USES-32C SPECIFICATION VERSION 1.10

ICs, LLC
P.O. Box 2236
2040 Warren Wagon Road
McCall, ID 83638
Technical contact: Lowell Miles <Lowell.Miles@ics-radhard.com>

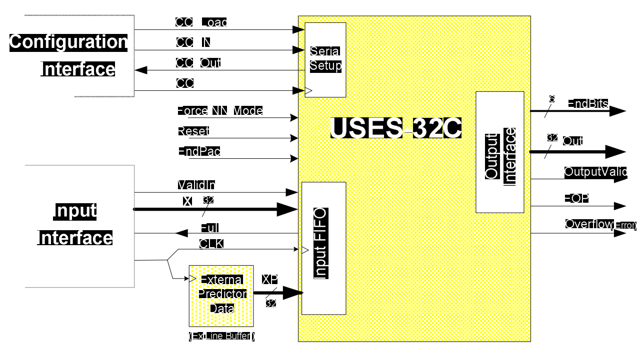
February 24, 2012

Features

- Lossless Compression up to 200Msamples/sec with current FPGA technology.
- Conformant to CCSDS *Lossless Data Compression* standard *CCSDS 121.0-B*
  - 31 Rice Algorithm coding options plus one low entropy option for compression of thresholded or low entropy data.
- Shorter code ID when N ≤ 4, for improved performance in compressing feature map, cloud maps. .
- Input specifications:
  - Quantization supported: N = {1…32}, Data format supported: positive or two's complement.
  - Selectable code block sizes (J) from 4 to 64 samples, padding to a full block supported.
  - Continuous input data stream as well as pauses in data flow between blocks.
- Output Specifications:
  - Parallel 32 bit output.
  - Output data concatenated into data packets with end-of-packet signal; Continuous packet mode and early-ending mode supported.
- Operating Modes:
  - Nearest Neighbor prediction.
  - Simple 2D prediction.
  - External prediction.
  - Entropy-coding mode: entropy coding only bypassing pre-processor.
  - Bypass mode: no compression performed, raw input data packed to a 32 bit output.
- Implementable in a Rad-Hard Field Programmable Gate-Array.
- FPGA Performance:
  - Altera Cyclone IV E (EP4CE115F29C8): throughput achieved at greater than 50 Msamples/sec with 13,603 logic elements (12%), 6,393 registers (6%), and 213,632 memory bits (5%);
  - Stratix III (EP3SE50F484C2): throughput achieved at 200 Msamples/sec with 6,255 combinational ALUTs (16%), 3,383 dedicated registers (9%), and 16,664 memory bits (<1%).

Applications

- Reduction of Science Instrument Data
- Reduction of Satellite RF communication bandwidth requirement
- Reduced storage requirement for ground science data archive



Interface Signal Diagram of USES-32C

Table of Contents

1.0 GENERAL DESCRIPTION 3
2.0 CIRCUIT OPERATION 4
  2.1 Data Flow 6
  2.2 Initialization 6
  2.3 Setting Quantization (Bits per sample) 8
  2.4 Compressing Two's Complement Data 9
  2.5 Setting Block Size J (Samples per block) 9
  2.6 Setting Blocks per Reference 9
  2.7 Managing Insufficient Input Data for a Data line 10
  2.8 Defining Data Packets 10
    2.8.1 Setting References per Packet (RefPac) 11
    2.8.2 Continuous Packets (ConPac) 11
    2.8.3 Ending a Packet Early (EndPac) 12
  2.9 Bypass Mode (Bypass) 12
  2.10 Prediction Modes (PreMod) 12
  2.11 Entropy Coding (EntCode) 13
3.0 TYPICAL OPERATION 13
4.0 DATA FORMATS 14
  4.1 Input Samples 14
  4.2 Output Format 14
  4.3 Coding Examples 16
5.0 TIMING DIAGRAM 19
6.0 REFERENCES 20

1.0 GENERAL DESCRIPTION

USES-32C (32-bit Universal Source Encoder for Space) accepts sampled data and encodes the data using a lossless algorithm. Input data may be two's complement or positive and quantized within the range N = {1, …, 32} bits. Maximum quantization is 32 bits/sample.

The sampled data may, for example, be imagery, digital audio, data from transform bins, or data from scientific measurements. The chip operates on small blocks of data rather than an arbitrary number of data samples. The block value J must be set to be between 4 and 64 samples per block. If a scan line does not end in an integral number of J samples, the final input block must be filled by setting the fill samples configuration variable.

USES-32C has two major modes of operation: compression or bypass. In compression mode the entropy coder is used to code input data, while the pre-processor may or may not be used. The coding may occur after the data is pre-processed using prediction modes, or input data may be routed around the pre-processor so that it goes directly to the entropy coder of the chip. The second major operating mode is the bypass mode, in which no compression or insertion of bits is performed: the chip functions merely as a packetizer forming data into 32 bit words.

When in compression mode, USES-32C supports 3 different types of prediction methods:

1. Nearest neighbor prediction.: Predictor is previous sample.
2. Simple 2D prediction: Predictor is average of adjacent X and Y samples.
3. External prediction: Predictor is externally supplied, used for prediction in Y and Z dimensions.

USES-32C compresses image data by encoding differences between a predicted sample and the present sample. A simple block diagram of a Rice algorithm based encoder is shown in Figure 1. Differences are mapped to a positive integer before being sent to the entropy coder. Because differences are coded, a reference sample must be included at the start of a set of blocks of data; typically, at the start of every scanline.

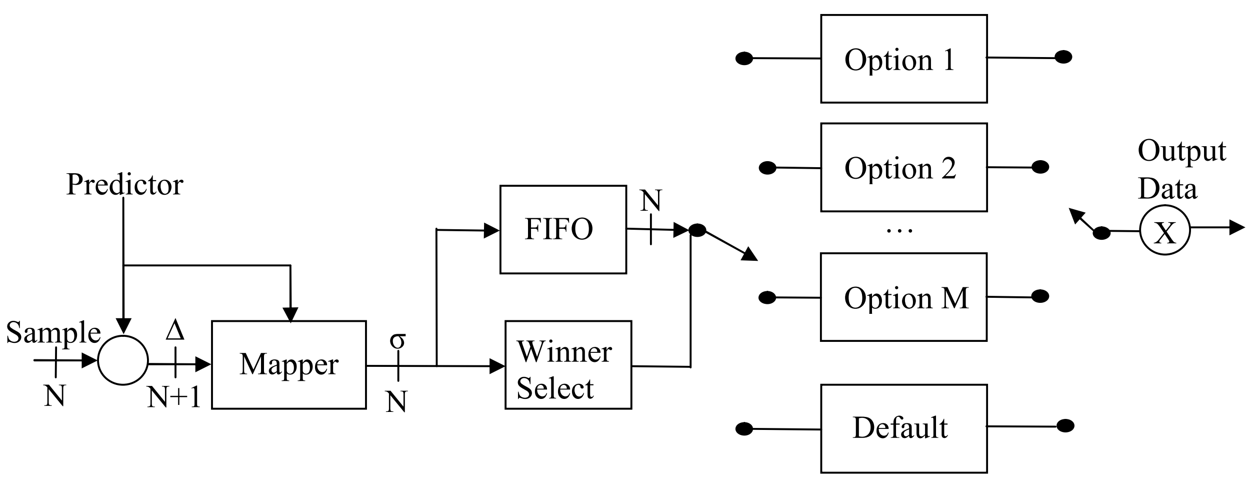


Figure 1: Block Diagram of Rice Algorithm Architecture.

