NAME

Mpeg1 – MPEG decompressor for compressed image sequences

DESCRIPTION

The Moving Picture Experts Group (MPEG), an ISO technical committee, has developed a general-purpose international standard for the compression of full motion video to a bit rate of 1.5 Mbits/second. The method employs transform coding, specifically the Discrete Cosine Transform (DCT), to obtain intraframe compression by reducing spatial redundancy, and motion compensation to obtain interframe compression by reducing temporal redundancy.

The XIL implementation supports the basic MPEG specification for video compression, but does not address audio and synchronization issues. Certain combinations of XIL operations are accelerated. These combinations should be used for the highest performance in MPEG decompression. For more information and example programs, see the *XIL Programmer's Guide.*

For a bitstream with B frames, the behavior of the **xil_cis_get_bits_ptr**(3) function differs from its usual behavior. For more information, see the discussion of B pictures in the *XIL Programmer's Guide.*

The current release of the XIL library does not contain an implementation of an MPEG compressor. Calls to **xil_compress**(3) will produce an error unless a third-party MPEG compressor has been installed. Also, streams with D frames will not be decompressed.

Creating a CIS

To decompress a compressed image sequence (CIS) with the XIL MPEG decompressor, specify "Mpeg1" for the *compressorname* argument in **xil_cis_create**(3).

Getting and Setting MPEG Attributes

Use **xil_cis_get_attribute**(3) and **xil_cis_set_attribute**(3) to get and set MPEG CIS attributes. These attributes are as described in the following sections. Refer to the example section for additional information.

MPEG Compression Attributes

The following paragraphs describe the MPEG CIS attributes available with the XIL library. All structures and enumerations are defined in **xil.h.** Note that all compression attributes are "settable" and "gettable."

Note that if you are setting an attribute and that attribute is a structure, you must pass the address of that structure. If you are getting an attribute, you always pass the address of the attribute. If you are getting a structure attribute, you must pass a pointer to a pointer to the structure, and XIL will set the pointer to the structure. You must free the memory for this structure (using **free** (3C)) when it is no longer needed.

Many of these attributes employ a "null default" (ND) convention under which setting an attribute to zero/null signifies that the compressor is allowed (required) to use a value that is optimal for its purposes. In all cases, the zero/null value would not otherwise be legal. ND attributes are "gettable" in the sense that they will return null/zero if they are so set, but are opaque with regard to the actual default value used by the compressor. In addition, all ND attributes have null/zero as the default.

COMPRESSOR_BITS_PER_SECOND

Description	Controls the output data rate of the MPEG bitstream in bits/second.
Access	set/get
Type	int value
Values	1 - 104,856,800, rounded upward to the nearest multiple of 400.
Default	1,152,000
Notes	Cannot be changed after the first frame is compressed. Should be set to no more than 1,856,000 if a Constrained Parameter bitstream is desired.

COMPRESSOR_INSERT_VIDEO_SEQUENCE_END

Description	Causes a Video Sequence End code to be inserted into the bitstream.
Access	set/get
Type	Xil_boolean
Values	FALSE - no end code inserted. TRUE - end code inserted after each subsequent call to **xil_cis_flush**(3), assuming this attribute value remains TRUE. Inserting the code is done in addition to the normal actions of the flush routine. When set to FALSE, this attribute doesn't affect the normal actions of the flush routine. The library prevents multiple end codes from being written to the same frame.
Default	FALSE
Notes	An MPEG-1 sequence isn't valid without the end code; therefore, the last frame in the sequence must be followed by the code. Since it cannot predict when an application will end a sequence, the MPEG-1 codec reserves the last frame or subgroup of frames in the CIS so the application can write the end code to that reserved frame. The frame or subgroup must be released before it can be retrieved with **xil_cis_get_bits_ptr**(3) or **xil_decompress**(3). This affects the logic used when making these calls, and also affects the logic used with loop continuation conditions that call **xil_cis_has_frame**(3) to control CIS decompression. A frame or subgroup is released under either of two conditions: when it's followed by an end code, or when it's followed by another frame or subgroup. See the *Xil Programmer's Guide* for more information on releasing reserved frames. There can be multiple video-sequence headers associated with one end code, since the header information changes as certain CIS attributes change (within the XIL limitations that there are no width/height changes). In addition, there may be multiple sequences within a bitstream. If frames are compressed into the CIS after the call to **xil_cis_flush** (3),

it's the compressor's responsibility to provide the video-sequence header information per sequence. Before the application changes an attribute that would result in a new sequence header, it must first output an end code for the current sequence.

COMPRESSOR_INTRA_QUANTIZATION_TABLE

Description	Set quantization matrix for I-frame compression.
Access	set/get
Type	Xil_unsigned8[64]
Values	1 - 255
Default	ND = null
Notes	Set by passing a pointer to an 8x8 matrix containing the desired quantization values. The first element in the array must be an 8. A null pointer sets the null default. Get returns a pointer to a matrix containing the quantization values. A null pointer indicates the null default.

