picture.

The frame buffers are configured by microcode during initialization of the video decode system. A memory base address is assigned to each frame buffer, and these memory base addresses are selected by the buffer pointers generated by frame buffer pointer control. The read and write video addresses refer to specific addresses within a selected frame buffer. Unless otherwise indicated, the term "frame buffer" is used herein below as inclusive of all frame buffer memory configured during initialization. "Frame buffer area" refers to one of the specific frame buffers depicted in Fig. 6.

The video display operates in real time, and therefore, frame buffer pointers must be switched according to the VSYNC timing. Since decoding is always ahead of the display, a frame buffer must be made available to store the

decoded picture. Therefore, the frame buffer pointers must be switched before decoding starts. To avoid the disturbance to the display frame buffer, a copy of the display buffer pointer is maintained. The buffer switching time is at the beginning of each picture decode. The display buffer pointer is also changed at that time, however, it will not be used until copy display pointer time which is the beginning of picture display. One embodiment of normal mode buffer pointer rotation is described below.

The following assumes four buffer pointers, with each pointer containing two bits to indicate which one of the three frame buffers (buffer 0, 1 and 2) is being accessed.

- current pointer - indicates the frame buffer to be used for the constructing picture;
- display pointer - indicates the frame buffer to be used for the display;
- future pointer - indicates the frame buffer to be used for the backward prediction; and
- past pointer - indicates the frame buffer to be used for the forward prediction.

At startup, the future pointer is initialized to "1" and the other pointers are set to "0". At the beginning of an I or P picture decode, the value from the past pointer is loaded into the current pointer and the value from the future pointer is loaded into the display pointer. The values in the future pointer and the past pointer are swapped. If the decoding picture is a B picture, the current pointer and the display pointer are set to "2".

Frame buffer 2 is reserved, in one example, for the B pictures. The future pointer and past pointer remain unchanged. Pointer switching in normal mode is described in greater detail in United States patent number 5,668,599 by Cheney et al., entitled "Memory Management For An MPEG-2 Compliant Decoder," the entirety of which is hereby incorporated herein by reference.

In scaled video mode, the display time of a picture is delayed by an additional field time in accordance with the present invention. The purpose of this delay is to decouple the decode and the display processes so that the decoded scaled video can be placed anywhere on the screen. **Fig. 7a** depicts one example of the delayed display timing in scaled video mode. This display timing is adjusted dynamically according to the mode, i.e., whether normal mode or scaled video mode. The one field time delay is needed in accordance with this invention to properly manage the buffers. At least five buffers are again assumed in the video scaling mode. As described above, two of these five buffers comprise full-size frame buffers, and are labeled frame buffer 0 and frame buffer 1 in **Fig. 6**. These full-size frame buffers are the same as the corresponding buffers used in normal video mode. The at least three small frame buffers, i.e., frame buffer 2, frame buffer 4 and frame buffer 6, are allocated in the same memory space occupied by frame buffer 2 used in the normal video mode. These three small frame buffers are controlled by a different algorithm than described above.

Specifically, four additional pointers are used in scaled video mode. These pointers are:

- small current pointer - indicates a small buffer for the decimated constructing picture;
- small display pointer - indicates a small buffer for the display;
- small future pointer - indicates the small buffer for the future display; and
- small transition pointer - indicates the small buffer for the transition.

When the decoder is initialized, the small current pointer, small display pointer, small future pointer and small transition pointer are set to 0, 2, 4 and 6, respectively. At the start of each picture decoding, the small current pointer is loaded from the small transition pointer and the small transition pointer is loaded from the small display pointer. If the decoding picture is a B picture, then the small display pointer is loaded from the small transition pointer and the small future pointer remains unchanged. If the decoding picture is an I or P picture, the small display pointer is loaded from the small future pointer and the small future pointer is loaded from the small transition pointer. One example of small frame buffer switching in accordance with the present invention is depicted in **Fig. 7b**.

The full-size frame buffers, frame buffer 0 and frame buffer 1, are switching as if the decoder is running in normal mode. These two buffers are needed for prediction, but are not for display in scaled video mode. When an I or P picture is being decoded, the picture is stored in both buffers indicated by the current (full frame) pointer and

the small current pointer. During a B picture decoding, frame buffer 2 indicated by the current (full frame) pointer will not be used. Only the small frame buffer identified by the small current pointer is used for the decimated picture. In normal mode, the display pointer is used for display, while in scaled video mode, the small display pointer is used. The two buffer switching algorithms operate simultaneously at the beginning of each picture decoding. The buffer pointers are simply selected depending upon which mode the decoder is in.