The output of the mapper is referred to as a “sigma value”. Each sigma value is sent to the entropy coder. The entropy coder first calculates which of the compression options has the best compression-- and then encodes the winner. The choice for the winning compression option is based on exact counts, not on estimates for the winning option. Exact counts guarantee USES-32C will always choose the winning option that provides maximum compression. When multiple coding options result in the same lowest number of bits, the default option is chosen first, otherwise the option associated with the lowest value of the ID is chosen. Compressed data is output on a 32-bit bus.

The entropy coder performs a 31 option Rice Algorithm coding set with the addition of a low entropy coding option. For an explanation of the algorithm refer to [1]. Description of previous work and the performance of the earlier version of this chip is given in [2]. The 32 coder options can be divided into 4 major types. The simplest is the default option, which sends uncompressed sigma data to the output. The default option limits the worst case coded length to be equal to the sum of the number of input bits plus the ID bits.

The second coder type is the fundamental sequence (FS). It codes sigma values so the encoded word length is equal to the sigma value plus one bit. Large differences between adjacent samples result in large encoded words while small differences are coded with small words. Coding examples using the FS and default options are given in Section 4.3.

The third option type uses FS coding with sample-splitting. A one bit sample-split, k=1, would have the least significant bit of the sigma value stripped off and saved for every sample in the block. The remaining N-k bits are then FS coded. The FS encoded data is sent before the least-significant bits, which are concatenated. The option set includes using 29 sample-split options k = {1…29} combined with the FS option (which is equivalent to k=0 case). In Figure 1 what is labeled "Option M" corresponds to the maximum k split option used in the USES-32C design: k = 29 bits/sample.

The fourth option type includes two techniques for coding low entropy data. One coder is designed for zero entropy conditions such as might be found in thresholded data. This zero option keeps track of blocks that are runs of zeros. The EOL condition (end of data line) is set as the 5th most likely event, following runs of 1…4 blocks of zeros. Therefore, an entire line of unchanging data may be coded in as few as 9 bits. The second low-entropy coding technique uses a special mapping algorithm which assigns a new non-negative integer value to every pair of sigma values. The assignment is such that more frequently occurring sigmas are given a smaller value which will then be FS coded. The pairing and the mapping results in reducing total coding bits. This technique is effective for input data entropy below 1.5 bits/sample.

2.0 CIRCUIT OPERATION

USES-32C may operate as a data packetizer or as a compressor on sampled data quantized from N = {1…32} bits/sample. It is designed to input data in blocks that are made up of a continuous string of samples. A block is made up of J samples (J = 4, …, 64) . Note that the CCSDS 121B recommendation only specifies J=8, 16, 32 and 64. If a line of data ends on a sample count that is not an integral multiple of J, the "fill samples configuration variable", discussed in sub-section 2.7 can be used to fill out the remainder of the line by internally filling sigma values of zero.

Blocks of data are grouped into larger blocks of data by the BlkRef configuration variable. For images, BlkRef is typically set to be equal to the number of blocks in a scanline. In modes in which a reference sample is saved, BlkRef determines how many blocks of data must be input before inserting another reference sample. Because there may, at times, be more than 1 reference per scanline, or the data may not be from an image, the term "dataline" will refer to the input data set in which the number of samples is defined by the product of J and BlkRef.

Datalines are grouped into packets by specifying the RefPac configuration variable. The product of J, BlkRef, and RefPac define the number of samples in a data packet. When the term "packet" is used in this specification, it simply refers to data that is concatenated and output as 32 bit words. The final word of a packet may contain fill (non-data bits) to fill out the output bits to a word boundary.

An input/output interface diagram of USES-32C is shown in Figure 2.

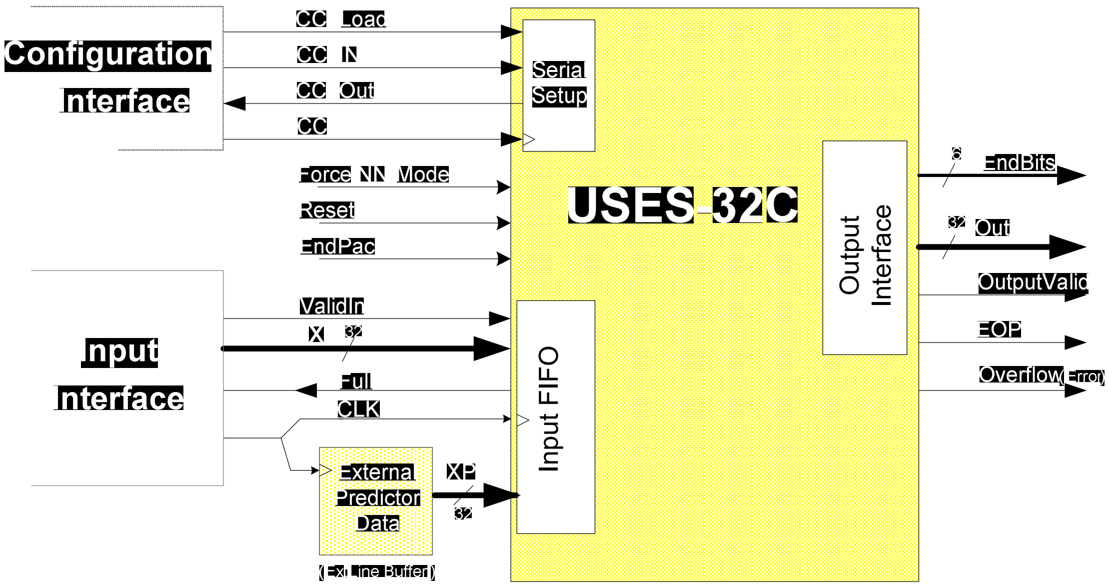


Figure 2: Input/Output Interface Signal Diagram of USES-32C.

The interface signal list is provided in Table 1.

Name	Type	Function
Clk	Input	System clock for USES-32C.
Reset	Input	System reset. Initializes circuit.
Force-NN-Mode	Input	Forces nearest neighbor prediction mode.
EndPac	Input	Will end packet in any mode. (Active High)
ValidIn	Input	Valid Block, high when valid data is input.
X[31:0]	Input	Sample input data bus.
XP[31:0]	Input	Predictor input bus.
CC	Input	Configuration clock.
CC_In	Input	Input for serial configuration data.
CC_Load	Input	When high the state of the internal configuration data is serially output on CC_Out.
Out[31:0]	Output	Output bus.
OutputValid	Output	Signals data is at the output when high.
EOP	Output	Data at output is end of packet. (Active High).
EndBits[5:0]	Output	Number of data bits in last word of packet. 1.. 32.
CC-Out	Output	Configuration data serial output.
Full	Output	USES-32C full of data. ValidIn signal must go low for one cycle with a latency of at most 8 cycles.
OverFlow	Output	USES-32C input fifo overflowed. Once this bit is set it will not be reset until the chip is reset. If OverFlow bit set then the data output from the chip will be corrupted.