COMPRESSOR_NON_INTRA_QUANTIZATION_TABLE

Description	Set quantization matrix for non-I-frame compression.
Access	set/get
Type	Xil_unsigned8[64]
Values	1 - 255
Default	ND = null
Notes	Set by passing a pointer to an 8x8 matrix containing the desired quantization values. A null pointer sets the null default. Get returns a pointer to a matrix containing the quantization values. A null pointer indicates the null default.

COMPRESSOR_PATTERN

Description	A structure containing a string of length greater than 0 and an integer repeat count. The string sets the pattern of picture types (in display order) which will be employed by the compressor in all subsequent groups of pictures (GOPs). The repeat count determines the number of times this pattern occurs in a GOP; i.e., the number of pictures in a GOP is the length of the pattern string multiplied by the repeat count. However, if the COMPRESSOR_PATTERN attribute is reset, if a new quantization table is loaded via the COMPRESSOR_INTRA_QUANTIZATION_TABLE attribute or the COMPRESSOR_NON_INTRA_QUANTIZATION_TABLE attribute, or if the COMPRESSOR_INSERT_VIDEO_SEQUENCE_END attribute is set, the current GOP is terminated, and a new GOP is started on the next frame with a picture pattern that is synchronized with the beginning of

the current pattern string.

Access set/get

Type

```
typedef struct __XilMpeg1Pattern {
  char*              pattern;
  Xil_unsigned32     repeat_count; } XilMpeg1Pattern;
```

Values The pattern string must contain only the characters 'I', 'B', 'P', and 'D', which indicate Intra, Predicted, Bidirectional, and DC pictures. The repeat count must be greater than zero.

Default ND = null

Notes Set by passing a pointer to the pattern structure. A null string sets the null default. Get returns a pointer to the structure. If this is null, the null default is indicated. After a get which does not return null, it is the application's responsibility to free the pattern string and the structure storage.

COMPRESSOR_PEL_ASPECT_RATIO

Description Describes the pixel aspect ratio of the compressed image.

Access set/get

Type

```
typedef enum {
        NullDefault,
        Ratio_1_0,          /* 1.0       */
        Ratio_0_6735,       /* 0.6735    */
        Ratio_0_7031,       /* 0.7031    */
        Ratio_0_7615,       /* 0.7615    */
        Ratio_0_8055,       /* 0.8055    */
        Ratio_0_8437,       /* 0.8437    */
        Ratio_0_8935,       /* 0.8935    */
        Ratio_0_9157,       /* 0.9157    */
        Ratio_0_9815,       /* 0.9815    */
        Ratio_1_0255,       /* 1.0255    */
        Ratio_1_0695,       /* 1.0695    */
        Ratio_1_0950,       /* 1.0950    */
        Ratio_1_1575,       /* 1.1575    */
        Ratio_1_2015        /* 1.2015    */
} XilMpeg1PelAspectRatio;
```

Values The enumeration forms a discrete set of "likely" possibilities defined in the MPEG specification; they vary from .6375 to 1.2015.

Default ND = NullDefault

COMPRESSOR_PICTURE_RATE

Description: Describes the picture rate in frames per second of the image sequence to be compressed.

Access: set/get

Type:

```
typedef enum {
        NullDefault,
        Rate_23_976,        /* 23.976     */
        Rate_24,            /* 24.0       */
        Rate_25,            /* 25.0       */
        Rate_29_97,         /* 29.97      */
        Rate_30,            /* 30.0       */
        Rate_50,            /* 50.0       */
        Rate_59_94,         /* 59.94      */
        Rate_60             /* 60.0       */
} XilMpeg1PictureRate;
```

Values: The enumeration forms a discrete set corresponding to commonly available sources of digital or analog video, varying from 23.96 to 60.0.

Default: ND = NullDefault

COMPRESSOR_SLICES_PER_PICTURE

Description: Provide a suggestion to the compressor on how many slices to use in each picture.

Access: set/get

Type: int value

Values: 1 - number of macroblocks in the picture

Default: ND = 0

Notes — Although the compressor is free to ignore this suggestion, setting this attribute to a high value may result in an inefficient use of the available bit rate.

COMPRESSOR_TIME_CODE

Description — A time code that applies to the first picture (in display order) in the group of pictures (GOP) to be encoded. It is included to provide video time identification to applications.

Access — set/get

Type

```
typedef struct __XilMpeg1TimeCode {
        Xil_boolean      drop_frame_flag;
        Xil_unsigned32   hours;
        Xil_unsigned32   minutes;
        Xil_unsigned32   seconds;
        Xil_unsigned32   pictures;
} XilMpeg1TimeCode;
```

Values — The time code structure contains fields with integer values: hours (0-23), minutes (0-59), seconds (0-59), and picture number (0-59).