Next, Fig. 8 depicts one embodiment of a decimation unit 82 (Fig. 5) employed in accordance with the present invention.

In previous implementations of the decode decimation unit, e.g., described in the above-incorporated patent application entitled "Multi-Format Reduced Memory MPEG-2 Compliant Decoder", the decimation unit was limited to operating on B pictures only, for either letterbox or memory reduction purposes. In a scaled video mode as presented herein, however, the decode decimation unit processes all picture types. This is desirable in order to save memory bandwidth at display time, since (in one embodiment) scaled pictures and multi-plane, high resolution OSD graphics may be mixed at the output.

In the embodiment of Fig. 8, the decimation unit includes decimation logic 200, which receives the decoded video data from the video decoder and outputs a decimated data flow to a decimation buffer 220. Output from decimation buffer 220 is multiplexed 230 with the undecimated, decoded video data received from the video decoder such that multiplexer 230 outputs the decoded video

data, as well as the scaled macroblocks for storage in frame buffers 0, 1, 2, 4 and 6 as described above when in scaled video mode. The write video address from the motion compensation unit of the video decoder is fed to memory write controls 240 within the decimation unit, which control writing of data from decimation buffer 220. The write video addresses, either with or without decimation scaling, are also output through a multiplexer 250 to the memory control unit (see **Fig. 5**).

Multiplexers 230 & 250 are controlled by decimate control signals 210. The decimate control logic receives as input a signal called "MCU_block_complete" from the motion compensation unit of the video decoder. This signal indicates when the decimator can begin to write the scaled macroblock. The decimator informs the motion compensation unit that it is currently busy through a signal labeled "decimator_busy".

For a given macroblock, there are two phases. One phase is for the luminance, and the other phase is for chrominance. Each phase requires a write of one full-sized macroblock and one scaled macroblock, again, assuming scaled video mode.

Various specific changes to the decimation hardware/process described in the above-incorporated "Multi-Format Reduced Memory MPEG-2 Compliant Decoder" application are intended herein. One change in the data flow of the decimation process is the addition (in one example) of a 4 to 1 horizontal reduction, which is implemented in the horizontal decimate function of the decimation logic. This is to support 1/16 size scaling.

Another change is to increase the decimation buffer size to 32 x 32 bits. As I and P pictures are processed, the full-sized macroblock is written to memory, while the decimator scales down the macroblock at the same time and stores a small macroblock in the decimation buffer 220. After the full-sized macroblock is written to memory, the decimator writes the scaled macroblock to another buffer location within memory (i.e., frame buffer 2, frame buffer 4 or frame buffer 6 in the example above). The larger decimation buffer allows for the storing of the small macroblock.

The decimate state machine logic is also changed to allow two modes of operation, i.e., again assuming a scaled video mode. The first mode is B picture processing and the second mode is reference picture processing. For B picture processing, only the small macroblocks are written to memory through decimation buffer 220. The data is paced through the decimation unit as fast as the motion compensation unit can deliver it, since the decimation buffer can hold a complete scaled macroblock. For reference picture operations, the full-size macroblocks are written to memory first through multiplexer 230, followed by the writing of the scaled macroblocks. This requires the data flow to be paced by the memory control unit responding to requests for writing.

Since the size of the source compressed image may vary, there are exceptions to the above process. The decimator is only required if some type of reduction is needed to form a scaled picture. Certain video sources will already be small in size, and one dimension, or both dimensions may not require scaling. For example, it is common to have 352 x 240 sized images (typical MPEG-1 size). In this case, it

would be unnecessary to do any decimation to provide a 1/4 scaling. For reference frames, the motion compensation unit is required to write the full-sized macroblock to the reference frame buffer in memory, and then to the display frame buffer in memory, since the display process is only operating on the display frame buffer during scaling.

For the same image size to be reduced to 1/16 scaling, there would need to be a decimation step. Once again, there is an exception in this case.

One of the objects of the scaling feature is to remove interlacing artifacts. On true MPEG-1 images, there is no interlacing since the pictures are exclusively frame encoded. MPEG-2 can allow interlaced pictures of the same resolution (352 x 240) and the decimator only uses the top field picture to create the scaled macroblock. The bottom field is discarded. Therefore, for a reference picture, the MCU would be required to write the macroblock for the top field picture to both the reference frame buffer and display buffer. For B pictures, the MCU would only need to write the top field picture into the display frame buffer.

