THE AES/EBU DIGITAL AUDIO INTERFACE

John Watkinson

Consultant, U.K.

Many of the advantages of digital audio recording and processing are lost if it is necessary to return to the analogue domain in order to connect different pieces of equipment together. It was the goal of the AES/EBU interface to provide a standard means for interconnection between digital audio devices which would be flexible enough to embrace the majority of requirements, and reasonably simple to implement.

At the time that the AES/EBU interface was specified, many incompatible systems had sprung up in an ad hoc manner to meet immediate needs. The majority of these required separate clock signals and so several conductors needed to pass from one device to another. Although this is only an inconvenience when equipment is adjacent, it becomes unacceptable in complex installations where distances may be greater and where a routing system will be used. A separate clock signal doubles the complexity of the router. An early decision was that the AES/EBU interface would be self-clocking so that it could utilise a single cable.

New technologies seldom have total freedom in a real world, because they inherit the environment developed for previous technologies. The greater the compatibility with existing methods, the greater the acceptance of a new approach is likely to be. In most audio installations, there is considerable investment in analogue balanced line twisted pair cabling and XLR-type connectors. The AES/EBU interface was designed to be able to use such cabling, so that there would essentially be no difference between analogue cable and a digital cable - a tremendous simplification.

Clearly in a single channel, the bits making up a sample of digital audio must be sent serially using some form of Pulse Code Modulation (PCM), and if the wordlength and the sampling rate to be used are multiplied together, the resulting bit rate can be obtained, which is typically about one megabit per second per audio channel. This represents a frequency immensely higher than that for which the cabling was designed, and it is not immediately obvious that it will work.

A typical audio cable consists of a pair of conductors which are closely twisted together to achieve an approximation to them being in the same place. If this is the case, electromagnetically induced hum voltages will be identically induced in both conductors, which the differential receiver will reject as a common-mode signal.

When high speed pulses propagate, they do not flow in the conductor alone, but travel as a packet of energy which is partly in the conductor, partly in the ground return, and partly in the dielectric between the conductors, which determines the impedance of the resulting transmission line. One common way of propagating such pulses is to twist tightly together the pair of conductors, since this gives a constant spacing between them and so a constant impedance. As this is the construction of an audio cable, it ought to work reasonably well as a pulse cable, and this is indeed the case, although the XLR connector is less than perfect as it was never designed to match the impedance of the cable at these frequencies - it was designed to be unbreakable.

At high frequencies, the action of the dielectric in a cable causes losses which change with the frequency, and so the cable has a frequency response which is not constant. It is helpful to restrict the range of frequencies which are sent, as this will then restrict the range of responses. It is also helpful because a signal with a restricted spectrum can be made to be free of a DC component, so that it will pass through coupling capacitors and isolating transformers.

Fig. 1 shows the electrical interface of the AES/EBU interface. The driver and receiver chips are the same as those used in data communications complying with RS422/CCITT-V.11. These can be directly connected together, but work by the BBC suggested that for longer cable runs, benefit would be had by incorporating equalisation and transformers. In the form of the AES interface adopted by the EBU, the transformers are mandatory.

The transmission down the above electrical interface is by means of a channel code which is designed to make the transmission self-clocking, free of DC and have a restricted bandwidth irrespective of the audio data content. Raw audio data cannot be sent, an obvious reason being that, in the case of digital audio muting, continuous zeros are transmitted. The signal never changes, and the receiver would soon lose lock. A channel code generates changes, known as transitions, even when there are none in the data, in order to retain the ability to clock the receiver.

There are many channel codes, of varying complexity and performance, and the AES/EBU interface uses FM channel code which is robust and simple to implement. The coding rules for FM are simple, and some examples are given in Fig. 2. Every data bit period is begun with a transition, and if the data bit is a one, there will be an additional transition in the centre of the data bit period. The result is that clock content is assured irrespective of data content, and the transmitted signal has only two frequencies, an octave apart. The low frequency results from continuous zeros, and the high frequency results from continuous ones. It will be seen that the signal spends as much time low as it does high, and so has no DC component. The high frequency is numerically equal to the bit rate.