Table 1: USES-32C Interface Signal List

The Clk and CC clock signals are independent signals with no relationships assumed. Signal input and output data delays will be referenced to the rising edge of Clk. Sample data will be input on every rising edge of Clk, resulting in a maximum input sample rate equal to the clock rate. All output signals and data are clocked on the rising edge of Clk, resulting in a maximum output transfer rate equal to the clock rate. Configuration input and output signals (CC_Load, CC_In, CC_Out, and Reset) will be similarly referenced to CC.

All inputs must be actively driven to a valid low or high. **No inputs may be left floating**. Pertinent control timing diagrams for the USES-32C are in Section 8 of this document.

2.1 Data Flow

Whenever valid data is to be received by USES-32C the ValidIn (Valid Input) signal must be high. After initialization, when using quantization near 32 and short block sizes along with low compression, it is possible that the input to the chip will need to stall the input stream. This is done by checking the "full" signal. When the full signal goes high after being low for two or more clock cycles, the USES-32C core's input FIFO has room for 8 more samples. The full signal can be used to determine if data is sent to the chip pipelined up to 8 cycles in the future. The Overflow signal will go high and stay high after the input FIFO has overflowed until the USES-32C core is reset.

While USES-32C is waiting for ValidIn to go high, the internal pipeline continues to process and output data. If the data has stopped at the end of a packet (or EndPac has been pulsed) all of the data will "drain out" of the chip's pipeline. Bits can be flushed out by using the EndPac signal to end a packet early as discussed in Section 2.8.3.

After reset has been brought low, the earliest ValidIn signal may be brought high is 8 CC cycles after the last configuration bit has been clocked in. Data will then be accepted and processed every Clk period. After the block of data is evaluated, the winning option is encoded, formatted and output on a 32 bit data bus. Any bits that do not make up a full 32 bit word width, will be saved and output with the start of the next block's data. The remainder data from the previous block will always be output in the most significant bit positions.

There will be no handshaking at the output. At the start of a packet the OutputValid signal will accompany the first data word by going high. OutputValid is true only when valid data is at the output. In the course of outputting a single block of data, the OutputValid signal may toggle high and low-- data will not be sent out in a contiguous fashion. The external system that receives output data must be ready to receive data any time OutputValid signal is active. All data transfers take 1 clock period. Between blocks, USES-32C will hold any remainder bits from the previous block in order to concatenate the data into a data packet. Only at the end of a packet will remainder bits be output. At the end of a packet the EOP signal and OutputValid signal will be high. When EOP is asserted, the EndBits bus will specify how many valid bits are in the last word of the packet. EndBits bus provides a binary count in the range 1, …, 32 bits. Figure 13 in Section 8 shows waveforms for an end-of-packet condition.

Latency is the number of clock cycles required from the time the data is at the input of USES-32C to the time data is ready at the output and OutputValid is high. The minimum latency of USES-32C is fixed in all operating modes but depends on block size. When a packet has ended, the maximum latency is less than 256 clock cycles for all modes. The maximum latency when a packet has not ended, is dependent on mode, quantization and data. The lowest entropy data will give the largest latency. It is possible to have no output from the chip until two data lines have been coded, or until 896 blocks have been coded, or until an end-of-packet has been received, whichever comes first. This long latency is due to the very high compression of zero entropy data.

2.2 Initialization

Before proper operation can begin, USES-32C must be initialized. This is accomplished by holding Reset high for a minimum of 5 clock cycles and keeping ValidIn low for the configuration window. This sets USES-32C into a known state. Figure 3 shows the general initialization window. Reset must be high for a minimum of 5 clock cycles. ValidIn must be low while reset is low during the initialization window and 8 cycles after the initialization window. Configuration data is then clocked in on the rising edge of CC. Once all the configuration data is clocked in initialization is complete. Figure 4 shows data input during the configuration window in more detail. Figure 5 shows configuration data being read out.

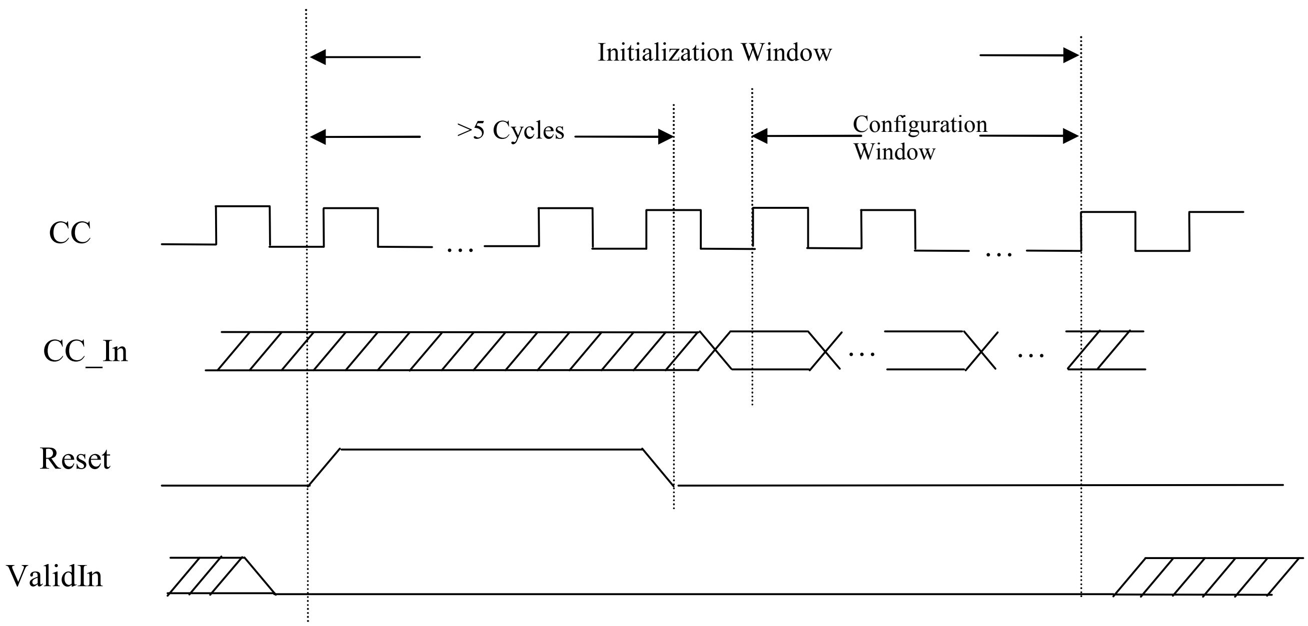


Figure 3: Initialization window.

