Universal Source Encoder for Space - USES

ICs LLC
2600-A E. Seltice Way #234
Post Falls, Idaho 83854
Phone:(208)755-8990 FAX:(208)773-5747

Version 3.1, November 3, 2005 - address change only

Features:

- Lossless Compression up to 20 Msamples/s.
- Uses 14 Rice Algorithm coding options plus one new low entropy option. The low entropy option allows compression of thresholded or other low entropy data.
- Variable identifier bit count to improve performance for N <= 8 bits. When N <= 8, only a 3 bit ID is used. Larger quantizations use 4 bits for code identifier.
- Input Specifications:
  - Quantization supported: $N = \{4 \ldots 15\}$. In multi-spectral mode the maximum quantization allowed is 14 bits.
  - Data may be positive or two's complement.
  - Data is coded in block sizes (J) of 8, 10, or 16 samples. If needed, provides a means for a partial block of samples to be filled to a full block by repeating the last sample.
  - Supports continuous input data stream as well as pauses in data flow between blocks.
- Output Specifications:
  - Parallel 16 bit output with tri-state enable.
  - Output data is concatenated into data packets with an output signal specifying end-of-packet. Packets may be ended early and a continuous packet mode is supported.
- Operating Modes:
  - Nearest Neighbor prediction.
  - Simple 2D prediction.
  - External prediction.
  - Multi-Spectral mode: typically improves compression 5%-30% for spatially registered multi-spectral data when the data maintains spatial correlation between spectral bands.
  - Entropy-coding: use entropy coder but not pre-processor.
  - Bypass mode: no compression, just packetizing raw input data to a 16 bit output.
  - Reference to reference differencing: allows compression of references. Only supported for nearest neighbor and 2D prediction within a packet.
- Rad-Hard Gate-Array in an 196 pin flat-pack utilizing 88 I/O pins.

Contents

1 General Description

USES (Universal Source Encoder for Space) accepts sampled data that it will code using lossless algorithms. Input data may be two's complement or positive and quantized within the range $N = \{4 \ldots 15\}$ bits. Maximum quantization is 15 bits/sample except in multi-spectral mode in which the maximum quantization is 14 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 8, 10, or 16 samples per block. If a scanline, for example, does not end in an integral number of J samples, the final input block must be filled out to J samples by using the input signal ValBlk.

USES has two major modes of operation: compression or bypass. In compression mode the entropy coder is used to code input data but 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 as a packetizer forming data into 16 bit words.

When in compression mode, USES supports 4 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. Externally supplied. Used for prediction in Y and Z dimensions.
4. Multi-spectral. A special preprocessor is used for multi-spectral data.

USES 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 Fig. 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.

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 15 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 will always choose the winning option that provides maximum compression. Compressed data is output on a 16 bit bus.

The entropy coder performs a 14 option Rice Algorithm coding set with the addition of one new low entropy coding option. USES efficiently codes data from 0 bits of entropy up to about 14.5 bits per sample. For an explanation of the Rice algorithm refer to Jet Propulsion Laboratory (JPL) publication 91-1, "Algorithms for a very high speed universal noiseless coding module," by R. F. Rice, P.S. Yeh, and W.H. Miller. For a description of previous work and the performance

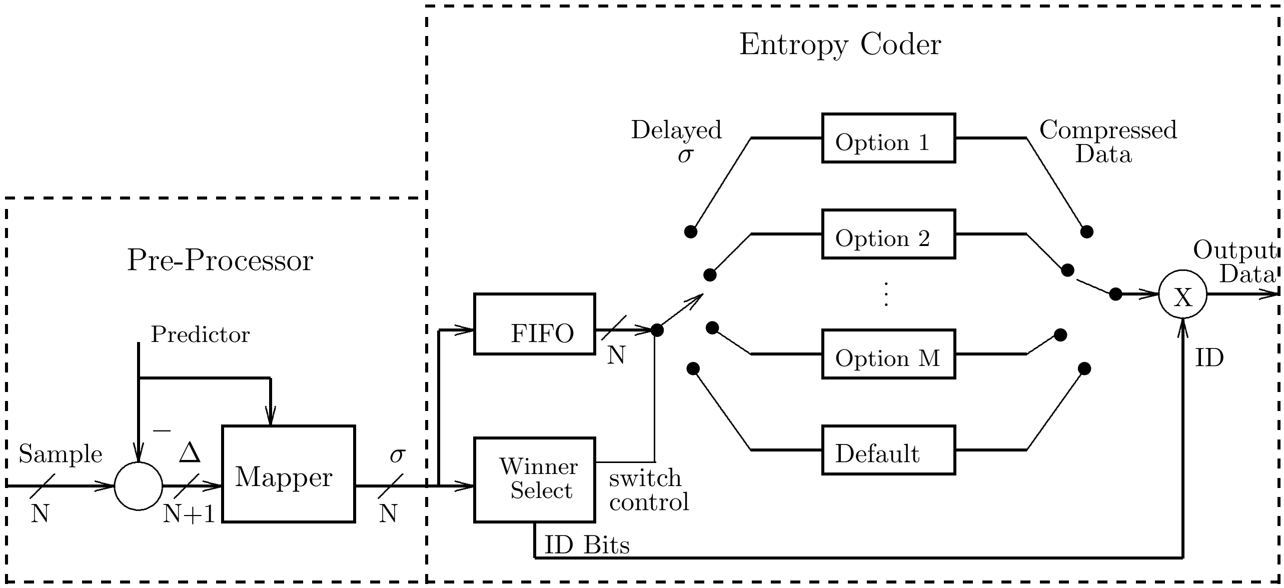


Figure 1: Block diagram of Rice algorithm architecture.

of the earlier version of this chip, the reader is referred to the journal article: "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.

The 15 coders can be divided into 4 major option types. The simplest is the default option, which sends uncompressed sigma data to the output. The default option limits the worst case expansion to be equal to the sum of the number of input bits plus the 3 or 4 ID bits.

The second coder type is called 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 15 option set includes using 12 sample-split options $k = \{1 \dots 12\}$ combined with the FS option. In Fig. 1 what is labeled "Option M" corresponds to the maximum k split option used in the USES design: k=12 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, just following runs of 1..4 blocks of zeros. 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.

The chip includes an operating mode designed specifically for multi-spectral data. This multi-spectral multi-source mode uses a specifically designed pre-processor to reduce the correlation before entropy coding the signal. The technique used in this implementation is aimed at signals that possess correlation between sources, yet may have possible varying bias levels or different sensor Analog-to-digital-converter (ADC) gains. The technique typically improves compression from 5% to 30%. In multi-spectral mode the largest quantization supported is 14 bits.

2 Circuit Operation

USES may operate as a data packetizer or as a compressor on sampled data quantized from $N = \{4 \dots 15\}$ 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=8, 10, or 16 samples. If a line of data ends on a sample count that is not an integral multiple of J, the ValBlk signal, discussed in sub-section 2.7 can be used to fill out the remainder of the line by repeating the last pixel to make an even multiple of J samples.

Blocks of data are grouped into larger blocks of data by the BlkRef signal. 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 are defined by the product of J and BlkRef.