Tests showed that typical audio cables were sufficiently loss free to allow the operating frequency to be doubled so that two simultaneous independent audio channels can be carried down one channel. The two audio channels can be a stereo pair, or completely unrelated in programme content, but they must have the same sampling rate.

Fig. 3 shows that one sample period, i.e. the time between instants when samples are taken, is divided first into two subframes, one for each audio channel, then each subframe is divided into 32 bit periods, so that the bit rate of the AES/EBU transmission is 64 times the sampling rate.

A variety of standard sampling rates exist, but these can be varied, for example in the case where the speed of a digital recorder is changed. Accordingly the AES/EBU interface simply runs at the instantaneous sampling rate with which it is being used. A phase-locked loop may well be used to generate the bit clock, and this simply multiplies the incoming sampling rate, whatever it may be, by 64. Clearly communication can only be achieved if the receiving device is capable of operating at, and locking to the sampling rate of the sending device, so one should not expect to make digital connections between devices of different sampling rate, or between devices where the sampling clock of one cannot become the slave of the other.

In a multiplexed serial transmission of this kind, synchronisation is vital, since synchronisation of the subframes allows the two channels to be kept apart, and synchronisation of the bits within the frames allows the correct reassembly of parallel samples with the most significant bit at the top. To achieve synchronisation it is necessary to have unique markers which are periodically inserted in the serial data stream and which specify where a subframe begins. It is not satisfactory to use a combination of data bits, since there is no way that such a combination can be prevented from occurring in the audio samples. The solution is that the synchronisation patterns violate the rules of FM coding, such that real data in any combination could not produce such waveforms.

Properly encoded data can never produce transitions which are further apart than one data bit period, and so synchronising is achieved by deliberately extending this period. Fig. 4 shows that the first four bit periods of each subframe are used for synchronising, and that the patterns which are used during these periods deliberately violate the coding rules. There are two main patterns, A-sync and B-sync, which denote the beginning of the subframes of the two channels, and are different so that the channels can never be reversed. There is then a further pattern which occurs once every 192 blocks, and replaces the A-sync pattern only. This pattern denotes the beginning of the channel status information sequence which will be described in due course.

The remaining 28 bits in the subframe will now be described. Bits 4-7 are auxiliary bits, which might, for example, be used for voice grade communication over the same channel as the main audio. Bits eight through 27 are the main audio sample data, and a 20 bit sample can be accommodated. The most significant bit is always bit 27, which is transmitted last. If wordlengths shorter than 20 bits are sent, the MSB must still be bit 27, and lower order bits are sent as zeros. This means that when units of different wordlength communicate, there is no difference in signal level because the MSB always has the same significance. Only the resolution increases with longer wordlengths. The reason for sending the most significant bit last is that serial arithmetic will generate carries which must be added in to the next higher bit. This is facilitated by sending the bits in ascending order of significance.

The V-bit, bit 28 is currently rather vaguely defined, but it has been proposed that the meaning of this bit should be that it indicates that the accompanying sample is suitable for conversion to analogue audio.

The U-bit is a user defined bit which can build up into a serial message over a number of blocks. The user is free to use this bit in any way, but will only be meaningful if both sending and receiving devices adhere to the same protocol.

The C-bit is the channel status bit which builds up into a 192-bit or 24 byte sequence over successive frames. The third synchronising pattern mentioned above identifies the beginning of the sequence.

Fig. 5 shows the contents of the channel status message, and individual bytes of the message are expanded in Figs. 6, 7, 8 and 9.

The 192 block message repeats every four milliseconds if a sampling rate of 48kHz is used, and pro-rata at other sampling rates.