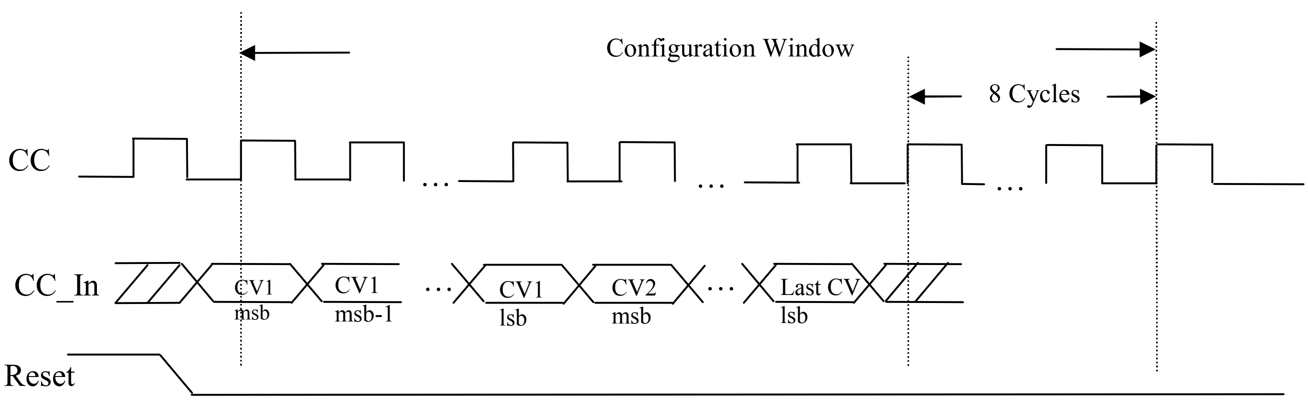


Figure 4: Configuration window in input mode.

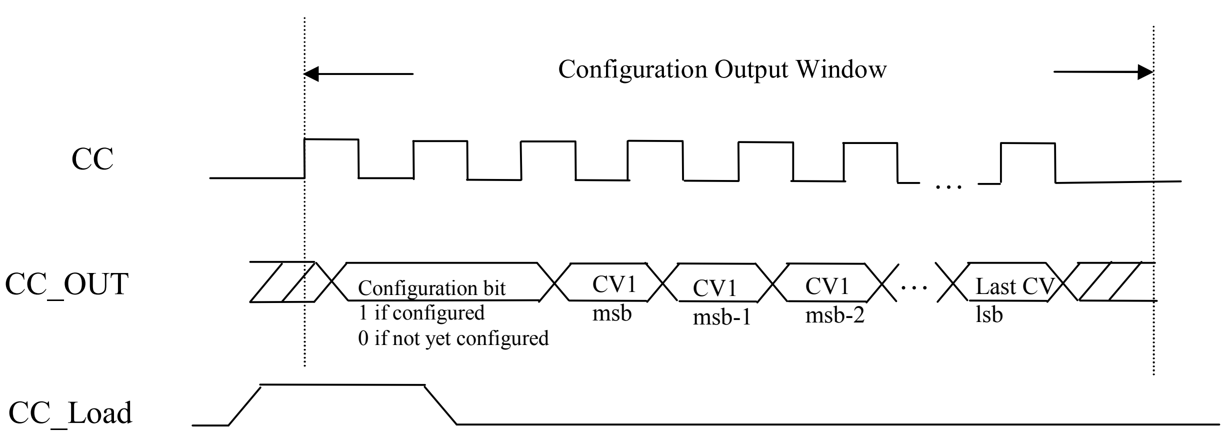


Figure 5: Reading stored configuration variables.

Once initialized, USES-32C must be reset to re-initialize to operate in a different configuration. After initialization, the output signal OutputValid will remain low until the first data word is at the output. The EOP signals will remain low until the end of the first packet. The EndBits signals will only be valid when EOP is active, at other times the count will be invalid and the outputs will not necessarily be low.

USES-32C is setup by serially loading a 48-bit long configuration setup stream described in Table 2. Bit 0 is the first bit loaded into the core.

USES-32 C Serial Configuration Settings			
Bit Range	Field Name	Width	Description
0-6	N	7 bits	Input data quantization in bits-per-sample, allowed range 1-32.
7- 19	BlkRef	13 bits	Blocks per reference, 13 bits wide. Allowed range 1-4096
20	Bypass	1 bit	Bypass mode: the chip will act as a packetizer. Active high. When selected, all other modes will be ignored (including EntCode).
21	ConPac	1 bit	Continuous-Packet-Mode: valid only during initialization. If selected, RefPac will be ignored.
22	EntCode	1 bit	Entropy coding of positive data. Valid only during initialization. Active high.
23-28	Fill-Samples	6 bits	Number of samples to fill at end of scanline. The range is 0-63
29-30	PreMod	2 bits	The prediction mode : 00 Nearest-neighbor 01 Simple 2D 10 External Prediction 11 Reserved
31-37	J	7 bits	Block size in Samples-Per-Block, 7 bits wide, allowed range 4-64.
38	Positive	1 bit	Positive when high: Allowing inputs to be positive or bipolar data (in the form of two's complement format). Valid only during initialization.
39-46	RefPac	8 bits	References-per-packet allowed range: 1-128
47	USES15-Mode	1 bit	Operate as legacy USES 16 bit chip. Active High

Table 2: USES-32C Serial Configuration Variables

OTHER SIGNALS
- ValidIn. Valid input data on input. Must be low during initialization.
- When ValidIn is low, the control signals EndPac and Force-NN-Mode are ignored.

Configuration data on the CC_In pin loads the configuration data serially into the chip. Each configuration variable is shifted in on the CC_In pin with the most significant bit first and with the number of bits detailed above. After all configuration bits have been shifted into the chip, the configuration clock (CC) may be turned off after 8 cycles.

The configuration data may be read out by bringing the CC_Load signal high to load the configuration data into the output shift register. Each CC clock cycle CC_Load is low the next bit of configuration data is output on the CC_Out pin. The configuration data is shifted out in the same order as it is shifted in. Shifting out the configuration data does not affect the operation of the USES-32C chip.

2.3 Setting Quantization (Bits per sample)

The quantization range supported is N=1, ..., 32. The binary value of bits per sample sets the value of N. For example, to set a quantization of ten, the configuration variable must have the bit pattern “001010” at the input. Unused most-significant bits of the two input buses X and XP, will be ignored by the chip. Even the ignored inputs, like all other inputs, must be driven with a valid high or low voltage level.

2.4 Compressing Two's Complement Data

USES-32C will compress positive and two's complement data. When using entropy coding, only positive data can be used. When quantization is specified (as defined in the Initialization Section) and the Positive configuration variable is set low during initialization, USES-32C will interpret the Nth bit as the sign bit. Internally USES-32C works using positive data by normalizing the two's complement values. For example, if the quantization (including sign bit) is 10 bits, initialize USES-32C to two's complement data with 10 bit quantization. The 10th bit will be interpreted as the sign bit. The dynamic range for an N bit two's complement number is -2^(N-1),..,-1,0,1,..,2^{N-1}-1. For a 10 bit positive number the dynamic range is 0…1023; for a 10 bit two's complement number the dynamic range is -512,..,511.

In Bypass mode, there is no difference in positive and two's complement data, only quantization matters.

2.5 Setting Block Size J (Samples per block)

USES-32C compresses blocks of data rather than a continuous data stream. USES-32C supports block sizes: J= 4, …, 64 samples per block. The small block sizes allow for rapid adaptation to image statistics. Typically J=16 produces the best results, but in some instances of rapidly varying statistics J=8 will produce better compression. Those instances may include data from DCT frequency bins as well as data in which there are pockets of very high entropy surrounded by pockets of low entropy The Samples-per-Block configuration variable specifies the block size.

