The following is a list of known errata in ATSC Document A/52, "Digital Audio Compression Standard (AC-3)", dated December 20, 1995. This list is maintained by Dolby Laboratories for informational purposes only, and is not an official document of the Advanced Television Systems Committee. To submit additional items for inclusion on subsequent versions of this list, please send errata to website@dolby.com.

In the following descriptions, bitstream variables and pseudo code are shown in the Arial typeface, while text taken directly from the ATSC document is shown in *italics*. Specific errata are highlighted in **bold**.

Section	Description
5.3.2 page 14	In the syntax for the BSI elements, the language code exists bit for dual mono channel 2 is incorrectly shown as **lngcod2e**. The correct spelling should be **langcod2e**.
5.3.3 page 18	The bit stream syntax for quantized mantissa values is somewhat confusing, and in fact contains a bug (shown in bold below, should be **chincpl[ch - 1]**): ch = 0 do { for (bin = 0; bin < nchmant[ch]; bin++) {chmant[ch][bin]} ..... (0-16) ch += 1 } while (**chinclp[ch]** == 0 && ch < nfchans) if (cplinu) { for (bin = 0; bin < ncplmant; bin++) {cplmant[bin]} ..... (0-16) } while (ch < nfchans) { for (bin = 0; bin < nchmant[ch]; bin++) {chmant[ch][bin]} ..... (0-16) ch += 1 } if (lfeon) { for (bin = 0; bin < nlfemant; bin++) {lfemant[bin]} ..... (0-16) } This syntax can be more easily expressed as follows: got_cplchan = 0 for (ch = 0; ch < nfchans; ch++) { for (bin = 0; bin < nchmant[ch]; bin++) {chmant[ch][bin]} ..... (0-16) if (cplinu && chincpl[ch] && !got_cplchan) { for (bin = 0; bin < ncplmant; bin++) {cplmant[bin]} ..... (0-16) got_cplchan = 1 } } if (lfeon) { for (bin = 0; bin < nlfemant; bin++) {lfemant[bin]} ..... (0-16) }

5.4.2.4 page 21	In Table 5.4, the linear value for the center mix level coefficient corresponding to the binary value '01' is specified as ***0.596*** *(-4.5 dB).* In fact, this value should be ***0.595***, or the fourth root of 1/8.
5.4.3.7 page 27	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.4.3.13 page 28	The text description in this section reads in part as follows: *The number of coupling bands,* ncplbnd*, may be computed from* ncplsubnd *and* ***cplbnstrc****:* *ncplbnd = (ncplsubnd - (cplbndstrc[**cplbegf+1**] + ... + cplbndstrc[**cplendf+2**]));* This is incorrect, as it implies that the coupling band structure array contains valid elements in the index range from cplbegf+1 through cplendf+2. In fact, this array only contains valid elements in the index range from 1 through ncplsubnd-1, as shown in the bitstream syntax on page 15. This sentence should be rewritten as follows: *The number of coupling bands,* ncplbnd*, may be computed from* ncplsubnd *and* ***cplbndstrc****:* *ncplbnd = (ncplsubnd - (cplbndstrc[**1**] + ... + cplbndstrc[**ncplsubnd-1**]));*
5.4.3.14 page 28	The text description in this section reads in part as follows: ***All coupling coordinates are always transmitted in block 0 of each syncframe.*** This is incorrect, as it implies that coupling is always enabled for all channels in block 0. This sentence should be rewritten as follows: ***This parameter may not be set to 0 in block 0, or in any block for which the corresponding channel is participating in coupling but was not participating in coupling in the previous block.***
5.4.3.19 page 29	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.4.3.21 page 30	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0, or in any block for which coupling is enabled but was disabled in the previous block.*
5.4.3.22 page 30	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.4.3.23 page 30	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.4.3.30 page 31	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*