Datalines are grouped into packets by specifying the RefPac signal. 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 16 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 diagram of USES is shown in Fig. 2. The inputs and input control are on the left, the outputs and output control are on the right. Signals related to packet size, packet generation, and ending a packet, are at the top of the figure. Other control signals are shown on the bottom.

The Clk signal has a maximum frequency of 20 MHz. All 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 of 20 Msamples/s. All output signals and data are clocked on the rising edge of Clk, resulting in a maximum output transfer rate of 20 Mwords/s.

All input pads are CMOS and must be actively driven to a valid low or high. **No inputs may be left floating**. All output pads, likewise, will be CMOS output pads.

Pertinent control timing diagrams for the USES are in Section 10 of this document.

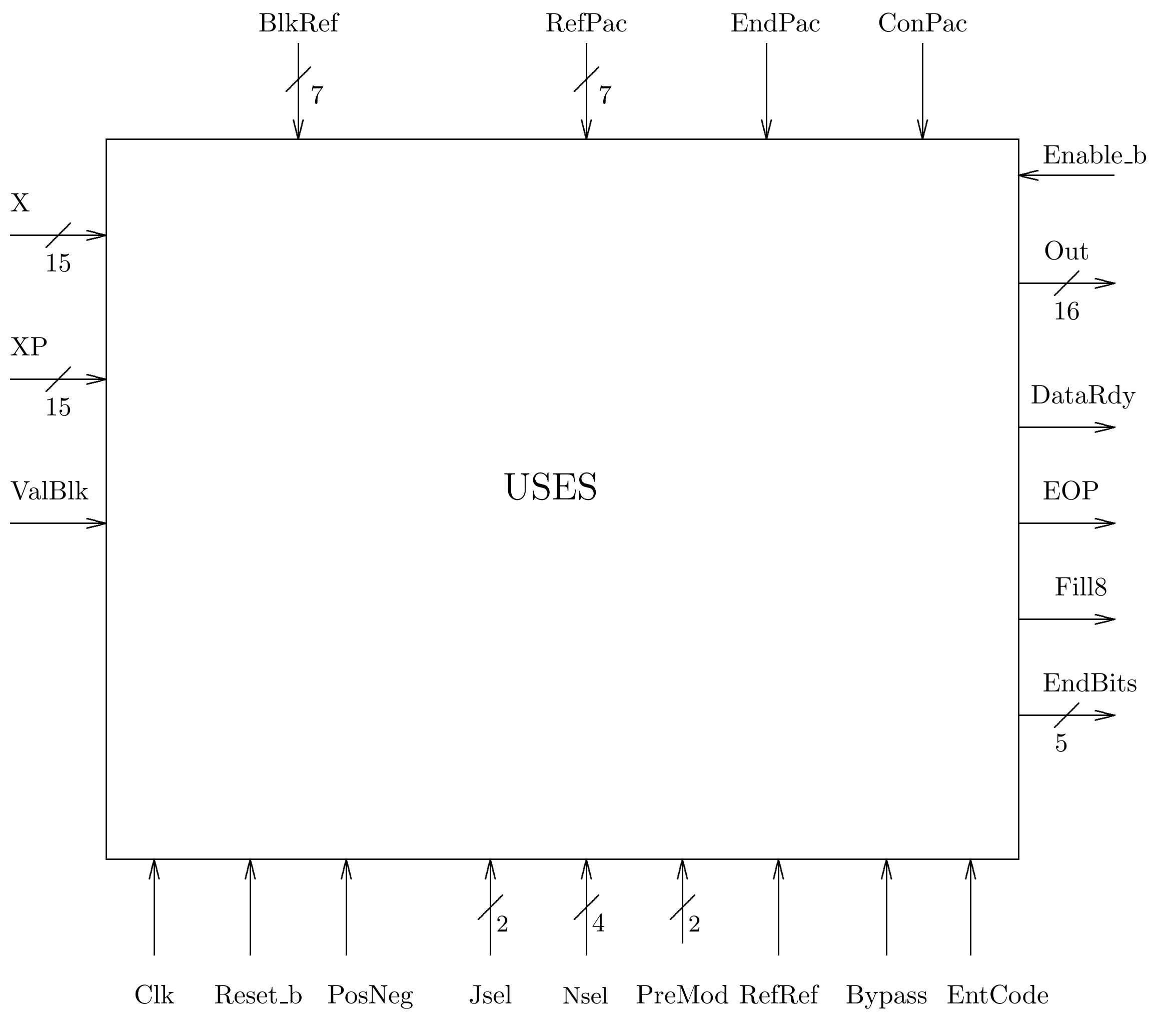


Figure 2: Input/Output Diagram of USES.

2.1 DataFlow

Whenever valid data is to be received by USES the ValBlk (Valid Block) signal must be high. After initialization, ValBlk must go high accompanying the first sample and remain high for all valid samples. Once ValBlk goes high, USES expects a continuous stream of J samples. The clock period immediately following a block of J samples determines whether USES will accept new data or go into a wait state. ValBlk remaining high just after the previous block signals USES to continue to accept a continuous stream of J data samples. (A block that does not contain J samples may be filled out to J samples as discussed in Section 2.7)

If ValBlk goes low after a block of samples is received, USES will wait until it goes high again, signaling the start of another block of samples. There are no wait states allowed within a block of samples. (ValBlk is not allowed to be brought low and then high again before the end of a single block of data.) While USES is waiting for ValBlk 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. If the data was stopped within a packet and it happens that the internal data has ended on a 16 bit boundary, all data will be output. In the majority of cases data doesn't end on a 16 bit boundary, and stopping data within a packet will cause USES to output all but the last word of the block. USES will keep the last partially-filled word and wait for more data in order to concatenate blocks to form a packet. 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 ValBlk signal may be brought high is just following the rising edge of the second Clk after Reset_b going high. 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 16 bit data bus. Any bits that do not make up a full 16 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.

In all the examples that follow it is assumed that Enable_b is tied low to enable the Out bus to pass data. When Enable_b is brought high only the 16 bit Out bus will tri-state, all other output signals continue to be enabled.

There will be no handshaking at the output. At the start of a packet the DataRdy signal will accompany the first data word by going high. DataRdy is true only when data is at the output. In the course of outputting a single block of data, the data ready 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 DataRdy signal is active. All data transfers take 1 clock period. Between blocks, USES 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 DataRdy signal will be high. One other output, Fill8, will go high coincident with EOP signal if the last word of the packet contains 8 or more bits of fill. The Fill8 signal is included to allow any packetizing circuitry that may be downstream of USES to decrease the overhead associated with ending a variable bit stream on a 16 bit boundary. 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..16 bits. Fig. 13 in Section 10 shows waveforms for an end-of-packet condition.

The worst case data transfer occurs when $N = 15$, $J = 8$ samples/block, and the default condition wins for at least four consecutive blocks. Under such conditions, there may be valid output data for up to 32 consecutive clock cycles.

Latency is the number of clock cycles required from the time the data is at the input of USES to the time data is ready at the output and DataRdy is high. The minimum latency of USES is fixed in all operating modes but depends on block size. Table 1 shows the minimum latency of USES under the different block sizes. The table indicates when an external circuit must be ready to receive data, if it is available.