The number of samples in a data line must be an integral multiple of the block size, which is set at J samples. If the number of samples in a data line is not a multiple of J, the line must be filled out using the Fill-Samples configuration value as described in Section 2.7. The maximum number of coded bits per block occurs when the quantization level N is 32, J=64, and the default option wins, resulting in 32*64+5 bits. The maximum block size for a given quantization and block size is given by the following equation:

*If (N>16) Max_Block_Size = Jsamples/block * Nbits/sample + 5IDbits*
*Else if (N>8) Max_Block_Size = Jsamples/block * Nbits/sample + 4IDbits*
*Else if (N>4) Max_Block_Size = Jsamples/block * Nbits/sample + 3IDbits*
*Else if (N>2) Max_Block_Size = Jsamples/block * Nbits/sample + 2IDbits*
*else Max_Block_Size = Jsamples/block * Nbits/sample + 1IDbits*

Equation 1

The maximum block size remains constant even when reference bits are inserted. References have the same number of bits as default data. When references are inserted, only J-1 sigma values follow in the block. When references are not inserted, J sigma values make up a block. In both cases, J maximum sized words (reference or sigma) make up the block-- keeping the maximum block size constant.

2.6 Setting Blocks per Reference

The BlkRef configuration variable must be set in all operating modes, even in the modes in which references aren't inserted into the data stream: External prediction, Bypass, and Entropy coding modes. In the three "non-reference" modes, BlkRef simply becomes an argument to Equation 1 that determines packet size as well as used in the zero-block option [1]. In all other modes, however, not only is BlkRef an argument in determining packet size but it is used to define how often a reference is inserted into the data stream.

A reference is an N bit word of raw sample data upon which succeeding sample differences are based upon. References are inserted when nearest neighbor or 2D mode is selected. When the External prediction option is chosen, the external system must provide all predictor values and references. References are needed by the Decoder to recover all the differenced values. For images, BlkRef is typically set to the number of blocks in the image scan line.

With modes other than Bypass, External prediction, or Entropy coding, the user must determine how often to insert references. References are inserted at the start of data lines. In images, references are typically inserted at the start of scan lines, but they may be inserted more often by setting the "BlkRef" configuration variable. The minimum BlkRef value is a 1. If BlkRef were set to 1, a reference is inserted every block; if BlkRef were set to 40, references would be inserted every J*40 samples. As a minimum, references will be inserted every 4096 blocks. The maximum samples between references is 2^18 samples.

If an input scan line of an image is larger than the maximum samples at the given block size, set BlkRef so that the number of samples between references repeats itself in the line an integral number of times. This will make more than one reference occur per scan line, but is of no appreciable effect on compression. For example, if an image scan line contained 40K samples (40960), BlkRef might be set so that a reference was inserted in the middle of the scanline (20480 samples) if J=8. BlkRef would be set to 2560 to allow two references per scan line (J*2560 = 20480). (Because there may be a very slight overhead due to inserting references, it is generally a good idea to set BlkRef to no smaller than the number of blocks in a scan line unless the scan line exceeds 4096 blocks of samples.)

2.7 Managing Insufficient Input Data for a Data line

When there is insufficient samples for a data line, set the configuration variable, Fill-Samples, to the number of samples needed to complete the last block of the scanline. These extra samples will need to be sent into the chip but will be changed internally to produce the best compression.

This paragraph describes filling out a block of data in modes that use the XP bus: 2D or External Prediction. While in the "XP modes", the XP bus is ignored during the fill times. This is done in order to guarantee that the fill is of the lowest possible entropy. The user should only use fill in the XP modes if their application allows them to use one of the two following strategies. First, fill should only be used if it is at the end of every data line. This constraint is due to fact that the decoder will need to know when to ignore the XP bus and repeat the last valid X sample. The future decoder will be designed so that it has a 6 bit bus specifying how many, if any, samples only at the end of each line are to be decoded as repeated samples. The second strategy a user may have is that when blocks are filled out, the decoded data samples will be either don't care values or will be weeded out by some post processing algorithm. Under such a scenario, the decoder will not know the data is fill data and will use whatever is present on the XP bus at the time to decode the samples. The filled out blocks may not be decoded as having repeated X samples-- but that may be acceptable for the application.

In all operating modes (including Bypass or when EndPac is pulsed), all blocks must have J samples.

2.8 Defining Data Packets

The term "packet," as used in this specification, is concatenated output data formatted into 32 bit words. The only time there may be fill bits (non-data bits) in the output data is at the end of a packet, needed for filling out the output word size to 32 bits. In forming packets of data, USES-32C concatenates any remainder bits from one block with the output bits from the next block.

The size of those packets, in terms of samples, is user selectable and is dependent on setting three configuration variables: J, BlkRef, and RefPac. *These three configuration variables must be set in all modes even in the three modes, External Prediction, Bypass, and Entropy coding, in which references are not inserted.* For those three "non-reference" modes, it will be useful to consider Blocks-Per-Reference and References-Per-Packet simply as two multiplying factors to determine packet size as defined in Equation 2.

The block size J must be selected to be J= 4.,.., 64 samples per block. The BlkRef parameter must be entered during initialization to specify how many blocks should be coded between references. Finally, the RefPac parameter must be initialized to determine how many references are to be in a packet. Packet size, in samples, is determined by the following equation:

$$Packet\_size = J * BlkRef * RefPac$$

Equation 2

The smallest number of bits in a packet occurs in entropy coding or external predictor modes (which include no references in the data stream) on zero entropy data, the bit count for an entire data line of zero entropy samples may be only 7 bits for a scanline of up to 64 blocks and 3 bits for a scanline containing one block. Including a reference, if other prediction modes are used, will add N more bits. Even using a prediction option that inserts a reference, an entire packet may be less than 1 full output word.

The largest packet size is indefinitely long if continuous packet mode is used (as described in the Section "Setting References per Packet"). The maximum packet size for a few selected J values is given in Table 3 and was calculated using Equation 2.

J	Max BlkRef	Max samples Between Ref	Max RefPac	Max samples in a Packet
8	4096	32,768	128	4,194,304
16	4096	65,536	128	8,388,608
32	4096	131,072	128	16,777,216
64	4096	262,144	128	33,554,432

Table 3: Maximum number of samples between references depends on J and BlkRef. Maximum packet size also depends on RefPac.

2.8.1 Setting References per Packet (RefPac)

Once block size J and BlkRef are defined, the final remaining signal to specify is RefPac. Typically, BlkRef will be set equal to the number of blocks in a dataline. The packet size, in terms of samples, is the product of J,BlkPac, and RefPac. The value of RefPac is equal to the binary representation for the number of references-per-packet. (A RefPac value of 5 would be coded ``00000101"). The smallest value for RefPac is 1, while the largest is 128. Packet size can be increased past 128 references by using continuous packet mode as described in the next section.

