Application Note AN3101-02: DSP-1K as an AES/EBU receiver
by Chris Maple

Introduction

The AES/EBU standard describes a digital serial method of stereo audio data transmission, using a physical transmission medium of a differential pair. The format provides many error-checking capabilities, including three precise preambles, transitions at every bit boundary, validity and parity bits with each channel's sample, and a CRC code for every block of 192 stereo samples or frames. There are also provisions for copyright protection. The format is:

4-bit Preamble,
4-bit Auxiliary Data or Least Significant Bits for audio data over 20 bits,
20-bit Audio Data, least significant bit first,
1-bit Validity, active low,
1-bit User Data,
1-bit Channel Status Data,
1-bit Parity, to make even parity with all preceding except preamble,
followed by the same bit assignments for a second channel.

The channel status data is accumulated over 192 frames, and includes, among other things, the copyright protection features, emphasis and word length information, and the CRC. The data is "Biphase-Mark", also known as "FM". There's a transition at the edge of every bit cell, and if the data bit is 1 there's a transition at the center also.

Algorithm

The DSP-1K may be used to perform some of the functions of an AES/EBU receiver. It cannot by itself synchronize to a raw AES/EBU input; this application note assumes that the WORDCLK input rises as the preamble for the first channel begins. One side of the differential input is applied to IN0 and DAC-compatible output is available at OUT0 (pin 11). One transition of the preamble is checked. Parity must be even and the validity bit must be low. Any error causes the output to be muted for the next 64 sample periods. Both input channels must be good or both output channels will be muted. The choice of IN0 as the input pin is deliberate because if two successive reads of $419 (direct in) are added, bit 20 in the accumulator is 1 if and only if IN0 is different between the two reads. This is not true for other inputs.

Since the DSP-1K does not have an XOR instruction, CRC accumulation is difficult, and consequently the CRC is calculated only for the left channel. User bits are placed on OUT1 and Left Channel Status bits are placed on OUT2 as received. CRC failure is placed on OUT3. Copyright data is in the Channel Status bits. Also, this application note does not check that there is a transition at every bit boundary. **IT IS THE RESPONSIBILITY OF ANY DESIGNER OF AES/EBU EQUIPMENT TO MAKE SURE THAT THE EQUIPMENT IS DESIGNED IN COMPLIANCE WITH THE LAW.**

The audio data is assembled into addresses $400 and $401. Parity is accumulated in $402 and $403. The error countdown is kept in $404. CRC is accumulated in bits 16-9 of $405, with bit 9 holding the "most significant" CRC bit. The frame count in each block is maintained in $406. $407 is temporary storage for the current Channel Status bit. Calculations from each channel's sample spill over into the preamble of the next channel's sample. 24 bit audio data is assumed. Control Word 0, bits 1-0 need to be set to 11 to enable 24 bit I/O.