Mode	Clock Cycles
J=8	16
J=10	18
J=16	22

Table 1: Minimum latency depends on J.

When an packet has ended (either by EndPac or internally), the maximum latency from the time the last word of the packet is at the input till the final packet data is at the output is given in Table 2.

Mode	Clock Cycles
J=8	32
J=10	36
J=16	48

Table 2: Maximum latency when a packet ends.

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 datalines have been coded, or until 128 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 must be initialized. This is accomplished by holding Reset_b low for a minimum of 4 clock cycles and keeping control lines stable for the following 2 clock cycles. This sets USES into a known state. At all times the Aeroflex requested testing pin, TestIn, must be low in order for the chip to operate properly.

Fig. 3 shows the initialization window. Reset_b must be low for a minimum of 4 clock cycles but control lines must be stable a minimum of 2 clock cycles before reset transitions high and two clock cycles after reset goes high. After initialization Nsel, Jsel, PosNeg, BlkRef, EntCode, RefPac, ConPac, and RefRef are ignored. During initialization ValBlk and EndPac must be low.

The prediction mode pins (PreMod) are allowed to change dynamically between nearest-neighbor and 2D. Other changes between prediction modes will be ignored after initialization. If PreMod[1] pin is high (indicating the chip will be in multi-spectral or external prediction modes) then after initialization, BOTH PreMod[1] and PreMod[0] pins will be ignored. Only when PreMod[1] is low at initialization, will PreMod[0] be sampled during the the last block of each dataline. (Recall, a dataline is the number of blocks defined by BlkRef. In images, a dataline is often equal to the number of blocks in a scanline). To change modes, the state of PreMod may change between "00" (nearest neighbor) and "01" (2D), or vice-versa any time while accepting a line of data. (This change may even happen during the last sample of the last block of a dataline). The change, however, will only be sampled during the last block of a dataline. The change will take affect after the last sample of the present data line has been clocked into the X input bus.

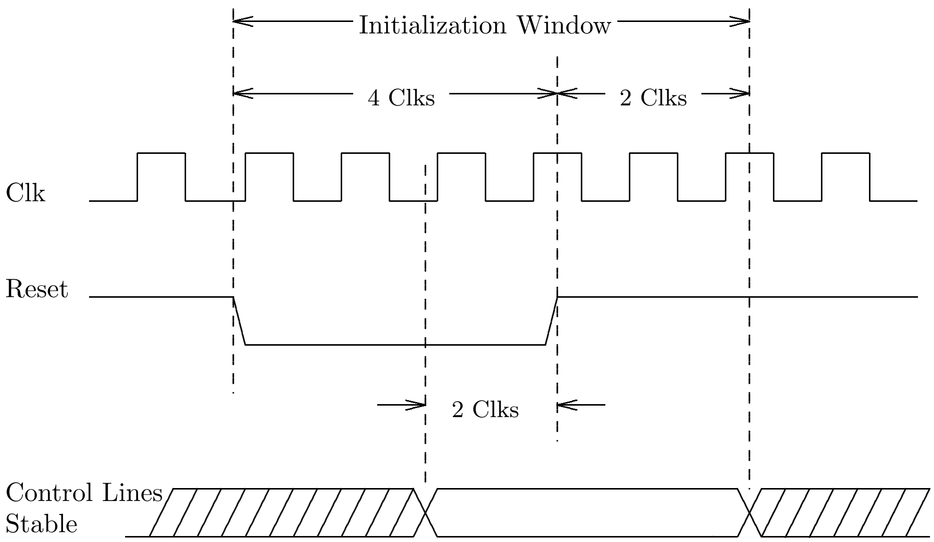


Figure 3: Initialization window.

Once initialized, USES must be reset to re-initialize to operate in a different configuration. After initialization, the output signal DataRdy will remain low until the first data word is at the output. The EOP, and Fill8 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.

- Nsel. Quantization: Valid only during initialization.
- PosNeg. Positive/Negative Data: Allowing inputs to be positive or bipolar data (in the form of two's complement format). Valid only during initialization.
- Jsel. Blocksize: Valid only during initialization.
- BlkRef. Blocks-per-reference: Valid only during initialization.
- RefPac. References-per-packet. Valid only during initialization.

- ConPac. Continuous packet mode. Valid only during initialization. If ConPac is selected RefPac will be ignored.
- ValBlk. Valid input data on input. Must be low during initialization.
- EndPac. End Packet. Must be low during initialization.
- PreMod. Prediction mode. If PreMod[1] pin is high (indicating External Prediction or Multi-spectral mode is selected), both PreMod pins will be ignored after initialization. If PreMod[1] is low (indicating 2D or nearest neighbor has been selected) than PreMod[1] pin will be ignored after initialization but PreMod[0] will continue to be sampled to allow the user to change between 2D and nearest neighbor prediction modes.
- RefRef. Reference to Reference Differencing. If high during initialization, USES will difference adjacent references in a packet in nearest neighbor and 2D prediction modes only. In those modes, each packet will contain only one reference at the start of the packet. Other references will be differenced and coded. Valid only during initialization.
- Bypass. Bypass mode in which chip will act as a packetizer. Active high. Valid only during initialization. When bypass is selected, all other mode pins will be ignored (including EntCode).
- EntCode. Entropy coding of positive data. Valid only during initialization. Active high. Bypass must be low when EntCode is selected. PreMod pins are ignored while in EntCode or Bypass.

In all modes, the entire initialization time requires 6 clock periods, 4 for reset active, and two for control signals remaining stable.

2.3 Setting Quantization (Nsel)

The quantization range supported is N=4..15. (In multi-spectral mode the maximum supported quantization is 14.) The binary value of Nsel, sets the value of N. For example, to set a quantization of ten, the pins must have the bit pattern "1010" 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 (PosNeg)

USES will compress positive and two's complement data. When quantization is specified (as defined in the Initialization Section) and the PosNeg signal is tied high during initialization, USES will interpret the Nth bit as the sign bit. Internally USES sign extends the data. For example, if the quantization (including sign bit) is 10 bits, initialize USES 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 $0\dots 1023$; 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. Entopy Coding of two's complement data is discussed in Section 2.11.

2.5 Setting Block Size J (Jsel)

USES compresses blocks of data rather than a continuous data stream. USES supports three different block sizes: J= 8, 10, 16 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. A block size of 10 is included for images that have scanline lengths that are multiples of 10 pixels. The Jsel inputs specify block size.

Jsel[1,0]	Block
00	J=16
01	J=10
10	J=8
11	J=16

Table 3: Setting block size. The leftmost column is the most-significant bit of Jsel.

The number of samples in a dataline must be an integral multiple of the block size, which is set at J samples. If the number of samples in a dataline is not a multiple of J, the line must be filled out using the ValBlk signal as described in Section 2.7. The maximum number of coded bits per block occurs when the quantization level N is 15, J=16, and the default option wins, resulting in 244 bits. The maximum block size for a given quantization and block size is given by the following equation.

$$\begin{aligned} if \quad (N > 8) \quad & Max\_Block\_Size = Jsamples/block * Nbits/sample + 4IDbits \\ else \quad & Max\_Block\_Size = Jsamples/block * Nbits/sample + 3IDbits \end{aligned} \tag{1}$$

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 (BlkRef)

The BlkRef signal 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 2 that determines packet size. 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, 2D, or multi-spectral modes are 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. BlkRef is set to the number of samples in a data line. For images, BlkRef is typically set to the number of blocks in the image scanline.