2.8.2 Continuous Packets (ConPac)

To provide additional flexibility for forming packets USES-32C can be put into a mode in which a control signal defines packet size. The continuous packet mode configuration bit puts the chip into continuous packet mode. In continuous mode, data will be continuously concatenated until the packet is ended. The control pin, EndPac, must be pulsed high coincident with the first sample of the last block of the packet to end the packet as shown in Figure 6. Continuous packets can be formed in any mode, including bypass mode. While in continuous mode, the RefPac value is ignored. The system will need to keep track of the end of each packet in continuous mode in order to signal the decoder when packets end.

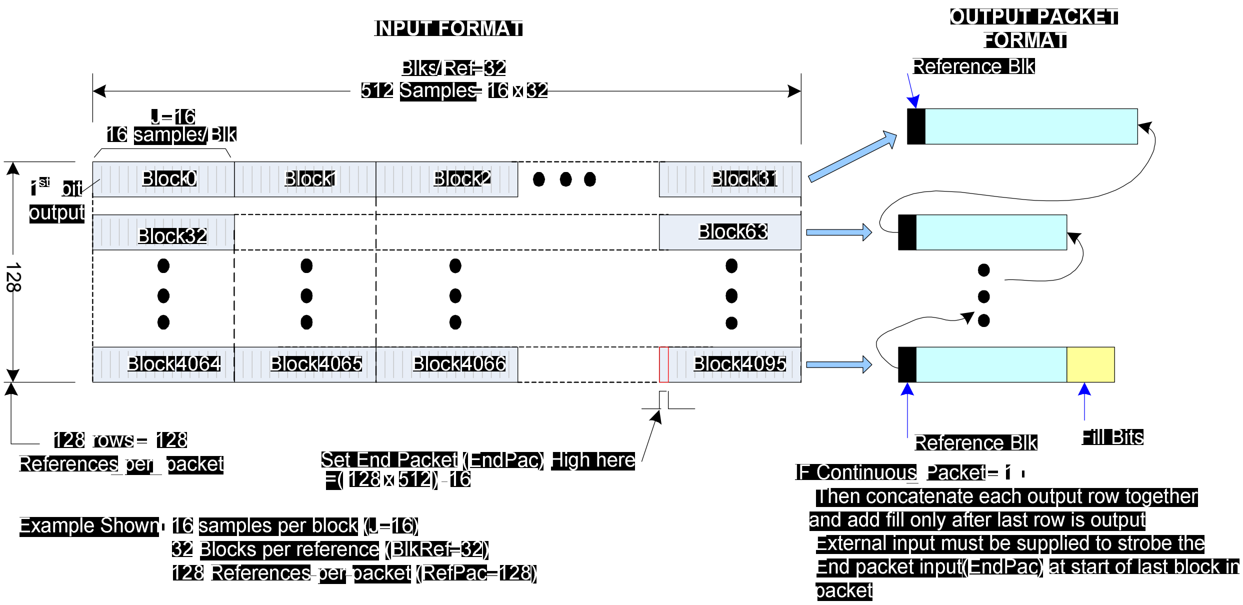


Figure 6: Example of using EndPac in ConPac mode.

2.8.3 Ending a Packet Early (EndPac)

If it is required to end a packet early in any mode, the EndPac signal must be pulsed high coincident with the first sample of the last block of data input to the encoder. When EndPac is pulsed high, the packet will end at the completion of that block. The start of the next data block will begin a new packet. For example, if a packet were ended in the middle of an image scan line, the next input sample of the next block will start another packet. In modes in which references are inserted, that sample will become a reference sample.

2.9 Bypass Mode (Bypass)

Bypass mode makes the chip operate as a packetizer, packing raw input data into 32 bit words, without inserting or changing any bits. To set the chip into bypass mode, when the bypass configuration bit is high, the positive configuration bit must be set to 1. Data packets are formed in the Bypass mode using *J, BlkRef* and *RefPac* values described in Section 2.8. Fill-Samples parameter in configuration has to be set to “0”. Any fill-sample in the input will be packed in the output.

The output signals OutputValid and EOP signals operate in exactly the same manner in both compression mode and the Bypass mode.

2.10 Prediction Modes (PreMod)

USES-32C supports 3 different types of prediction modes and can also function as an entropy coder in which data is not pre-processed before being coded. Input samples are input on the ``X" bus, while the corresponding predicted value (if any) must be on the ``XP" bus.

1. Nearest neighbor prediction: Previous sample is predictor for next sample. XP bus is not used. References are inserted.
2. Simple 2D prediction: Average the value of the previous sample with data at the XP bus. References are inserted. Typical operation of the 2D prediction mode is to code the first scanline using nearest neighbor prediction mode. This is done by bringing the Force-NN-Mode signal high for the first scanline of X bus input data. If Force-NN-

Mode not used in 2D prediction, user has to supply data to the XP bus for all scanlines.

3. External Prediction: User supplies a predictor at XP bus. No References are inserted.

In all prediction modes, all inputs need to be driven to a valid logic value. In 2D mode, even when "Force-NN-Mode" is used for the first scanline, the XP bus has to be supplied with dummy values which will not be used for the first scanline. The nearest neighbor prediction mode is usually the easiest to use because there is no need for the user to store external scanlines to create a predictor. The previous sample acts as the predictor for the present sample. References are automatically inserted in the data stream.

The simple 2D predictor can frequently improve compression by up to 5% over the nearest neighbor mode, but requires the user to provide more external memory and control. Instead of just using the previous sample as predictor, the simple 2D predictor is the average of the previous sample and what is supplied on the XP bus. Often, this is the corresponding sample from the previous line or frame.

External prediction allows the user to provide whatever prediction method is desired, but a word of **caution** is needed. If, in external prediction mode the predictor is simply the adjacent sample from the previous scan line, or the corresponding sample from the previous frame, the decoder can automatically decode the compressed data provided the predictors are supplied. More sophisticated predictors can certainly be used, but it may not be easy to decode without additional hardware. The reason for this is the decoder will be designed so that it will require the previous sample to be available every clock cycle. Before the decoder is able to decode the next sample, it must know what the previous sample was. Every clock cycle, a user would have to receive the current output sample, perform calculations, and feed the new predictor onto the XP bus of the decoder. This must be done in time for the decoder to use the predictor in generating the next sample. This problem can be solved, however, by setting the future decoder into an entropy decoder mode in which it outputs raw sigma values instead of samples. This allows the user flexibility to process the output data in any way desired as long as the decoder's post-processing function can be exercised on the data outside of the decoder. No references are inserted in this External prediction mode.

The prediction modes can be selected using the PreMod configuration variable as defined in Table 4.

Prediction Mode	**PreMod[1:0]**
Nearest-neighbor	00
Simple 2D	01
External Prediction	10
Reserved	11

Table 4: Selecting prediction modes.

2.11 Entropy Coding (EntCode)

Both positive and two's complement data can be coded by USES-32C. However, for entropy coding only positive data can be used. Entropy coding forces the data to bypass the pre-processing section of the chip and go directly into the coding section. In order to have compression in this mode, the more likely occurring data values must be smallest in absolute magnitude. While entropy coding, no references will be inserted in the data stream.