Default — ND = null

Notes — Set by passing a pointer to the time code structure. A null pointer sets the null default. Get returns a pointer to a structure containing the time information or null if the null default is set.

MPEG Decompression Attributes

DECOMPRESSOR_QUALITY

Description — Provide a suggestion to the decompressor, enabling it to trade off reconstruction quality in exchange for an increase in decoding speed.

Access — set/get

Type — int value

Values — Value can be between 1 and 100. A value of 100 sets the quality to maximum. A value of 1 sets the speed to its maximum and allows the quality to decrease to the minimum allowed by the decompressor. The decompressor is free to ignore this suggestion.

Default — 100

Notes — The MPEG decompressor may increase speed by such devices as decreasing the number of quantized coefficients that it uses in reconstruction, rounding motion vectors to integer values, etc.

DECOMPRESSOR_BROKEN_LINK

Description Describes whether the B-pictures that precede the first I-picture in the GOP can be correctly decoded.

Access get

Type Xil_boolean

Values FALSE - can be correctly decoded; TRUE - cannot be correctly decoded.

Default FALSE

Notes If this attribute is set to TRUE, it implies that the I or P picture from the previous group required to form the predictions is not available (presumably because it was removed as part of an editing process).

DECOMPRESSOR_CLOSED_GOP

Description Describes whether the group of pictures is open or closed.

Access get

Type Xil_boolean

Values FALSE - open group; TRUE - closed group.

Default None

Notes Closed groups can be decoded without using decoded pictures of the previous group for motion compensation. Open groups require such pictures to be available.

DECOMPRESSOR_FRAME_TYPE

Description Gives the picture type of the current picture in the group.

Access get

Type

```
typedef enum {
        I, P, B, D
}XilMpeg1FrameType
```

Values Values of the enumerated type are I, P, B, and D.

Default None

Notes The values 'I', 'B', 'P',and 'D' indicate Intra, Predicted, Bidirectional, and DC pictures.

DECOMPRESSOR_PEL_ASPECT_RATIO_VALUE

Description Describes the pixel aspect ratio of the decompressed image.

Access get

Type	float value
Values	The set of possible values forms a discrete set of "likely" possibilities defined in the MPEG specification; they vary from .6375 to 1.2015.
Default	1.0

DECOMPRESSOR_PICTURE_RATE_VALUE

Description	Describes the picture rate of the MPEG encoded image sequence in frames per second.
Access	get
Type	float value
Values	The set of possible values forms a discrete set corresponding to commonly available sources of digital or analog video, varying from 23.96 to 60.0.
Default	None

DECOMPRESSOR_TEMPORAL_REFERENCE

Description	Gives the number in the temporal reference field of the current picture in the group.
Access	get
Type	int value
Values	Between 0 and 1023
Default	None
Notes	This may be useful, because MPEG pictures are not transmitted in display order, but rather in the order in which the decoder needs to decode them.

DECOMPRESSOR_TIME_CODE

Description	A time code that applies to the first picture (in display order) in a group of pictures (GOP). It is included to provide video time identification to applications.
Access	get
Type	

```
typedef struct __XilMpeg1TimeCode {
        Xil_boolean       drop_frame_flag;
        Xil_unsigned32    hours;
        Xil_unsigned32    minutes;
        Xil_unsigned32    seconds;
        Xil_unsigned32    pictures;
} XilMpeg1TimeCode;
```

Values — The time code structure contains fields with integer values: hours (0-23), minutes (0-59), seconds (0-59), and picture number (0-59).

Default — None

EXAMPLES

The following example opens and closes an MPEG CIS using the XIL library.

```
XilSystemState State;
XilCis cis;
State = xil_open();
cis = xil_cis_create(State, "Mpeg1");

-- calls to MPEG-specific compression routines --

xil_cis_destroy(cis);
xil_close(State);
```

The following example sets an MPEG CIS attribute called *COMPRESSOR_SLICES_PER_PICTURE* to 3. Note that because this attribute is not a structure, it is not necessary to pass the address of this attribute when setting it.

```
XilCis cis;

xil_cis_set_attribute(cis,"COMPRESSOR_SLICES_PER_PICTURE", (void *) 3);
```

The following example returns the value of an MPEG CIS attribute called *COMPRESSOR_SLICES_PER_PICTURE.* Note that when getting an attribute, it is always necessary to pass the address.

```
Xil_unsigned32 slices;
XilCis cis;

xil_cis_get_attribute(cis, "COMPRESSOR_SLICES_PER_PICTURE", (void **) &slices);
```

NOTES

The **xil_cis_set_attribute**(3) and **xil_cis_get_attribute**(3) calls are used to modify the default behavior of a specific compressor. Generic attributes of compressors are set by individual function calls.

SEE ALSO

xil_cis_create(3), **xil_cis_get_attribute**(3), **xil_cis_get_bits_ptr**(3), **xil_compress**(3), **xil_decompress**(3), **xil_cis_has_frame**(3).