NAME

CellB – XIL driver for CellB video compression/decompression

DESCRIPTION

CellB is a video compression format based on the techniques of block truncation coding and vector quantization. It is well suited for video conferencing. Even though it uses interframe compression, it guarantees that all cells are intraframe encoded periodically, allowing for dropped frames.

Creating a CellB CIS

To compress a compressed image sequence (CIS) with the XIL CellB compressor, specify "CellB" for the *compressorname* argument in **xil_cis_create**(3).

Getting and Setting CellB Attributes

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

CellB Attributes

The following paragraphs describe the CellB CIS attributes available with the XIL library. All structures and enumerations are defined via **xil.h.** Note that some attributes are "set-only" and others are "get-only." This is noted under the *Access* heading for each attribute.

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 its address.

WIDTH

Description: Sets the frame width of the encoded bitstream. It is only necessary to set this attribute to decompress a bitstream that has been input via a call to **xil_cis_put_bits**(3) or **xil_cis_put_bits_ptr**(3).

Access: set

Type: integer

HEIGHT

Description: Sets the frame height of the encoded bitstream. It is only necessary to set this attribute to decompress a bitstream that has been input via a call to **xil_cis_put_bits**(3) or **xil_cis_put_bits_ptr**(3).

Access: set

Type: integer

IGNORE_HISTORY

Description: CellB bitstreams do not contain any frames containing all the information necessary to reproduce the entire image. In general, this means that these bitstreams are not randomly seekable, because it is expensive to back up far enough so that all cells/macroblocks can be properly decoded for the frame you want to seek to.

By setting *IGNORE_HISTORY* to TRUE, you can build an application that compresses images and decompresses them immediately for display. Setting this attribute to TRUE also allows faster seeking for applications that need to move forward and backward in the bitstream.

Values	*FALSE:* the decoder sets the *RandomAccess* attribute of such CISs to FALSE (i.e., **xil_cis_get_random_access**(3) returns FALSE), and it becomes impossible to seek backwards. Also, seeks forward will actually decode all intermediate frames, instead of just jumping to the appropriate location and decoding the sought frame.

TRUE: (i.e., **xil_cis_get_random_access**(3) returns TRUE), seeking backwards is possible, and forward seeks may not decode the intermediate frames. After an *IGNORE_HISTORY* seek, the decoded picture may have some bad cells (macroblocks). As these are encoded in subsequent frames, these will "twinkle" in.

Type	Boolean

Access	set/get

Default	FALSE

ERRORS

For a complete list of XIL error messages by number, consult Appendix B of the *XIL Programmer's Guide.*

EXAMPLES

The following example opens and closes a CellB CIS using the XIL library:

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

-- calls to CellB-specific compression routines --

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

NOTE

The CellB bitstream definition (unlike the one for **H261**(3)) does not define a maximum number of frames before a cell must be encoded in the bitstream. However, the encoder that comes with the XIL library does enforce this behavior.

SEE ALSO

xil_cis_get_attribute(3), **xil_cis_create**(3), **xil_cis_put_bits**(3), **xil_cis_put_bits_ptr**(3), **xil_cis_get_bits_ptr**(3), **xil_compress**(3), **xil_decompress**(3).