3.0 TYPICAL OPERATION

Data can be accepted after initializing USES-32C as described in Section 2.2. ValidIn signal must go high when the first data word is sent to the X inputs. It is possible to end a packet early, if desired, as described in Section 2.8.3. This section will go through an example of processing a block of image data. The configuration parameter settings are:

Bits per sample 14 bits per sample
PreMod nearest neighbor prediction
BlkRef 4 (insert reference every 4 blocks)

RefPac   2 (two references per packet)

The first sample accepted after initialization will be used as a reference sample. For the first 4 blocks, all differences will be based on this reference sample. The 14 bits making up the reference word will be saved while the differences are processed.

The second sample received will have the previous sample (in this case the reference) subtracted from it. The difference is 15 bits (N+1) and will go to a mapper that will reduce the number of bits back down to 14 (N).

The mapped difference, or sigma value, is sent to the evaluation section that calculates exactly how many bits it will take to encode the sigma value. The coding section can be thought of as having 31 coding machines processing data in parallel. Each coder adds up the bit counts while coding. If a reference is present, there are only 15 sigma values for that block. When a reference is not present (the typical case), there are 16 sigma values per block. After coding is completed on the J sigma values, the option with the fewest number of bits is chosen as the winner. With settings of N = 14 and J = 16, the maximum number of bits for a block occurs when the default condition wins and is equal to 228 bits (N*J+4).

ValidIn signals valid input. ValidIn may go low at any time to put USES-32C into a wait state.

While the first block of data is being encoded, the second block of data is being evaluated. After the compressed data has been formatted, OutputValid signal will go high when the first data word is at the output register. The external system will have only one Clk period to receive each output word. In this example, the first word out contains a 4 bit ID followed by the 14 bit reference sample plus the 7 FS coded pixels the ID for the second block. The OutputValid signal will go low during a block of data if there is no data to output. Any time valid data is at the output, OutputValid will be high. The data within each block will not necessarily be output in a contiguous stream-- there may be pauses between valid output data.

Between blocks of output data, any remainder bits that didn't make up an output word are saved, and then concatenated with the start of the next block's data.

In this example, there are 2 references per packet. After 4 blocks of data are input, the start of the next block will contain the second reference. Processing and outputting those 4 blocks will complete the data packet. Another packet of data will already be in the pipeline of the chip in the process of being coded. When the last data word of a packet is at the output, OutputValid will be high and the end-of-packet (EOP) signal will be high for that clock cycle. The EndBits count will be valid only during EOP, and will specify how many bits in the last word were data bits.

4.0 DATA FORMATS

4.1 Input Samples

Input data may be quantized from N={1, ..., 32} bits per sample. Input data is accepted on every rising edge of Clk when ValidIn is high. Data should be positioned so that the least significant bit of X is input to the X[0] input. Similarly XP[0] is the least significant input bit on the XP bus and the least significant bit of the predictor should be input to XP[0]. All inputs, including unused inputs, must be driven. The unused most significant bits of XP or X may be tied high or low. For positive data, internal circuitry will zero out bits greater than N. For two's complement data, the sign bit is the Nth bit. Two's complement data will be internally normalized to positive values.

4.2 Output Format

When USES-32C is in Bypass mode no IDs or References will be inserted in the data stream. The raw input samples will be packed and output. The first data word will be packed so that its most significant bit is at the most significant bit position of the output bus: OUT[31]. Subsequent data words will be packed so that the most significant bit will be adjacent to the least significant bit of the previously packed word.

An example is given below for packing eight, 4-bit words that were packed in Bypass mode. The numeric values to be coded are: 15, 2, 9, 12, 8, 3, 1, and 14. The eight words fill up one output word and are arranged from left to right, with commas to delineate words in the following:

MSB LSB
1111,0010,1001,1100,1000,0011,0001,1110

While in compression mode, data is output in a parallel manner, 32 bits at a time. The start of every packet will contain an ID placed at the most-significant-bit position of the first coded word. (No header word is used for USES-32C.) If the operating mode is not external prediction or entropy coding, an N bit raw sample will be inserted next as a reference. Data is always output in the following order:

1. ID bits.
2. Reference word (if needed).
3. Fundamental sequence data or default data.
4. Concatenated k least-significant bits (if present).

At the start of a packet, the most significant bits contain the ID bits as per Table 5.

	ID	**ID**	**ID**	**ID**	**ID**
Option	**N≤2**	**2<n≤4**	**4<n≤8**	**8<n≤16**	**16<n≤32**
Zero Block	00	000	0000	00000	000000
2nd Extension	01	001	0001	00001	000001
FS	---	01	001	0001	00001
k=1	---	10	010	0010	00010
k=2	---	---	011	0011	00011
k=3	---	---	100	0100	00100
k=4	---	---	101	0101	00101
k=5			---	0110	00110
k=6				0111	00111
k=7				1000	01000
K=8				1001	01001
K=9				1010	01010
k=10				1011	01011
k=11				1100	01100
k=12				1101	01101
k=13				1110	01110
k=14				---	01111
k=15				---	10000
k=16					10001
k=17					10010
k=18				---	10011
k=19					10100
k=20					10101
k=21					10110
k=22					10111
k=23					11000
k=24					11001
k=25					11010
k=26					11011
k=27					11100
k=28					11101
k=29				---	11110
No compression	1	11	111	1111	11111

Table 5: ID Codes for Entropy Coding Options

The coded data block may or may not contain a reference value. When a reference is present, it follows the ID and only J-1 coded sigma values follow the reference. Figure 7a shows the case where a sample-splitting option is the winner (k=1,..k=12) and a reference is inserted in the data block. Only J-1 coded sigma values follow the reference, with the J-1 k-split-groups following the coded data. Figure 7b shows the same case only without a reference. Note there are J coded sigmas followed by J k-split-groups.

Coded Data
k splits

MSB		LSB
ID	N-bit Reference	FS data J-1 samples
k split data for J-1 samples		

a) Output block format with sample-splitting and reference word.

Coded Data
k splits

MSB	LSB
ID	FS data J samples
k split data for J samples	

b) Output block format with sample-splitting but without a reference.

Figure 7: Output block format at the beginning of a packet when a sample-splitting option wins.

The k split bits from a given sample are not bit-sliced. The k split bits from the first sample will be the first group of k bits following the FS data. The second k-split-group will follow and so on until all k-groups are concatenated. If the block of bits doesn't end on a 32-bit boundary the remainder bits are held and concatenated with the next block's output data. At the end of a packet fill bits will be added to finish the block on a word boundary. The fill bits are zeros and occur in the least-significant bit positions.

When the winning option is the default case, FS, or low entropy, no sample-splitting is done and the k bits are not present in the data block. The ordering of the data, however, remains the same whether or not data contains sample-split bits. The case where a reference is present is shown in Figure 8a; the non-reference case is shown in Figure 8b. Any fill bits needed at the end of a packet to round the block out to a 16 bit word will be in the least-significant bits.

Coded Data

MSB		LSB
ID	N-bit Reference	Coded 15 samples

a) Output block without sample-splitting but with reference.

Coded Data

MSB	LSB
ID	Coded data from 16 samples

b) Output block format without sample-splitting or a reference.

Figure 8: Output block format at the beginning of a packet for default, FS,and low entropy options.