Source Code

```
I#  INSTRUCTION             COMMENT
0   SKIP  Z 2               ;skip if not zero (no parity error)
1   C     0x40                 ;load 64 into the accumulator
2   SCA   1.0 0x404            ;store 64 to the error countdown to start new countdown
3   CM    1.0 0x404         ;load the error countdown count
4   SKIP  Z 5               ;if no error active, skip
5   1AC   0xFFFFFFF            ;subtract 1 from the error downcount
6   SCA   1.0 0x404            ;save new error downcount
7   C     0                    ;load 0
8   SCA   1.0 0x400            ;store 0 to first channel to mute it
9   SCA   1.0 0x401            ;store 0 to second channel to mute it
A   CM    1.0 0x401         ;get second channel data
B   SCA   1.0 0x411         ;write to output
C   CM    1.0 0x400         ;get first channel data
D   SCA   1.0 0x410         ;write to output
E   SKIP  6                 ;next section checks for first transition after preamble start
; -----------             NEW CYCLE - FIRST CHANNEL STARTS HERE
15  LCM   0x040000 0x419    ;get the input into the B register
16  SKIP  6
; -----------
1D  CMB   1.0 0x419         ;input + B
1E  ANDC  0x0100000         ;should be 1 in bit 20
1F  SKIP  Z 2               ;if not,
20  C     0x40                 ;64
21  SCA   1.0 0x404            ;goes to the error countdown
22  C     0                 ;
23  SCA   1.0 0x400         ;initialize OUT0 left accumulation area
24  SCA   1.0 0x401         ;initialize OUT0 right accumulation area
25  CMB   1.0 0x419         ;"Preamble Z" (block start) <=> input same as instruction $15
26  ANDC  0x0100000         ;isolate bit 20 for XOR
27  SKIP  !Z 4              ;skip if not start
28  C     0x001FE00           ;CRC starts with all ones
29  SCA   0 0x405              ;save CRC, zero acc
2A  SCA   1.0 0x406            ;save sample count
2B  SKIP  0x19                ;goto start of data accumulation
2C  CM    1.0 0x406            ;get sample count
2D  1AC   0x0000001            ;increment sample count
2E  SCA   1.0 0x406            ;save new sample count
2F  SKIP  0x15              ;goto start of data accumulation
; -----------
45  LCM   1.0 0x419         ;get input data into B register  START DATA AREA
46  SKIP  6
; -----------
4D  CMB   1.0 0x419         ;input + B
4E  ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
4F  CAD   4.0 0             ;times 4 (moves data into bit 22)
50  SCA   2.0 0x402         ;save data as 1st parity count, *2 moves data to bit 23
51  SCA   1.0 0x400         ;save first bit (lsb)
52  SKIP  2
; -----------
55  LCM   1.0 0x419         ;get input data into B register  SUBSEQUENT DATA BITS
56  SKIP  6
; -----------
5D  CMB   1.0 0x419         ;input + B
5E  ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
5F  CAD   4.0 0             ;times 4
60  XCMA  1.0 0x402         ;save data in B, add data to parity counter
61  SCB   2.0 0x402         ;store new parity counter;  2*B moves data into bit 23
62  CMA   0.5 0x400         ;acc+[$400]*0.5;  shift old data right and insert new bit
63  SCA   1.0 0x400         ;store accumulated data
64  SKIP  0
; -----------   REPEAT THE CODE IN 55-64 22 MORE TIMES TO GET 24 BITS TOTAL (NOT SHOWN)
```

```
1C5 LCM   1.0 0x419         ;get first part of validity bit into B register
1C6 SKIP  6
1CD CMB   1.0 0x419         ;input + B                                   VALIDITY BIT
1CE ANDC  0x0100000         ;isolate bit to get XOR, 1 = change = not valid audio
1CF SKIP  Z 4               ;skip if valid
1D0 CAM   4.0 0x402            ;4*acc+[$402] add to parity counter (bit 22)
1D1 SCA   1.0 0x402            ;store new parity counter
1D2 C     0x0000040            ;
1D3 SCA   1.0 0x404            ;store 64 to the error downcounter
1D4 SKIP  0                 ;
1D5 LCM   1.0 0x419         ;get first part of user bit into B register
1D6 SKIP  6
; -----------
1DD CMB   1.0 0x419         ;input + B                                   USER BIT
1DE ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
1DF CAD   4.0 0             ;times 4
1E0 XCMA  1.0 0x402         ;save data in B, add data to parity counter
1E1 SCB   0.5 0x402         ;store new parity counter, 0.5*B moves data to bit 21
1E2 SCA   1.0 0x422         ;send user bit to OUT1
1E3 SKIP  1
; ----------
1E5 LCM   1.0 0x419         ;get first part of Channel Status bit into B register
1E6 SKIP  6
; ----------
1ED CMB   1.0 0x419         ;input + B                               CHANNEL STATUS BIT
1EE ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
1EF SXCA  4.0 0x407         ;save Channel Status bit at $407 bit 20, also reg B; then acc*4
1F0 SCA   1.0 0x424         ;transmit Channel Status bit at OUT2
1F1 CBM   4.0 0x402         ;4*B+[$402] add bit to parity counter
1F2 SCA   1.0 0x402         ;store new parity counter
1F3 SKIP  1
; -----------
1F5 LCM   1.0 0x419         ;get first part of parity bit into B register
1F6 SKIP  6
; ----------
1FD CMB   1.0 0x419         ;input + B                                 PARITY BIT
1FE ANDC  0x0100000         ;isolate parity bit
1FF CAM   4.0 0x402         ;[$402]+4.0*acc;  add parity bit to parity count
200 ANDC  0x0400000         ;isolate lsb of parity count to get parity
201 SKIP  Z 2               ;skip if even (passes parity test)
202 C     0x40                 ;load 64 into accumulator
203 SCA   1.0 0x41             ;store 64 to the error downcounter
204 SKIP  0x10
; -----------             SECOND CHANNEL STARTS HERE
215 LCM   1.0 0x419         ;get the input into the B register
216 SKIP  6
; -----------
21D CMB   1.0 0x419         ;input + B
21E ANDC  0x0100000         ;should be 1 in bit 20
21F SKIP  Z 2               ;if not,