The video decode system in accordance with the present invention provides smooth transitions when entering and exiting small picture mode. Since frame buffer 2 is used for capture and display of small picture images (including reference and B pictures) when in video scaling mode, care must be taken to prevent interference between the decode and display processes at the time of display format switching. Also, there is a latency adjustment of one field time that must take place during the transition. Normal display modes have a 1.5 frame latency between decoding and displaying reference pictures, and a 0.5 frame latency for B pictures.

In small picture mode, the reference frame latency is changes to two frames and the B frame latency is changed to one frame.

For the display format change to occur seamlessly, the display must not be in the process of displaying a B picture when the transition occurs, otherwise the picture will appear disturbed. Therefore, transition must take place when a reference picture is being displayed. This is forced to happen by the microcode during a sequence header, when the first frame of the new sequence is a reference frame, and the display is acting on the last frame of a previous sequence.

During the transition into and out of small picture mode, the hardware must make the adjustment in latency without disturbing the decode or display process. Frame sync must be adjusted to the new mode. Further, field parity must be maintained. As a result of making the adjustment into small picture mode, a delay of one frame time is introduced, which may effect PTS comparison. Subsequently, a skipped frame may be required in order to make up the time difference. This only occurs when entering small picture mode. When leaving small picture mode, there is no loss of synchronization. The transition could also come at a time when a picture is already being skipped or repeated.

Referring to Fig. 9, the display format change signal is written asynchronously by the host. The format is received as a control signal into a display format register 310, and microcode waits until processing the sequence header before writing the information into display format register 310. This information is then seen by sync

generator 300, as well as register stages 330, 340 and 360. Register stage 1 330 captures the information at the next frame sync. Decode processes use the stage 1 register 330, and the display processes use the stage 3 register 360.

Field counter 320 simply counts down from a starting number of fields in a frame to a value of 1, and then repeats. Counter 320 is loaded by sync generator 300 via a control signal as shown. Sync generator 300 also receives the VSYNC signal, as well as the output of stage 1 register 330. Sync generator 300 creates three signals, namely, a "frame sync" signal, a "new picture" signal and a "block video" signal. The "frame sync" signal indicates to the decode process when to begin decoding a new frame. The "new picture" signal indicates to the display process when to begin displaying a new frame. "Block video" is used to selectively suppress one frame of video image during transition of the video decode system from a normal frame to a scaled frame. The frame sync and new picture signals are pulses that are created once every two field times. In normal mode, the signals are 180° out of phase, but in scaling mode (in accordance with this invention) the signals are in phase. This is described further below in connection with the flowchart of **Fig. 10**.

In all cases involving a switch into scaled picture mode, there is a repeated frame which is blocked from view at the display. The block is necessary due to buffer conflicts between the current reference frame and the reference frame that is currently being displayed. When video is blocked, the output of the decoder can be forced to a background color, such as black.

The latency adjustment is performed as soon as the stage 1 register changes. There is an absence of a frame sync which allows the current display frame to be scheduled to repeat. The sync generator then adjusts the frame sync to occur in phase with the new picture, causing the latency adjustment. During the repeated reference frame, the video is blocked for one frame time.

Fig. 10 is a flowchart of one embodiment of processing implemented by sync generator 300 (**Fig. 9**).

With initiation 600, processing waits for a VSYNC signal 610 representative of the start of a new field. Upon receipt of the VSYNC signal, processing generates a "new picture" sync signal and inquires whether the field is being repeated based on received MPEG-2 syntax 630. The initial field counter (FC) value depends upon whether the field is to be repeated. If 3:2 pulldown is employed then the initial value of the field counter is 3 640, otherwise normal interlace is desired and the field counter is loaded with a value of 2.

Once the field counter is set, processing inquires whether scaling is to be implemented 650 and 670, respectively. If no, then the decode system is in non-scaling or normal video mode. In this case, processing waits for a next VSYNC signal 680 and then inquires whether the field count equals two 690. If no, (e.g., because the field counter was loaded with a value 3), the field counter is decremented 710 and processing waits for a next VSYNC signal 680. Once the field count equals 2, the "frame sync" signal is generated 700, after which the field count is decremented 710 and processing determines whether the field count value now equals 1 720. If the value equals 1,

processing generates a "new picture" signal 620 after waiting for a new VSYNC 610.

Assuming scaling mode is desired, then processing proceeds from inquiry 650 or 670 to wait for a next VSYNC 730, after which determination is made whether the field count equals 1 740. If no, the field counter is decremented and processing returns to wait for a next VSYNC 730. If the field count value is 1, then a new picture sync signal is generated 750. Thereafter, the field counter is loaded with a value of 2 and the block video signal is generated 760. Again, the block video signal is output from the sync generator to the display output interface (see **Fig. 5**) for blocking of a next frame of video.