There are three cases, however, in which BlkRef is ignored. BlkRef is ignored in the non-reference modes, External prediction, Bypass, and Entropy coding, when packets are being generated using continuous mode.

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 scanlines, but they may be inserted more often by setting the 7 control lines labeled “BlkRef”. 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 128 blocks. When J=16 the maximum samples between references is 2,048 samples. When J=8, the maximum is 1024, and when J=10, the maximum is 1280 samples between references. The binary value of BlkRef is equal to the number of blocks between references except in the zero case. The maximum setting (128) is achieved by setting all the BlkRef control lines to zero.

If an input scanline 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 scanline, but is of no appreciable effect on compression. For example, if an image scanline contained 3K pixels (3072), BlkRef might be set so that a reference was inserted in the middle of the scanline (1536 pixels). If J=16, BlkRef would be set to 96 to allow two references per scanline ($J * 96 = 1536$). (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 scanline unless the scanline exceeds 128 blocks of samples.)

2.7 When there is Insufficient Input Data for a Dataline

The ValBlk signal has an added function besides providing wait states. If a block contains less than J samples, such as at the end of a scanline, the ValBlk signal may be used to fill out the final block to make it J samples. ValBlk should be kept high when actual data is being input to USES, but brought low and kept low during the fill time. When ValBlk goes low, the input circuitry will repeat the last sample value and ignore new samples. This fills the block with the lowest possible entropy data. If ValBlk remains low after the block is filled, the encoder will go into a wait state; if ValBlk goes high with the first new data sample, another block will be processed.

This paragraph describes filling out a block of data in modes that use the XP bus: 2D, External Prediction, and Multi-spectral. 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 dataline. 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 4 bit bus specifying how many, if any, samples only at the end of each line are to be decoded as repeated pixels. 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 nearest neighbor or entropy coding modes, filling out a block may be done at any point in a data line. In bypass mode, bringing ValBlk low during the block will repeat the last valid sample. **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 16 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 16 bits. In forming packets of data, USES concatenates any remainder bits from one block with the output bits from the next block.

There are some restrictions on how small a packet can be. When operating USES in prediction modes other than multi-spectral, packets must be at least 8 blocks in size only when N=15 and J=8 or J=10. This is due to the overhead caused by rounding a packet out to the nearest 16 bit word. Any fill bits used to end the packet on a word boundary represents overhead. If data can be stopped, however, for at least 2 clock cycles after an end-of-packet condition, packets can be as small as a single block even under the conditions just specified. When using prediction modes other than multispectral mode, there are no restrictions on the minimum blocks per packet when N is less than 15 bits/sample or when the block size is set at 16 samples/block.

When operating in multi-spectral prediction mode, and N=14 and J=8 or J=10, all packets must contain at least 8 blocks of data. Smaller quantizations have no restrictions on packet size and may be as small as 1 block per sample. There are no restrictions on minimum packet size when operating in Bypass mode regardless of quantization or block size.

The size of those packets, in terms of samples, is user selectable and is dependent on setting three control lines: Jsel, BlkRef, and RefPac. *These three control lines 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 BlkRef and RefPac simply as two multiplying factors to determine packet size as defined in Equation 2.

The block size Jsel must be selected to be J= 8, 10, or 16 samples per block. The BlkRef control lines must be selected to specify how many blocks should be coded between references. Finally, the RefPac signal must be specified to determine how many references are to be in a packet. Packet size, in samples, is determined by the following equation:

$$Packet\_size = J * BlkRef * RefPac \tag{2}$$

The smallest number of bits in a packet occurs in entropy coding or external predictor modes (which include no references in the data stream) and zero entropy data, the bitcount for an entire dataline of zero entropy samples may be only 9 bits. 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 is given in Table 4 and was calculated using Equation 2.

J	Max BlkRef	Max samples between Ref	Max RefPac	Max samples in a Packet
8	128	1024	128	131,072
10	128	1280	128	163,840
16	128	2048	128	262,144

Table 4: 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 is 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. RefPac=0, sets RefPac equal to the maximum value of 128. Packet size can be increased past 128 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 can be put into a mode in which a control signal defines packet size. The pin "ConPac" must be held high during the initialization process to put 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 anytime during the last block of the packet to end the packet. Continuous packets can be formed in any mode, including bypass mode. While in continuous mode, the RefPac value is ignored. Once initialized, the ConPac input will be 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.

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 for a minimum of one clock cycle once during the time when the last block of data is 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 scanline, the next input pixel of the next block will start another packet. In modes in which references are inserted, that pixel will become a reference pixel.

2.9 Setting Bypass Mode (Bypass)

Bypass mode makes the chip operate as a packetizer, packing raw input data into 16 bit words, without inserting or changing any bits. To set the chip into bypass mode, the Bypass pin must be kept high during initialization. After initialization, the Bypass pin will be ignored. Bypass mode uses quantization but always ignores the PosNeg pin.

In compression mode and Bypass mode the output signals DataRdy, EOP and Fill8 signals operate in exactly the same manner.

2.10 Prediction Modes (PreMod)

USES supports 4 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.
3. External Prediction: User supplies a predictor at XP bus. No References are inserted.
4. Multi-spectral mode: Use one input at the XP bus as reference for the second input at the X bus. References are inserted.

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 pixel from the previous scanline, or the corresponding pixel 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 mode.

The final mode is designed to compress multi-spectral data. Unlike the other modes, the largest supported quantization for multi-spectral data is 14 bits. As mentioned previously in the "General Description" Section of this specification, the multi-spectral multi-source mode uses a specifically designed pre-processor to reduce the correlation before entropy coding the signal. The technique used in this implementation is aimed at signals that possess correlation between sources, yet may have possible varying bias levels or different sensor Analog-to-digital-converter (ADC) gains. The technique typically improves compression from 5% to 30%. To use this mode, simply input the corresponding sample from one of the other spectral bands to be used as a predictor for the present sample. References are automatically inserted in this mode. The spectral bands, must however, be spatially registered.

The prediction modes can be selected using the PreMod pins as defined in Table 5.

Pred. Mode	PreMod[1-0]
nearest-neighbor	00
Simple 2D	01
External Prediction	10
Multi-spectral	11

Table 5: Selecting prediction modes.

2.11 Entropy Coding(EntCode)

Both positive and two's complement data can be entropy coded. 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 occuring data values must be smallest in absolute magnitude. While entropy coding, no references will be inserted in the data stream.

There are two ways to initialize the chip to entropy code positive data but only one way to initialize the chip to code two's complement data. The first way to entropy code positive data is to set the EntCode pin high, and the PosNeg pin low. This puts USES into positive data Entropy coding mode. With EntCode set high during initialization, the XP bus will always be ignored and data to be coded should be placed at the X inputs. The second way to entopy code positive data is to set the chip in External Prediction mode, tie all XP inputs low, and set PosNeg to low. The two methods are identical in function; the EntCode pin was added to automatically zero out the XP bus.