5.4.3.36 page 32	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.4.3.44 page 32	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0, or in any block for which coupling is enabled but was disabled in the previous block.*
5.4.3.47 page 33	The text description in this section reads in part as follows: *If deltbaie is 0 in block 0, then **cpldeltnseg and deltnseg[ch] are set to 0**, and no delta bit allocation is applied.* This is somewhat confusing, as the variables cpldeltnseg and deltnseg[ch] are defined to only take on values between 1 and 8. To clarify its meaning, this sentence should be rewritten as follows: *If deltbaie is 0 in block 0, then **cpldeltbae and deltbae[ch] are set to the binary value '10'**, and no delta bit allocation is applied.*
5.4.3.48 page 33	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0, or in any block for which coupling is enabled but was disabled in the previous block.*
5.4.3.49 page 33	The following text should be added to the end of this paragraph: *This parameter may not be set to 0 in block 0.*
5.5 page 37	The first item in the list of bit stream constraints reads as follows: *1. **The size of block 0 and block 1 combined,** will never exceed 5/8 of the frame.* This is not quite correct, as it fails to include the syncinfo and BSI fields. This item should read as follows: *1. **The combined size of the syncinfo fields, the bsi fields, block 0, and block 1** will never exceed 5/8 of the frame.* The second item in the list reads as follows: *2. **The sum of block 5 mantissa data and auxiliary data** will never exceed the final 3/8 of the frame.* This is not quite correct, as it fails to include the errorcheck fields. This item should read as follows: *2. **The combined size of the block 5 mantissa data, the auxiliary data fields, and the errorcheck fields** will never exceed the final 3/8 of the frame.* The list of bit stream constraints should be extended to include the following: *5. Coupling will never be used in dual mono (1+1) or mono (1/0) mode. For blocks in which coupling is used, there will always be at least two channels in coupling.*

	6. *Bit stream elements will not be reused from a previous block if other bit stream parameters change the dimensions of the elements to be reused. For example, exponents will not be reused if the start or end mantissa bin changes from the previous block.*
7.2.2.1 page 52-53	The generalized pseudo code for computing the snroffset for each full bandwidth channel, the coupling channel, and the LFE channel is shown in part as follows: snroffset = ((csnroffst - 15) << 4 + fsnroffst) << 2; The desired computation is that (csnroffst - 15) should be shifted by 4, then added to fsnroffst, and then the result shifted by 2. However, the rules of precedence in the C programming language, on which our pseudo code is based, specify that the operation of addition takes precedence over the operation of shifting. In order to conform to these rules of precedence, the correct pseudo code should be: snroffset = (**(**(csnroffst - 15) << 4**)** + fsnroffst) << 2; The pseudo code for initializing the fastleak and slowleak values for the coupling channel is shown in part as follows: **if (cplleake)** **{** fastleak = (cplfleak << 8) + 768; slowleak = (cplsleak << 8) + 768; **}** This implies that the initialization of fastleak and slowleak should only occur if cplleake is nonzero. Although it is true that the cplfleak and cplsleak parameters will only be resent in the bitstream when cplleake is nonzero, the initialization should occur whenever coupling is in use. The correct pseudo code should be: fastleak = (cplfleak << 8) + 768; slowleak = (cplsleak << 8) + 768
7.2.2.3 page 54	The pseudo code for PSD integration is shown in part as follows: j = start; k = masktab[start]; do { bndpsd[k] = psd[j]; j++; **for (i = j; i < min(bndtab[k + 1], end); i++)** { bndpsd[k] = logadd(bndpsd[k], psd[j]); j++; } k++; } **while (end > bndtab[k++]);** This erroneously shows the k index being incremented twice per loop. It also requires that the bndtab[] table contain one more element than is defined in Table 7.12, to accomodate an index of k+1. The correct pseudo code should be:

	j = start; k = masktab[start]; do { **lastbin = min(bndtab[k] + bndsz[k], end);** bndpsd[k] = psd[j]; j++; **for (i = j; i < lastbin; i++)** { bndpsd[k] = logadd(bndpsd[k], psd[j]); j++; } k++; } **while (end > lastbin);**
7.2.2.4 page 54	The pseudo code for computing the bit allocation excitation function is shown in part as follows: for (bin = 2; bin < 7; bin++) { lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin); fastleak = bndpsd[bin] - fgain; slowleak = bndpsd[bin] - sgain; excite[bin] = fastleak - lowcomp; if (bndpsd[bin] <= bndpsd[bin+1]) { begin = bin + 1; break; } } for (bin = begin; bin < min(bndend, 22); bin++) { lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin); fastleak -= fdecay; fastleak = max(fastleak, bndpsd[bin] - fgain); slowleak -= sdecay; slowleak = max(slowleak, bndpsd[bin] - sgain); excite[bin] = max(fastleak - lowcomp, slowleak); } This is somewhat confusing, as the note regarding the last bin of the lfe channel is not reflected in the pseudocode. Also, the comparison of bndpsd[bin] to bndpsd[bin+1] should not be made for the last bin of the lfe channel. The correct pseudo code should be: for (bin = 2; bin < 7; bin++) { **if ((bndend != 7) || (bin != 6)) /* skip for last bin of lfe channel */** **{** lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin); **}** fastleak = bndpsd[bin] - fgain; slowleak = bndpsd[bin] - sgain; excite[bin] = fastleak - lowcomp; **if ((bndend != 7) || (bin != 6)) /* skip for last bin of lfe channel */**

```
    {
        if (bndpsd[bin] <= bndpsd[bin+1])
        {
            begin = bin + 1;
            break;
        }
    }
}
for (bin = begin; bin < min(bndend, 22); bin++)
{
    if ((bndend != 7) || (bin != 6))  /* skip for last bin of lfe channel */
    {
        lowcomp = calc_lowcomp(lowcomp, bndpsd[bin], bndpsd[bin+1], bin);
    }
    fastleak -= fdecay;
    fastleak = max(fastleak, bndpsd[bin] - fgain);
    slowleak -= sdecay;
    slowleak = max(slowleak, bndpsd[bin] - sgain);
    excite[bin] = max(fastleak - lowcomp, slowleak);
}
```

7.2.2.7
page 57

The pseudo code for computing the bit allocation pointer array is shown as follows:

```
i = start;
j = masktab[start];
do
{
    mask[j] -= snroffset;
    mask[j] -= floor;
    if (mask[j] < 0)
    {
        mask[j] = 0;
    }
    mask[j] &= 0x1fe0;
    mask[j] += floor;
    for (k = i; k < min(bndtab[j] + bndsz[j], end); k++)
    {
        address = (psd[i] - mask[j]) >> 5;
        address = min(63, max(0, address));
        bap[i] = baptab[address];
        i++;
    }
}
while (end > bndtab[j++]);
```

This requires that the bndtab[] table contain one more element than is defined in Table 7.12, to accomodate the comparison at the end of the while loop. The correct pseudo code should be:

```
i = start;
j = masktab[start];
do
{
    lastbin = min(bndtab[j] + bndsz[j], end);
    mask[j] -= snroffset;
    mask[j] -= floor;
```

	if (mask[j] < 0) { mask[j] = 0; } mask[j] &= 0x1fe0; mask[j] += floor; **for (k = i; k < lastbin; k++)** { address = (psd[i] - mask[j]) >> 5; address = min(63, max(0, address)); bap[i] = baptab[address]; i++; } **j++;** } **while (end > lastbin);**
7.7.1.2 page 79	The text description in this section reads in part as follows: *The combination of X and Y values allows dynrng to indicate gain changes **from 24.08 - 0.14 = +23.94 dB, to -18.06 - 6 = -24.06 dB**.* Not all of these terms are correct to the second decimal place. This sentence should read as follows: *The combination of X and Y values allows dynrng to indicate gain changes **from 24.08 - 0.14 = +23.95 dB, to -18.06 - 6.02 = -24.08 dB**.* Note that the first result is +23.95 dB (not 23.94 dB), since the extended precision equation is actually 24.08239965 - 0.13678849 = 23.94561116 dB.
7.7.2.2 page 81	The title of Table 7.30, ***Meaning of 3 msb of compr***, should be ***Meaning of 4 msb of compr***. Also, the text description in this section reads in part as follows: *The combination of X and Y values allows compr to indicate gain changes **from 48.16 - 0.28 = +47.88 dB, to -42.14 - 6 = -48.14 dB**.* Not all of these terms are correct to the second decimal place. This sentence should read as follows: *The combination of X and Y values allows compr to indicate gain changes **from 48.16 - 0.28 = +47.89 dB, to -42.14 - 6.02 = -48.16 dB**.* Note that the first result is +47.89 dB (not 47.88 dB), since the extended precision equation is actually 48.16479931 - 0.27576569 = 47.88903362 dB.
7.10.2 page 97	Several items in the list of known bit stream error conditions are incomplete and need further restriction in order to satisfy the bit stream constraints added to section 5.5. Item 2 is shown as follows: 2) (cplinu == 1) && (**no channels in coupling**);