After sending the block video signal, processing enters a steady state, video scaling subprocess beginning by waiting for a next VSYNC signal 780, after which processing determines whether the field count equals 1 790. If no, processing inquires whether the field count equals 2 840, and again if no, processing decrements the field counter 860 and returns to wait for the next VSYNC signal 780. Otherwise, a determination is made as to whether the scaling command has now been turned off by the host system 850. If no, the field counter is decremented and processing waits for a next VSYNC signal 780. If the scaling mode has been switched off, then the field counter is decremented at instruction 710 in the non-scaling process described above.

If the field count equals 1 at inquiry 790, then processing generates both the "new picture" signal and the "frame sync" signal in the same phase. Again, to implement scaling it is necessary to change the latency between the decode process and the display process from one and a half

frame times to two frame times for reference pictures, making the new picture signal and frame sync signal in phase. Processing then determines whether the MPEG-2 repeat field is set 810 to decide whether to load the field counter with a value of 2 830 or 3 820 depending upon whether normal interlacing or 3:2 pulldown is desired. This is necessary even though an adjustment in latency is made in order to accommodate any type of frame rate conversion. After setting the field counter, processing returns to wait for a next VSYNC signal 780.

Those skilled in the art will note from the above discussion that in order to implement smooth switching between normal mode and scaled video mode it is necessary to pass through a transitional phase before reaching the steady state scaling process. Further, it is necessary to adjust the frame sync signal to occur within the same phase as the new picture signal.

The present invention can be included, for example, in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. This media has embodied therein, for instance, computer readable program code means for providing and facilitating the capabilities of the present invention. The articles of manufacture can be included as part of the computer system or sold separately.

Additionally, at least one program storage device readable by machine, tangibly embodying at least one program of instructions executable by the machine, to perform the capabilities of the present invention, can be provided.

The flow diagrams depicted herein are provided by way of example. There may be variations to these diagrams or the steps (or operations) described herein without departing from the spirit of the invention. For instance, in certain cases, the steps may be performed in differing order, or steps may be added, deleted or modified. All of these variations are considered to comprise part of the present invention as recited in the appended claims.

While the invention has been described in detail herein in accordance with certain preferred embodiments thereof, many modifications and changes therein may be effected by those skilled in the art. Accordingly, it is intended by the appended claims to cover all such modifications and changes as fall within the true spirit and scope of the invention.

Claims

1. An integrated video decoding system comprising:

a video decoder for decoding an encoded stream of video data, said video decoder producing a decoded stream of video data; and

a decimation unit coupled to said video decoder and adapted to selectively scale said decoded stream of video data for display.

2. The video decoding system of claim 1, wherein said video decoding system further comprises a frame buffer, and wherein said decimation unit comprises means for scaling said decoded stream of video data prior to storage thereof in said frame buffer.

3. The video decoding system of claim 2, wherein said frame buffer comprises multiple defined memory areas for receiving I, P & B frames of said decoded stream of video data, said multiple defined memory areas comprising a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with said first area and said second area receiving said full size I and P frames.

4. The video decoding system of claim 3, wherein said at least one third area comprises at least three areas, said at least three areas receiving said scaled I, P & B frames of said decoded stream of video data commensurate with said first area and said second area receiving said full size I and P frames of said decoded stream of video data.

5. The video decoding system of claim 2, wherein said frame buffer comprises memory of predefined size, and wherein said video decoding system further comprises a normal video mode for displaying said decoded stream of video data without scaling, and a scaled video mode for scaling said decoded stream of video data prior to display, and wherein said video decoding system further comprises frame buffer pointer control logic for partitioning said memory of predefined size into multiple buffer areas when said video decoding system is in said normal video mode, wherein said multiple buffer areas comprise at least three buffer areas for receiving decoded I, P & B frames of said decoded stream of video data, and wherein said frame buffer pointer control logic comprises means for partitioning said memory of predefined size into at least five buffer areas when said video decoding system is in said scaled video mode, said at least five buffer areas comprising a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and at least a third area, a fourth area and a fifth area for receiving scaled I, P & B frames of said decoded stream of video data.

6. The video decoding system of claim 5, wherein said decimation unit comprises means for performing B frame memory reduction only on said decoded stream of video data when said video decoding system is in said normal video mode, said B frame memory reduction being performed prior to storage of each B frame of said decoded stream of video data in said frame buffer.