To entropy code two's complement data, the external prediction mode is used. (The EntCode pin only works for positive data, not two's complement data.) To entropy code two's complement data, all of the inputs to the XP bus must be tied low and the chip set in External Prediction mode with Entcode and Bypass tied low (Section 2.10) and PosNeg set high.

2.12 Reference to Reference Differencing (RefRef)

In two prediction modes, nearest neighbor and 2D prediction, the references may be compressed. This mode is set by keeping the RefRef pin high during initialization. In reference differencing mode, each packet (including the packets formed using continuous mode) will contain a single reference at the start of the data packet. At the start of the next dataline, as defined by BlkRef, another reference typically would be inserted. Reference differencing mode will take the difference between the present "reference" value and the previous one. In nearest neighbor mode, that difference will be mapped and coded. (No reference will be inserted). In 2D mode, the previous reference will be averaged with what is on the XP bus to form the predictor. The predictor will be subtracted from the present value of X, and the difference will be mapped and coded. For large images, reference differencing will not provide much compression gain. For small images, compressing reference may produce a measurable gain in compression.

Reference differencing assumes that references have some correlation. For images, this is generally true if references are to be inserted at the start of every scanline. Adjacent references, in that case, have spatial correlation. If more than one reference were to be inserted in a long scanline, for example, references would no longer have spatial correlation and hence, differencing might actually cause slight expansion.

3 Typical Operation

Data can be accepted after initializing USES as described in Section 2.2. ValBlk signal must go high when the first data word is sent to the X inputs. The ValBlk signal and all input data (including any External Predictor Data) must be valid for at least one Clk period in order to form a block. If ValBlk signal goes low during a block, the remainder of the block will be filled with the last valid data word. 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 control line settings are:

Nsel	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 15 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 ($N * J + 4$).

Once the first block of data has been input, ValBlk may go low to put USES in a wait state. It will continue to process the first block of data, but will not accept any new data until the ValBlk signal goes high. Once it goes high, samples must come continuously until the entire block of data is input; no waits are allowed within a block.

While the first block of data is being encoded, the second block of data is being evaluated. After the compressed data has been formatted, DataRdy 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 12 bits of the 14 bit reference sample. The DataRdy signal will go low during a block of data if there is no data to output. Any time data is at the output, DataRdy 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, DataRdy will be high and the end-of-packet (EOP) signal will be high for that clock cycle. If the last word of a packet contains 8 bits or more of fill, the Fill8 signal will go high for that clock cycle. If the last word of a packet has less than 8 bits of fill, Fill8 will remain low. The EndBits count will be valid only during EOP, and will specify how many bits in the last word were data bits.

4 Data Formats

4.1 Input Samples

Input data may be quantized from $N = \{4..15\}$ bits per sample. Input data is accepted on every rising edge of CLK when ValBlk 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. 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 internally be sign extended.

4.2 Output Format

When USES 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[15]. 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.

Fig. 4 shows the bit positions for four, 4-bit words that were packed in Bypass mode. The numeric values to be coded are: 8, 3, 1, and 14. The four words fill up one output word and are arranged from left to right, with commas to delineate words.

```
MSB             LSB
1000,0011,0001,1110
```

Figure 4: Outputting four, 4-bit input words packed in Bypass mode.

While in compression mode, data is output in a parallel manner, 16 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.) 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. When quantization is 8 bits or less, a 3 bit ID will be output. Larger quantizations will use a 4 bit ID.

Option	3-bit-ID	Decimal Equivalent
low	000	0
FS	001	1
k=1	010	2
k=2	011	3
k=3	100	4
k=4	101	5
k=5	110	6
default	111	7

Table 6: ID codes for data that is 8 bits or less.

Table 6 shows the options set and ID bits used in compressing data that is quantized to 8 bits or less. The default option is used to limit possible expansion to 3 ID bits per block.

The 4 bit ID specifies which of the 15 options is the winner for that block of samples. The ID codes are shown in Table 7. USES determines the exact count for each option and selects the option with the fewest bits. In the event of a tie between default and any other option default will win. Note that except for the default, the number of k splits is equal to ID - 1.

Option	4-bit-ID	Decimal Equivalent
low	0000	0
FS	0001	1
k=1	0010	2
k=2	0011	3
k=3	0100	4
k=4	0101	5
k=5	0110	6
k=6	0111	7
k=7	1000	8
k=8	1001	9
k=9	1010	10
k=10	1011	11
k=11	1100	12
k=12	1101	13
default	1111	15

Table 7: ID codes for 15 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. Fig. 5a 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. Fig. 5b shows the same case only without a reference. Note there are J coded sigmas followed by J k-split-groups.

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

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

	MSB	LSB
Coded Data	ID	FS data for J samples
k splits	k split data for J samples	

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

Figure 5: 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 16 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 Fig. 6a; the non-reference case is shown in Fig. 6b. 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.

	MSB		LSB
Coded Data	ID	N bit Reference	Coded 15 samples

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

	MSB	LSB
Coded Data	ID	Coded data from 16 samples

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

Figure 6: Output block format at the beginning of a packet for default, FS, and low entropy options.

4.3 Coding Examples

Table 8 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 16th 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 Fig. 7, 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 8 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.

PIXEL	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 8: Pixel 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
11,1,1,0001,1,1,1,1}{[0100],	← 7 FS, start of blk k=3 ID="0100"
01,001,1,001,01,1,1,001,	← 8 FS words total for k=3
001,100,011,000,001,0	← Start of k=3 splits, grouped in threes
10,000,000}{[1111],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 7: Output for 3 blocks of data each with N=14 and J=8.

5 Further Reading

The algorithms in the low entropy coding section of the chip have not been published yet and will not be discussed in detail in this specification. The algorithm behind the multi-spectral mode in this chip likewise has not been published yet.

The following sources provide background on the Rice algorithm or past work in developing a chip to implement the algorithm.

For an explanation of the Rice algorithm refer to JPL publication 91-1, "Algorithms for a very high speed universal noiseless coding module," by R. F. Rice, P.S. Yeh, and W.H. Miller. The mapper uses in USES in this specification produces the same mapping at that defined on page 16 of JPL publication 91-1.

For a description of previous work and the performance of the earlier version of this chip, the reader is referred to the journal article: "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.

For more information on coding aspects and theory behind the Rice Algorithm refer to JPL Publication 91-2 "On the Optimality of Code Options for a Universal Noiseless Coder," by P.S. Yeh, R. Rice, and W. Miller.