4.3 Coding Examples

Table 6 shows 3 blocks of data that represent one packet of samples. Each block is 8 samples and nearest neighbor prediction is used. The intent of this example is to show how the most significant bit position is the 32nd bit and illustrate the position of fill bits at the end of a packet. The fill bits at the end of the packet are shown in Figure 9, and are in the least-significant-bit positions. Data enclosed in curly brackets represent coded data for that block. Data enclosed in brackets is the ID data that starts each block. Only the first block contains a reference. The winning option for block 1 is the FS coding option. The winner for the second block is the k=3 coding option. The third block is coded with default data.

Table 6 includes the sample data, the delta values (difference between the present sample and the previous sample), and the sigma values (mapped differences).

The following example was simulated with the following settings:

N	14 bits per sample
PreMod	Nearest neighbor prediction
BlkRef	3 Insert a reference once every 3 blocks
J	8 Samples per block

SAMPLE	**delta**	**sigma**	**Binary sigma** **MSB LSB**
967	-	-	00001111000111 ← reference
967	0	0	00000000000000 Block 1
967	0	0	00000000000000
965	-2	3	00000000000011
965	0	0	00000000000000
965	0	0	00000000000000
965	0	0	00000000000000
965	0	0	00000000000000
960	-5	9	00000000001001 Block 2
970	10	20	00000000010100
968	-2	3	00000000000011
976	8	16	00000000010000
971	-5	9	00000000001001
972	1	2	00000000000010
972	0	0	00000000000000
980	8	16	00000000010000
2957	1977	2957	00101110001101 Block 3
11156	8199	11156	10101110010100
5883	-5273	10500	10100100000100
11271	5388	10776	10101000011000
12000	729	1458	00010110110010
4568	-7432	11815	10111000100111
6639	2071	4142	01000000101110
12257	5618	11236	10101111100100

Table 6: Sample Data, Deltas, and Sigma Data for a packet of data. Each block has 8 samples and there are 3 blocks per packet.

MSB LSB	
{[0001],000011110001	← FS ID=”0001” followed by reference word
11,1,1,001,1,1,1,1}{[0100],	← 7 FS, start of blk k=3 ID=”0100”
01,001,1,001,01,1,001,	← 8 FS words total for k=3
001,100,011,000,001,0	← Start of k=3 splits, grouped in threes
10,000,00}{111],0010	← Start of default data ID=”1111”
1110001101,101011	← 14 bits of sigma values
10010100,10100100	
000100,1010100001	
1000,000101101100	
10,10111000100111,	
01000000101110,10	
101111100100}xxxx	← End of packet, 4 fill bits x=0

Figure 9: Output block format at the beginning of a packet for default, FS,and low entropy options.

5.0 TIMING DIAGRAM

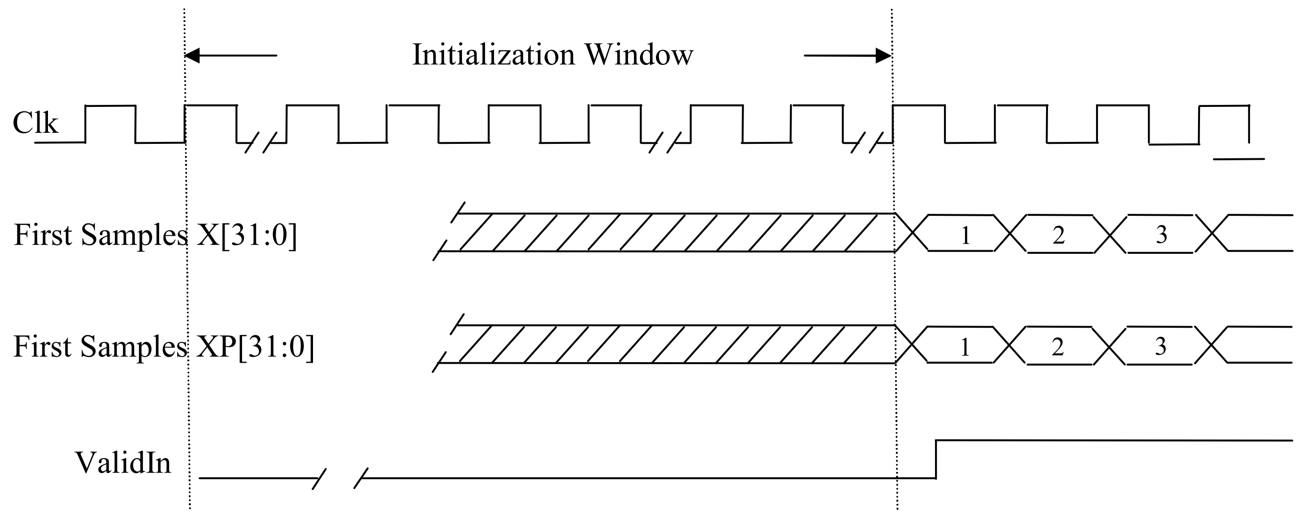


Figure 10: Initialization and arrival of first samples. ValidIn is held low during initialization to set USES-32C in compression mode.

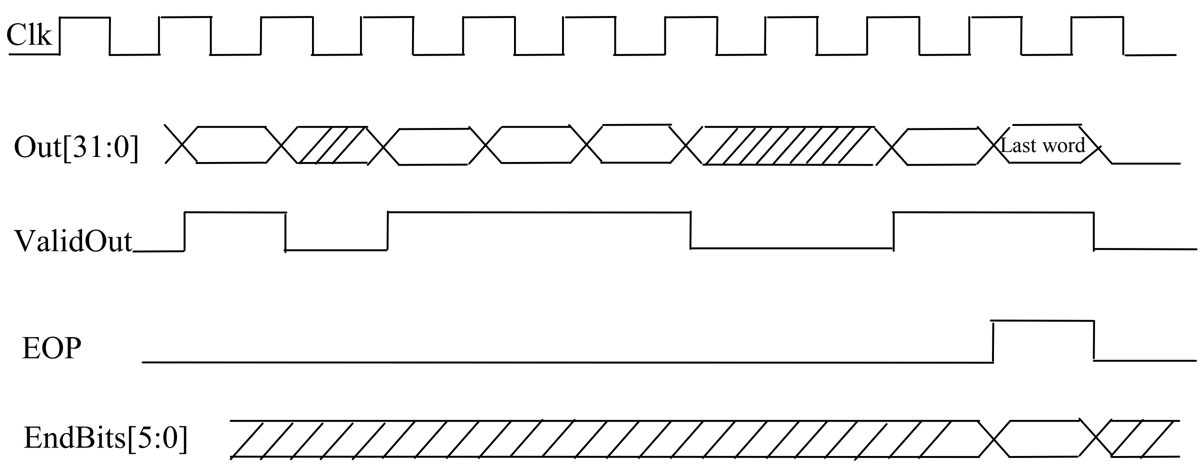


Figure 11: Outputting one packet of data.

6.0 REFERENCES

[1] *Lossless Data Compression*. Recommendation for Space Data System Standards, CCSDS 121.0-B-1. Blue Book, Issue 1, Washington, D.C.: CCSDS. Shorter ID codes for N ≤4 specified in Issue 2 to be released 2011

[2] ``A VLSI Chip Set for High Speed Lossless Data Compression", *IEEE Transactions on Circuits and Systems for Video Technology*, December 1992, by J. Venbrux, P.S. Yeh, and M. N. Liu.