7. The video decoding system of claim 1, wherein said video decoding system further comprises a frame buffer, and wherein said decimation unit comprises means for scaling said decoded stream of video data prior to storage thereof in said frame buffer, and wherein said video decoding system further comprises a normal video mode and a scaled video mode, and wherein said decimation unit is adapted to scale said decoded stream of video data for display when in said scaled video mode.

8. The video decoding system of claim 7, wherein said decimation unit is adapted to perform B frame memory reduction only on said decoded stream of video data prior to storage for each B frame in said frame buffer when said video decoding system is in said normal video mode.

9. The video decoding system of claim 7, further comprising display mode switch logic for switching between said normal video mode and said scaled video mode, wherein said switching occurs without perceptual degradation of the display.

10. The video decoding system of claim 9, wherein said display mode switch logic comprises means for blocking a frame of video from said display when switching between said normal video mode and said scaled video mode.

11. The video decoding system of claim 10, wherein said display mode switch logic comprises a sync generator, said sync generator outputting a "new picture" signal and a "frame sync" signal, and wherein said sync generator comprises means for outputting said new picture signal and said frame sync signal in different phase when in said normal video mode, and means for outputting said new picture signal and said frame sync signal in phase when in said scaled video mode.

12. The video decoding system of claim 9, wherein a frame of said decoded stream of video data includes an MPEG-2 repeat field signal, and wherein said display mode switch logic comprises means for ignoring said MPEG-2 repeat field signal when decoding said frame simultaneous with switching between said normal video mode and said scaled video mode.

13. The video decoding system of claim 9, wherein said video decoder comprises a decode process, and wherein said display comprises a display process, and said display mode switch logic comprises means for changing a latency between said decode process and said display process when switching between said normal video mode and said scaled video mode, said changing of said latency ensuring that said decoded stream of video data is accommodated within said frame buffer when in said scaled video mode.

14. The video decoding system of claim 1, wherein said decimation unit's scaling of said decoded stream of video data comprises at least one of one-quarter or one-half frame reduction.

15. A digital video decoding system comprising:

a video decoder for decoding an encoded stream of video data, said video decoder producing therefrom a decoded stream of video data;

a video scalar for scaling said decoded stream of video data prior to storage thereof in a frame buffer, wherein said video decoding system comprises a normal video mode and a scaled video mode, said video scalar scaling said decoded stream of video data when said digital video decoding system is in said scaled video mode; and

display mode switch logic for switching between said normal video mode and said scaled video mode, wherein said switching occurs without perceptual degradation of a video display of said decoded stream of video data.

16. The digital video decoding system of claim 15, wherein said display mode switch logic includes logic for blanking said video display for a defined period of time when switching between said normal video mode and said scaled video mode.

17. The digital video decoding system of claim 16, wherein said predefined period of time for blanking said video display comprises one frame time of the digital video decoding system.

18. The digital video decoding system of claim 15, wherein said video scalar comprises a decimation unit coupled to said video decoder and adapted to scale said decoded stream of video data prior to storage thereof in said frame buffer when said digital video decoding system is in said scaled video mode, said decoded stream of video data comprising I, P & B frames and said decimation unit being adapted to scale said I, P & B frames prior to storage thereof in said frame buffer.

19. The digital video decoding system of claim 18, wherein said decimation unit is further adapted to perform B frame memory reduction only of said decoded stream of video data prior to storage of B frames thereof in said frame buffer when said digital video decoding system is in said normal video mode.

20. The digital video decoding system of claim 18, wherein said video decoder comprises a decode process and said video display comprises a display process, and wherein said decode process and said display process have a first latency when in said normal video mode, and a second latency when in said scaled video mode, and wherein said display mode switch logic comprises means for switching between said first latency and said second latency when switching between said normal video mode and said scaled video mode.

21. The digital video decoding system of claim 20, wherein said display mode switch logic comprises a sync generator, said sync generator comprising means for ignoring an MPEG-2 repeat field signal of a frame of said I, P & B frames being processed when switching between said normal video mode and said scaled video mode.

22. The digital video decoding system of claim 21, wherein said sync generator outputs a "new picture" signal and a "frame sync" signal for use in retrieving from said frame buffer and displaying said decoded stream of video data, wherein said sync generator outputs said new picture signal and said frame sync signal in phase when said digital video decode system is in said scaled video mode.