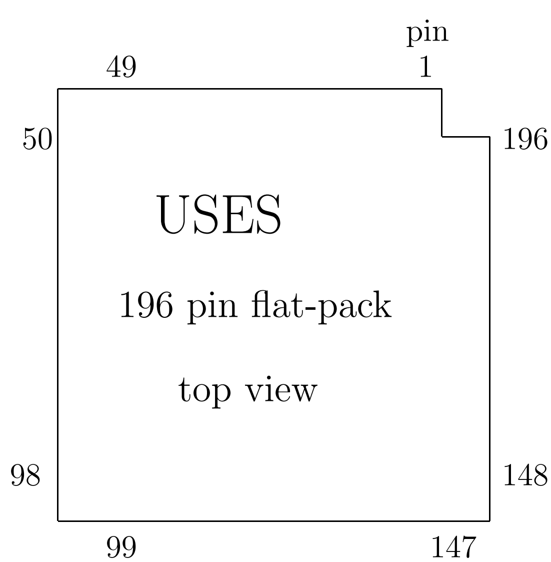
Other publications include JPL publications 79-22 and 83-1 Parts I and II of "Some Practical Universal Noiseless Coding Techniques" by Robert F. Rice and Jun-ji Lee, and JPL publication 91-3 "Some Practical Universal Noiseless Coding Techniques, Part III, Module PSI14,K+" by Robert F. Rice .

6 Test Mode

Aeroflex requires that their gate arrays be able to be put into a test mode in which the input pads are tested for sensitivity to voltage levels. The TestIn pin is reserved for putting the chip into the test mode. In normal operation, this pin must be tied low, The TestOut pin is reserved as the test output pin needed for Aeroflex's test.

7 USES Pins Description

Name	Type	Function
Clk	In	System clock for USES.
Reset_b	In	System reset. Initializes circuit. (Active low).
Enable_b	In	Tri-state enable for Out bus. Low enables data output. High is tristate.
PosNeg	In	Input data is pos if low, bipolar if high.
Jsel[1..0]	In	Select Blk Size J=8,10,16.
Nsel[3..0]	In	Select Quantization. N=4..15.
PreMod[1,0]	In	Select Prediction mode.
EntCode	In	Entropy code positive data if high. PosNeg and Bypass must be low.
RefRef	In	Ref. to ref. differences. Only in nearest neighbor and 2D. (High)
Bypass	In	Acts as a packetizer to form 16 bit output words.(Active High)
BlkRef[6..0]	In	Blocks between references 1..128.
RefPac[6..0]	In	Number of Refs in a packet. 1..128
EndPac	In	Will end packet in any mode. (Active High)
ConPac	In	Continuous Packets. RefPac ignored. (Active High).
ValBlk	In	Valid Block, high when valid data is input.
X[14..0]	In	Sample input data bus.
XP[14..0]	In	Predictor input bus.
Out[15..0]	Out	Output bus.
DataRdy	Out	Signals data is at output when high.
EOP	Out	Data at output is end of packet. (Active high).
Fill8	Out	High only at EOP and at least 8 bits of fill.
EndBits[4..0]	Out	Number of data Bits in last word of packet. 1..16.
Vdd	Pwr	Power Supply for Output Pads. Normally +5V.
VSS	Pwr	Ground Reference for Output Pads.
VddQ	Pwr	Power Supply for Inputs. Normally +5V. Quiet.
VssQ	Pwr	Ground Reference for Inputs. Quiet.
TestIn	In	Test Mode Input for Aeroflex.(Active High).
TestOut	Out	Test Output for Aeroflex.



8 USES Pinouts: Aeroflex's standard 196 pin flat-pack

01	VDD	50	VDDQ	99	VDD	148	VDDQ
02	VSS	51	VSSQ	100	VSS	149	VSSQ
03	EndBits0	52	EOP	101	EntCode	150	XP6
04	noconnect	53	noconnect	102	noconnect	151	noconnect
05	EndBits1	54	Fill8	103	ConPac	152	XP5
06	noconnect	55	VDD	104	noconnect	153	noconnect
07	EndBits2	56	TestOut	105	RefRef	154	XP4
08	noconnect	57	VSS	106	noconnect	155	noconnect
09	EndBits3	58	noconnect	107	PosNeg	156	XP3
10	noconnect	59	TestIn	108	VSSQ	157	noconnect
11	EndBits4	60	VSSQ	109	Nsel3	158	XP2
12	VDD	61	Enable_b	110	noconnect	159	noconnect
13	Out0	62	VSSQ	111	Nsel2	160	XP1
14	VSS	63	BlkRef6	112	noconnect	161	noconnect
15	Out1	64	noconnect	113	Nsel1	162	XP0
16	noconnect	65	BlkRef5	114	noconnect	163	noconnect
17	Out2	66	noconnect	115	Nsel0	164	X14
18	noconnect	67	BlkRef4	116	noconnect	165	noconnect
19	Out3	68	noconnect	117	PreMod1	166	X13
20	noconnect	69	BlkRef3	118	noconnect	167	noconnect
21	Out4	70	noconnect	119	PreMod0	168	X12
22	noconnect	71	BlkRef2	120	noconnect	169	noconnect
23	Out5	72	noconnect	121	VSSQ	170	X11
24	VDDQ	73	BlkRef1	122	Clk	171	VDDQ
25	VSSQ	74	VSSQ	123	VSSQ	172	VSSQ
26	OUT6	75	VDDQ	124	VDDQ	173	noconnect
27	noconnect	76	BlkRef0	125	Reset_b	174	X10
28	OUT7	77	noconnect	126	VSSQ	175	noconnect
29	noconnect	78	RefPac6	127	EndPac	176	X9
30	OUT8	79	noconnect	128	noconnect	177	noconnect
31	noconnect	80	RefPac5	129	ValBlk	178	X8
32	OUT9	81	noconnect	130	VSSQ	179	noconnect
33	noconnect	82	RefPac4	131	XP14	180	X7
34	OUT10	83	noconnect	132	noconnect	181	noconnect
35	VSS	84	RefPac3	133	XP13	182	X6
36	OUT11	85	noconnect	134	noconnect	183	noconnect
37	VDD	86	RefPac2	135	XP12	184	X5
38	OUT12	87	noconnect	136	noconnect	185	noconnect
39	noconnect	88	RefPac1	137	XP11	186	X4
40	OUT13	89	noconnect	138	noconnect	187	noconnect
41	noconnect	90	RefPac0	139	XP10	188	X3
42	OUT14	91	noconnect	140	noconnect	189	noconnect
43	noconnect	92	JSEL1	141	XP9	190	X2
44	OUT15	93	noconnect	142	noconnect	191	noconnect
45	noconnect	94	JSEL0	143	XP8	192	X1
46	noconnect	95	noconnect	144	noconnect	193	noconnect
47	DataRdy	96	Bypass	145	XP7	194	X0
48	VDD	97	VDDQ	146	VDD	195	VDDQ
49	VSS	98	VSSQ	147	VSS	196	VSSQ

9 Electrical Specifications

Reference is ground (Vss).

Absolute Maximum Ratings				
Symbol	Parameter	Min.	Max.	Units
Vdd	Supply Voltage	-0.3	7.0	V
Vi	Voltage at Digital Inputs	-0.3	Vdd+0.3	V
Ii	Current into Digital Inputs	-10	10	mA
Vo	Voltage at Digital Outputs	-0.3	Vdd+0.3	V
Io	Current into Digital Outputs	-10	10	mA
Tst	Storage Temperature	-65	150	Deg C

Recommended Operating Conditions					
Symbol	Characteristics	Min.	Max.	Units	Test Conditions
Vdd	Supply Voltage	4.5	5.5	V	
Vi	Input Voltage	0	Vdd	V	
Top	Operating Temperature	-55	+125	Deg C	

DC Electrical Characteristics					
Symbol	Characteristics	Min.	Max.	Units	Test Conditions
Idd	Supply Current	0	10	μA	25 C, static state, outputs unloaded
Vih	Input High Voltage(CMOS)	0.7Vdd	Vdd	V	All inputs
Vil	Input Low Voltage(CMOS)	Vss	0.3Vdd	V	All inputs
Iil	Input Leakage	-10	10	uA	
Voh	Output High Voltage	Vdd-0.05	Vdd	V	Ioh=-1.0μA
Vol	Output Low Voltage	Vss	Vss+0.05	V	Iol= 1.0μA
Ios	Short-circuit Output Current		100	mA	1 sec max duration, see Note 8.
P	Power Dissipation		0.092	W/MHz	Outputs at 50pF, see Note 8.