	Since use of coupling requires at least two channels in coupling, item 2 should be rewritten as: 2) (cplinu == 1) && (**fewer than two channels in coupling**); Item 9 is shown as follows: 9) **(cplinu == 1) &&** **(cplbegf != previous cplbegf) &&** **(chincpl[n] == 1) &&** (chexpstr[n] == 0); This restriction addresses the case where the number of exponents changes while the exponents themselves are reused. This can be simplified, as well as generalized to include the case where coupling is disabled, by rewriting item 9 as follows: 9) **(nchmant[n] != previous nchmant[n]) &&** (chexpstr[n] == 0); Also, the list of bit stream error conditions should be extended to include the following: 22) (cplinu == 1) && (acmod < 2); 23) (cplinu == 1) && ((cplbegf != previous cplbegf) || (cplendf != previous cplendf)) && (cplcoe[n] == 0); 24) (cplinu == 1) && (cplbndstrc != previous cplbndstrc) && (cplcoe[n] == 0); 25) (acmod == 2) && (number of rematrixing bands != previous number of rematrixing bands) && (rematstr == 0); 26) (cplinu == 1) && (previous cplinu == 0) && ((deltbaie == 0) || (cpldeltbae == 0)); 27) (cplinu == 1) && ((cplbegf != previous cplbegf) || (cplendf != previous cplendf)) && (previous cpl delta bit allocation active) && ((deltbaie == 0) || (cpldeltbae == 0)); 28) (nchmant[n] != previous nchmant[n]) && (previous delta bit allocation for channel n active) && ((deltbaie == 0) || (deltbae[n] == 0));
8.2.2 page 100	The text description in this section reads in part as follows: *1) High-pass filtering: The high-pass filter is implemented as a cascaded biquad **direct form II** IIR filter with a cutoff of 8 kHz.* This is incorrect, as the filter type is erroneously specified. The correct text should be:

	*1) High-pass filtering: The high-pass filter is implemented as a cascaded biquad **direct form I** IIR filter with a cutoff of 8 kHz.*
8.2.5.2 page 102	The text description in this section begins as follows: *Coupling coordinates are formed by taking **power** ratios within of each coupling band. The power in the original channel within a coupling band is divided by the power in the coupling channel within the coupling band. This **power** ratio becomes the coupling coordinate.* This text incorrectly states that the coupling coordinates should be formed by taking a power ratio, when in fact the correct operation is a magnitude ratio. The correct result can be derived by taking the square root of the power ratio. The correct text should be: *Coupling coordinates are formed by taking **magnitude** ratios within each coupling band. The power in the original channel within a coupling band is divided by the power in the coupling channel within the coupling band**, and the square root of this result is then computed**. This **magnitude** ratio becomes the coupling coordinate.*
8.2.12 page 104	The text description in this section reads in part as follows: *The coarse SNR offset adjusts in **6** dB increments, and the fine offset adjusts in **3/8** dB increments.* In fact, these values are not correct, and should each be half as large. The correct text should be: *The coarse SNR offset adjusts in **3** dB increments, and the fine offset adjusts in **3/16** dB increments.*