23. The digital video decoding system of claim 18, wherein said frame buffer comprises multiple defined memory areas for receiving I, P & B frames of said decoded stream of video data, said multiple defined memory areas comprising a first area and a second area for receiving unscaled I and P frames of said decoded stream of video data, and at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with the first area and the second area receiving said unscaled I and P frames.

24. The digital video decoding system of claim 23, wherein said unscaled I and P frames comprise full size I and P frames, and wherein said scaled I, P & B frames comprise a fractional size of said full size I and P frames, said fractional size comprising at least one of one-quarter or one-half said full size.

25. A digital video decoding system comprising:

a normal video mode and a scaled video mode, wherein when in said normal video mode, full size frames are output for display on a video display coupled to said digital video decoding system and when in scaled video mode, scaled frames comprising a fractional size of said full size frames are output for display on said video display;

a frame buffer for temporarily storing said full size frames and said scaled frames after a decoding time thereof and prior to a display time thereof, wherein there is a predefined latency between said decoding time and said display time; and

wherein said predefined latency between said decoding time and said display time comprises a first latency when in normal video mode and a second latency when in scaled video mode.

26. The digital video decoding system of claim 25, wherein said decoded stream of video data comprises an interlaced format, and said first latency comprises one and one half frame times for reference frames, and said second latency comprises two frame times for reference frames.

27. The digital video decoding system of claim 25, further comprising display mode switch logic for switching between said normal video mode and said scaled video mode, wherein said switching occurs without perceptual degradation of said video display.

28. The digital video decoding system of claim 27, wherein said display mode switch logic includes logic for blanking said video display for a predefined period of time when switching between said normal video mode and said scaled video mode.

29. The digital video decoding system of claim 27, wherein said display mode switch logic includes a sync generator, said sync generator including means for ascertaining during said switching from normal video mode to scaled video mode whether a next frame to be displayed includes an MPEG-2 repeat field signal, and if so, for suppressing said MPEG-2 repeat field signal when displaying said next frame.

30. The digital video decoding system of claim 25, wherein said frame buffer comprises multiple defined memory areas for receiving I, P & B frames of a decoded stream of video data, said multiple defined memory areas comprising a first memory area and a second memory area for receiving full size I and P frames of said decoded stream of video data, and when in said scaled video mode, at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with said first area and said second area receiving said full size I and P frames.

31. The digital video decoding system of claim 30, further comprising a video decoder for decoding an encoded stream of video data and producing therefrom said decoded stream of video data, and a decimation unit coupled to said video decoder and adapted to scale said decoded stream of video data for said video display when said digital video decoding system is in said scaled video mode.

32. A frame buffer for a digital video decoding system having video scaling capabilities, said frame buffer comprising:

multiple defined memory areas for receiving I, P & B frames of a decoded stream of video data, said multiple defined memory areas comprising:

(i) a first area and a second area for receiving full size I and P frames of said decoded stream of video data; and

(ii) at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with said first area and said second area receiving said full size I and P frames.

33. The frame buffer of claim 32, wherein said at least one third area comprises at least three areas, said at least three areas receiving said scaled I, P & B frames of said decoded stream of video data.

34. The frame buffer of claim 33, wherein said multiple defined memory areas comprise a predefined region of memory coupled to said digital video decoding system, wherein said full size I and P frames of said decoded stream of video data are employed by a decode process of said digital video decoding system and said scaled I, P & B frames of said decoded stream of video data are employed by a display process of said digital video decoding system when said digital video decoding system is in a scaled video mode.

35. A frame buffer for a digital video decoding system having video scaling capabilities, said frame buffer comprising:

memory associated with said digital video decoding system, said memory being of a predefined size;

control logic for partitioning said memory of predefined size into three buffer areas when said digital video decoding system is in a normal video mode, wherein said three buffer areas receive full size I, P & B frames of a decoded stream of video data; and

wherein said control logic is further adapted to partition said memory into at least five buffer areas when said digital video decoding system is in a scaled video mode, said at least five buffer areas comprising a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and at least a third area, a fourth area and a fifth area for receiving scaled I, P & B frames of said decoded stream of video data.

36. A method for processing an encoded stream of video data employing a digital video decoding system, said method comprising:

decoding the encoded stream of video data to produce a decoded stream of video data;

scaling the decoded stream of video data when the digital video decoding system is in a scaled video mode, said scaling producing a scaled decoded stream of video data; and

after said scaling, buffering said scaled decoded stream of video data in a frame buffer to await display thereof.

37. The method of claim 36, further comprising displaying said scaled decoded stream of video data upon retrieval from said frame buffer, wherein said decoding and said displaying occur in phase when said digital video decoding system is in said scaled video mode.