AC Parameters **AC Electrical Characteristics**					
Symbol	Parameter	Min.	Max.	Units	Test Conditions
tck	Clk Cycle Time	50.0		ns	Note 1
tckr	Clk Rise Time		4.0	ns	Note 2
tckf	Clk Fall Time		4.0	ns	Note 3
tckw	Clk Pulse Width	18.0		ns	Notes 1 and 4
tisu	Non-Clock Inputs Set Up	9.0		ns	Note 5
tih	Non-Clock Inputs Hold	2.0		ns	Note 6
tod	All Outputs Delay		33.0	ns	CL=50pF, Note 7
tzd	Enable/Disable OUTbus		21.0	ns	Tri-state control
tIcyc	Input sample cycle time	1		clks	
tOcyc	Output cycle time	1		clks	
tRstw	Reset_b Pulse Width Low	4		clks	
tInitsu	Control lines stable	2		clks	Before Reset_b high
tInith	Control lines stable	2		clks	After Reset_b high

Note:

1. Measured from valid Vil to valid Vil.
2. Measured from valid Vil to valid Vih.
3. Measured from valid Vih to valid Vil.
4. Measured from valid Vih to valid Vih.
5. Measured from valid Vil or Vih to invalid Vil (rising Clk edge).
6. Measured from valid Vih (rising Clk edge) to invalid Vih or Vil.
7. Measured from valid Vih (rising Clk edge) to valid Voh or Vol.
8. Guaranteed by design but not tested.

10 Timing

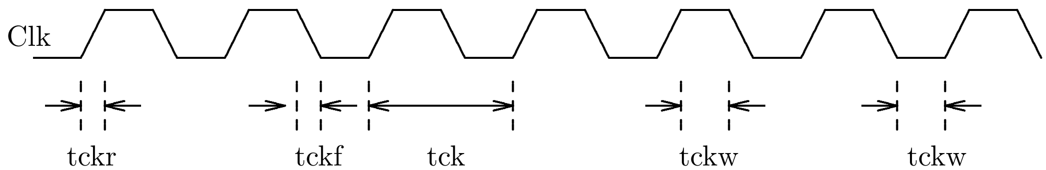


Figure 8: Clk timing

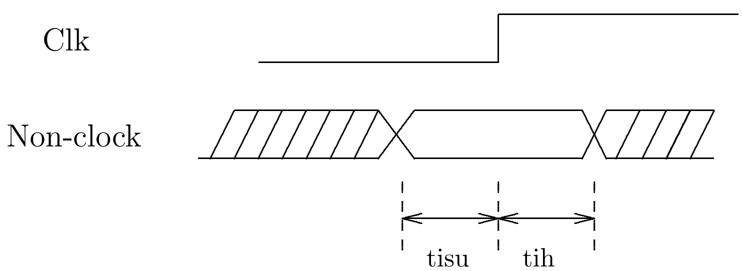


Figure 9: Input setup and hold times.

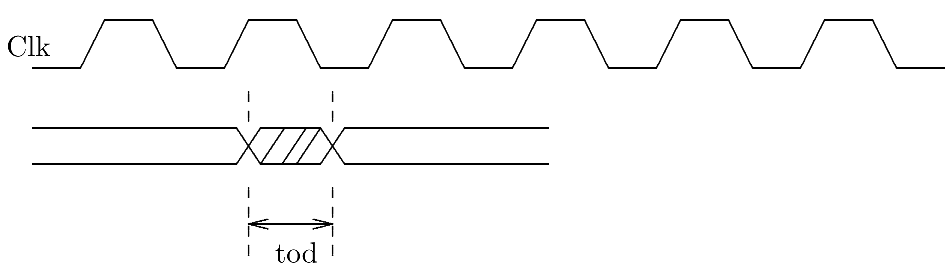


Figure 10: Output timing delays.

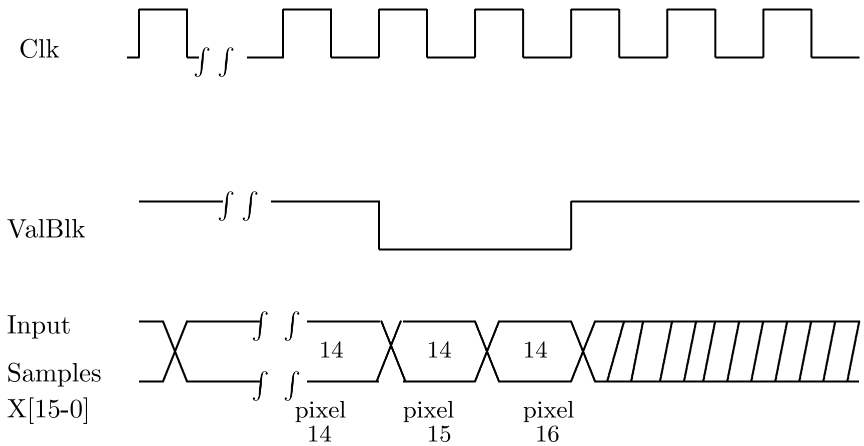


Figure 11: Using ValBlk to fill a scanline out to 16 samples by repeating the last valid sample.