220 C     0x40                 ;64
221 SCA   1.0 0x404            ;64 goes to the error countdown
222 CM    1.0 0x406         ;read sample number
223 1AC   0xFFFFF48         ;subtract 184
224 SKIP  !N 0xB            ;skip if last 8 blocks
225 CM    1.0 0x405            ;existing CRC in bits 16-9
226 CMA   0x000080 0x407       ;add Channel Status bit (20->9) to oldest CRC bit (9)
227 ANDC  0x0000200            ;isolate the XOR at bit 9
228 C     !Z 0x0240A00         ;for CRC 8-4-3-2-0, bits 21-9
229 XCM   0x101000 0x405       ;save to B; replicate CRC bits 16-9 in acc 18-11 and 10-3
22A ANDC  0x00552A80           ;mask even bits 18-11, odd bits 10-3
22B CAB   4.0                  ;add in saved bits & move up to acc 20-13 and 12-5. 21 also
22C ANDC  0x0354AA0            ;21-5. Zeroes mask out potential carries, leaving only XORs
22D CAD   0x101000 0x0         ;times hexadecimal 4.04 merges fields. 15-7 of interest now
22E CAD   2.0 0                ;shift result into 16-8
22F SKIP  0xA                  ;past code for last 8 blocks
230 SKIP  !Z 2                 ;if 184th block
231 C     0x0                     ;clear CRC error, if any
232 SCA   1.0 0x428               ;OUT3
233 CM    0x000080 0x407       ;get current Channel Status bit, move down to bit 9
234 CMA   1.0 0x405            ;add CRC and Channel Status bit
235 ANDC  0x0000200            ;mask for XOR at bit 9
236 SKIP  Z 2                  ;Z is good; no CRC error
```

```
237 C     0x0800000               ;create CRC error bit
238 SCA   1.0 0x428               ;transmit error bit   OUT3
239 CM    0.5 0x405            ;get CRC; shift right 1
23A ANDC  0x001FE00         ;mask for CRC field
23B SCA   1.0 0x405         ;save new CRC
23C SKIP  8                 ;goto start of data accumulation
; -----------
245 LCM   1.0 0x419         ;get input data into B register  START DATA AREA
246 SKIP  6
; -----------
24D CMB   1.0 0x419         ;input + B
24E ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
24F CAD   4.0 0             ;times 4 (moves data into bit 22)
250 SCA   2.0 0x403         ;save data as 1st parity count, *2 moves data to bit 23
251 SCA   1.0 0x401         ;save first bit (lsb)
252 SKIP  2
; -----------
255 LCM   1.0 0x419         ;get input data into B register  SUBSEQUENT DATA BITS
256 SKIP  6
; -----------
25D CMB   1.0 0x419         ;input + B
25E ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
25F CAD   4.0 0             ;times 4
260 XCMA  1.0 0x403         ;save data in B, add data to parity counter
261 SCB   2.0 0x403         ;store new parity counter;  2*B moves data into bit 23
262 CMA   0.5 0x401         ;acc+[$400]*0.5;  shift old data right and insert new bit
263 SCA   1.0 0x401         ;store accumulated data
264 SKIP  0
; -----------   DO 255-264 22 MORE TIMES TO GET 24 BITS TOTAL (NOT SHOWN)
3C5 LCM   1.0 0x419         ;get first part of validity bit into B register
3C6 SKIP  6
3CD CMB   1.0 0x419         ;input + B                                    VALIDITY BIT
3CE ANDC  0x0100000         ;isolate bit to get XOR, 1 = change = not valid audio
3CF SKIP  Z 4               ;skip if valid
3D0 CAM   4.0 0x403            ;4*acc+[$403] add to parity counter (bit 22)
3D1 SCA   1.0 0x403            ;store new parity counter
3D2 C     0x0000040            ;
3D3 SCA   1.0 0x404            ;store 64 to the error downcounter
3D4 SKIP  0                 ;
3D5 LCM   1.0 0x419         ;get first part of user bit into B register
3D6 SKIP  6
; -----------
3DD CMB   1.0 0x419          ;input + B                                    USER BIT
3DE ANDC  0x0100000          ;isolate bit to get XOR, 1 = change
3DF CAD   4.0 0             ;times 4
3E0 XCMA  1.0 0x403         ;save data in B, add data to parity counter
3E1 SCB   0.5 0x403         ;store new parity counter, 0.5*B moves data to bit 21
3E2 SCA   1.0 0x422         ;send user bit to OUT1
3E3 SKIP  1
; ----------
3E5 LCM   1.0 0x419         ;get first part of Channel Status bit into B register
3E6 SKIP  6
; ----------
3ED CMB   1.0 0x419         ;input + B                                CHANNEL STATUS BIT
3EE ANDC  0x0100000         ;isolate bit to get XOR, 1 = change
3EF CAM   4.0 0x403         ;4*acc+[$403] add bit to parity counter
3F0 SCA   1.0 0x403         ;store new parity counter
3F1 SKIP  3                 ;Right Channel Status is NOT saved or transmitted
; -----------
3F5 LCM   1.0 0x419         ;get first part of parity bit into B register
3F6 SKIP  6
; ----------
3FD CMB   1.0 0x419         ;input + B                                 PARITY BIT
3FE ANDC  0x0100000         ;isolate parity bit
3FF CAM   4.0 0x403         ;[$403]+4.0*acc;  add parity bit to parity count
; -----------    START AGAIN AT INSTRUCTION 0
```

NOTICE

Wavefront Semiconductor reserves the right to make changes to their products or to discontinue any product or service without notice. All products are sold subject to terms and conditions of sale supplied at the time of order acknowledgement. Wavefront Semiconductor assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Information contained herein is only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked, no responsibility is assumed for inaccuracies.

Wavefront Semiconductor products are not designed for use in applications which involve potential risks of death, personal injury, or severe property or environmental damage or life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness.

Contact Information:

Wavefront Semiconductor
200 Scenic View Drive
Cumberland, RI 02864 U.S.A.
Tel: +1 401 658-3670
Fax: +1 401 658-3680
On the web at www.wavefrontsemi.com
Email: info@wavefrontsemi.com

Application note revised March, 2005