38. The method of claim 37, wherein said decoding and said displaying occur out of phase when said digital video decoding system is in a normal video mode in which said decoded stream of video data does not undergo said scaling.

39. The method of claim 38, further comprising switching between said normal video mode and said scaled video mode, said switching comprising transitioning said decoding and said displaying from out of phase in said normal video mode to in phase in said scaled video mode.

40. The method of claim 39, wherein said switching includes blanking said display for one frame time when transitioning between said normal video mode and said scaled video mode.

41. The method of claim 39, wherein said switching between said normal video mode and said scaled video mode comprises delaying switching until said display is displaying a reference frame of said decoded stream of video data.

42. The method of claim 36, wherein said decoded stream of video data comprises I, P & B frames, and wherein said scaling comprises scaling said I, P & B frames when said digital video decoding system is in said scaled video mode.

43. The method of claim 42, wherein said buffering comprises buffering unscaled I and P frames in said frame buffer commensurate with buffering scaled I, P & B frames when in said scaled video mode.

44. The method of claim 43, further comprising performing B frame memory reduction only on said decoded stream of video data when said digital video decoding system is in a normal video mode, said B frame memory reduction on said decoded stream of video data being performed prior to buffering of said decoded stream of video data in said frame buffer.

45. The method of claim 36, wherein said digital video decoding system includes a normal video mode, and said method comprises switching between said normal video mode and said scaled video mode, and wherein said switching comprises ignoring an MPEG-2 repeat field signal of a frame of the decoded stream of video data being processed for display when switching between said normal video mode and said scaled video mode.

46. A method for processing an encoded stream of video data employing a digital video decoding system having a normal video mode and a scaled video mode, said method comprising:

decoding the encoded stream of video data to produce a decoded stream of video data;

scaling, when in said scaled video mode, the decoded stream of video data prior to storage thereof in a frame buffer; and

switching between said normal video mode and said scaled video mode without perceptual degradation of a video display of said decoded stream of video data.

47. The method of claim 46, wherein said switching includes blanking said video display for at least one frame time when transitioning between said normal video mode and said scaled video mode.

48. The method of claim 46, further comprising decimating, when in said normal video mode, the decoded stream of video data prior to storage thereof in a frame buffer, said decimating comprising performing B frame memory reduction only on said decoded stream of video data.

49. The method of claim 46, wherein said switching comprises ignoring an MPEG-2 repeat field signal of a frame of the decoded stream of video data being processed for display when switching between said normal video mode and said scaled video mode.

50. The method of claim 46, further comprising displaying said decoded stream of video data upon retrieval from said frame buffer, wherein said decoding and said displaying occur in phase when said digital video decoding system is in said scaled video mode, and said decoding and said displaying occur out of phase when said digital video decoding system is in said normal video mode.

51. A method for processing an encoded stream of video data employing a digital video decoding system having a normal video mode and a scaled video mode, said method comprising:

decoding the encoded stream of video data to produce a decoded stream of video data;

buffering the decoded stream of video data;

displaying the buffered decoded stream of video data; and

controlling said displaying and said decoding such that there exists a first latency between decoding time and displaying time for a frame when said digital video decoding system is in said normal video mode and a second latency between decoding time and display time for a frame when said digital video decoding system is in said scaled video mode.

52. The method of claim 51, wherein said decoded stream of video data comprises an interlaced format, and wherein said first latency comprises one and one half frame times for reference frames, and said second latency comprises two frame times for reference frames.

53. The method of claim 51, further comprising switching between said normal video mode and said scaled video mode, said switching including changing between said first latency and said second latency of decoding time to display time, wherein said switching occurs without perceptual degradation of the display of said decoded stream of video data.

54. The method of claim 53, wherein said switching includes blanking said display for a predefined period of time when switching between said normal video mode and said scaled video mode.

55. The method of claim 54, wherein said switching comprises ascertaining when switching from normal video mode to scaled video mode whether a next frame to be displayed includes an MPEG-2 repeat field signal, and if so, suppressing said MPEG-2 repeat field signal when displaying said next frame.

56. The method of claim 51, wherein said buffering comprises buffering the decoded stream of video data in a frame buffer comprising multiple defined memory areas for receiving I, P & B frames of said decoded stream of video data, said buffering comprising, when in said scaled video mode, buffering full size I and P frames of said decoded stream of video data commensurate with buffering scaled I, P & B frames of said decoded stream of video data.