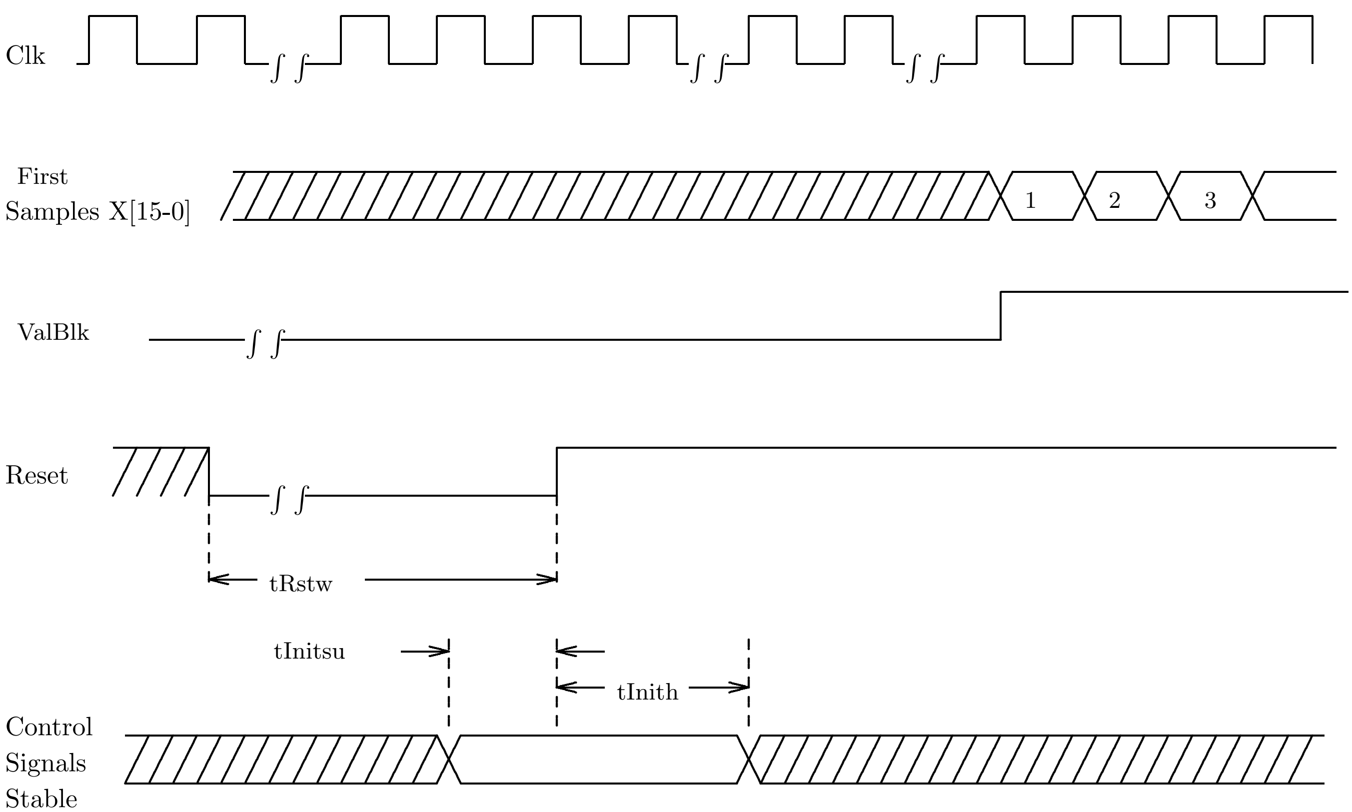


Figure 12: Initialization and arrival of first samples. ValBlk is held low during initialization to set USES in compression mode.

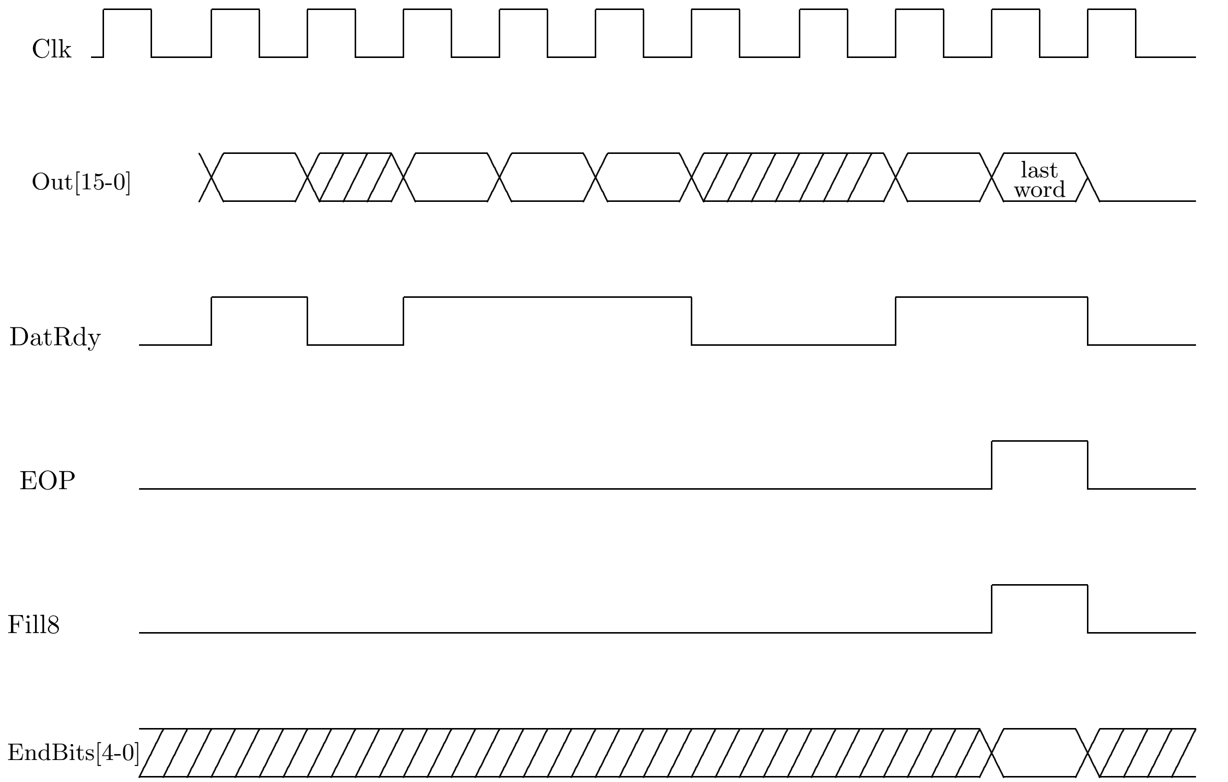


Figure 13: Outputting one packet of data. In this example, the last word of the packet contains at least 8 bits of fill.

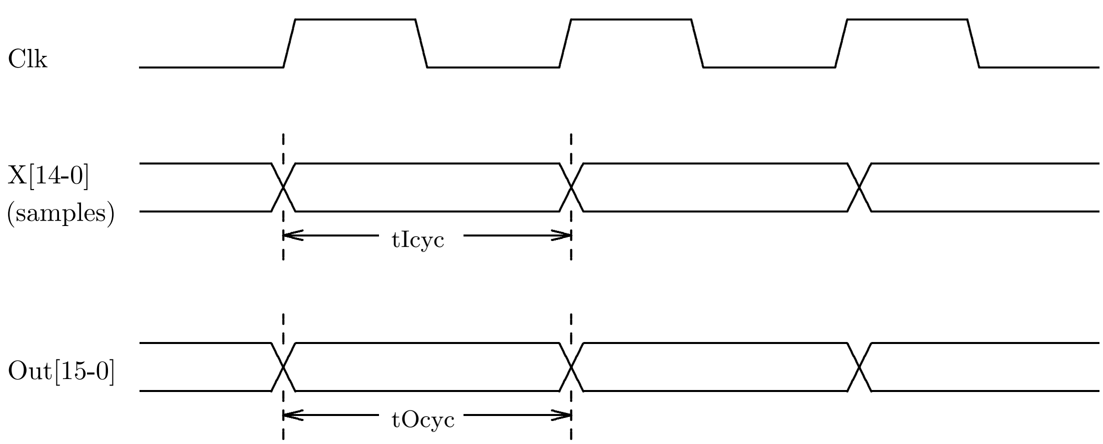


Figure 14: Input sample cycle time and Output cycle time.