57. A method for allocating a frame buffer for a digital video decoding system having video scaling capabilities, said method comprising:

partitioning the frame buffer into multiple memory areas for receiving I, P & B frames of a decoded stream of video data, said partitioning comprising:

defining a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and

defining at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with the first area and the second area receiving the full size I and P frames.

58. A method for allocating a frame buffer for a digital video decoding system having video scaling capabilities, said method comprising:

partitioning the frame buffer into three buffer areas when said digital video decoding system is in a normal video mode, wherein said three buffer areas receive full size I, P & B frames of a decoded stream of video data; and

partitioning said frame buffer into at least three buffer areas when said digital video decoding system is in a scaled video mode, said at least three buffer areas comprising a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and at least one third area for receiving scaled I, P & B frames of said decoded stream of video data.

59. The method of claim 58, wherein said at least one third area comprises at least a third area, a fourth area and a fifth area for receiving said scaled I, P & B frames of said decoded stream of video data, wherein said at least third area, fourth area and fifth area each comprise less area than said first area or said second area.

60. An article of manufacture comprising:

a computer program product comprising a computer usable medium having computer readable program code means therein for use in processing an encoded stream of video data employing a digital video decoding system, said computer readable program code means in said computer program product comprising:

computer readable program code means for causing a computer to effect decoding the encoded stream of video data to produce a decoded stream of video data;

computer readable program code means for causing a computer to effect scaling the decoded stream of video data when the digital video decoding system is in a scaled video mode, said scaling producing a scaled decoded stream of video data; and

computer readable program code means for causing a computer to effect buffering said scaled decoded stream of video data in a frame buffer to await display thereof.

61. An article of manufacture comprising:

a computer program product comprising a computer usable medium having computer readable program code means therein for use in processing an encoded stream of video data employing a digital video decoding system having a normal video mode and a scaled video mode, said computer readable program code means in said computer program product comprising:

computer readable program code means for causing a computer to effect decoding the encoded stream of video data to produce a decoded stream of video data;

computer readable program code means for causing a computer to effect scaling the decoded stream of video data prior to storage thereof in a frame buffer when said digital video decoding system is in said scaled video mode; and

computer readable program code means for causing a computer to effect switching between said normal video mode and said scaled video mode without perceptual degradation of a video display of said decoded stream of video data.

* C L M S *

62. An article of manufacture comprising:

a computer program product comprising a computer usable medium having computer readable program code means therein for use in processing an encoded stream of video data employing a digital video decoding system having a normal video mode and a scaled video mode, said computer readable program code means in said computer program product comprising:

computer readable program code means for causing a computer to effect decoding the encoded stream of video data to produce a decoded stream of video data;

computer readable program code means for causing a computer to effect buffering the decoded stream of video data;

computer readable program code means for causing a computer to effect displaying the buffered decoded stream of video data; and

computer readable program code means for causing a computer to effect controlling said displaying and said decoding such that there exists a first latency between decoding time and displaying time for a frame when said digital video decoding system is in said normal video mode and a second latency between decoding time and display time for a frame when said digital video decoding system is in said scaled video mode.

63. An article of manufacture comprising:

a computer program product comprising a computer usable medium having computer readable program code means therein for allocating a frame buffer for a digital video decoding system having video scaling capabilities, said computer readable program code means in said computer program product comprising:

computer readable program code means for causing a computer to effect partitioning the frame buffer into multiple memory areas for receiving I, P & B frames of a decoded stream of video data, said computer readable program code means for causing a computer to effect partitioning comprising:

(i) computer readable program code means for causing a computer to effect defining a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and

(ii) computer readable program code means for causing a computer to effect defining at least one third area for receiving scaled I, P & B frames of said decoded stream of video data commensurate with said first area and said second area receiving the full size I and P frames.

64. An article of manufacture comprising:

a computer program product comprising a computer usable medium having computer readable program code means therein for use in allocating a frame buffer for a digital video decoding system having video scaling capabilities, said computer readable program code means in said computer program product comprising:

computer readable program code means for causing a computer to effect partitioning the frame buffer into three buffer areas when said digital video decoding system is in a normal video mode, wherein said three buffer areas receive full size I, P & B frames of a decoded stream of video data; and

computer readable program code means for causing a computer to effect partitioning said frame buffer into at least three buffer areas when said digital video decoding system is in a scaled video mode, said at least three buffer areas comprising a first area and a second area for receiving full size I and P frames of said decoded stream of video data, and at least one third area for receiving scaled I, P & B frames of said decoded stream of video data.