The P-bit, or parity bit has two functions. First it allows a simple check to be made on the integrity of the incoming subframe by counting the number of data ones to see if it is odd or even. The parity bit is generated on transmission to make the total number of ones even, so odd received polarity indicates an error. The second function of the bit is that the even number of ones causes the polarity at the end of the subframe to be the same as at the beginning, so that sync patterns are always received with the same polarity.

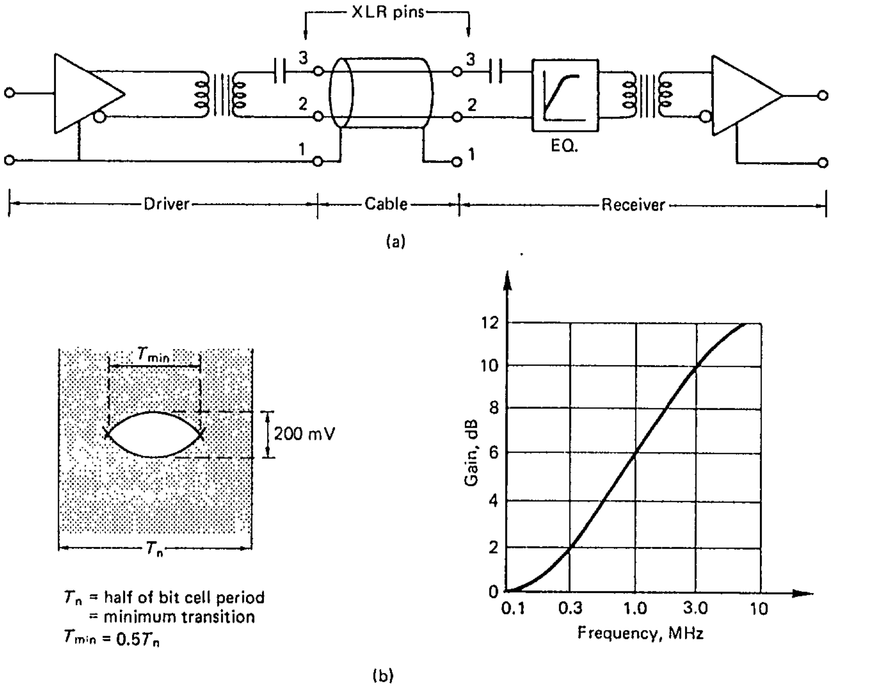

Figure 1 The configuration of the AES/EBU interface which is suggested for long cables. XLR pin connections are the same as balanced-line audio cables, which can be used. At (b) the minimum eye pattern which a receiver must be able to detect, and a suggested equalization response.

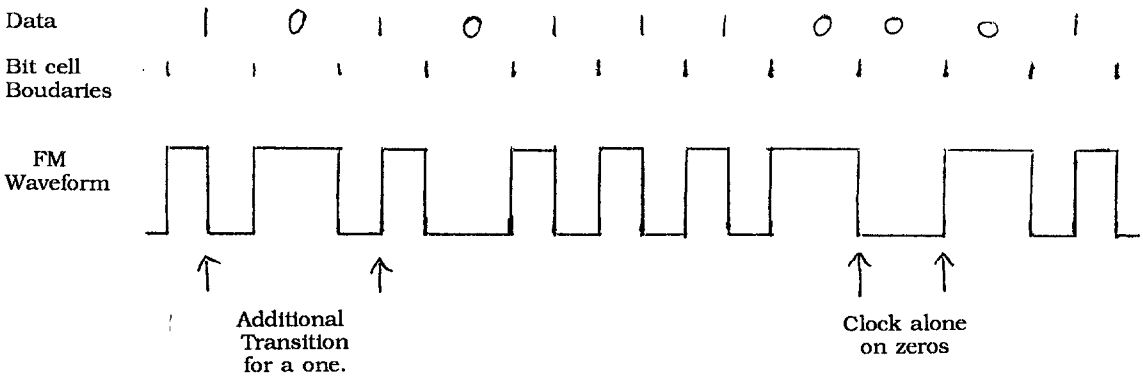

Figure 2 The encoding rules for FM channel code.

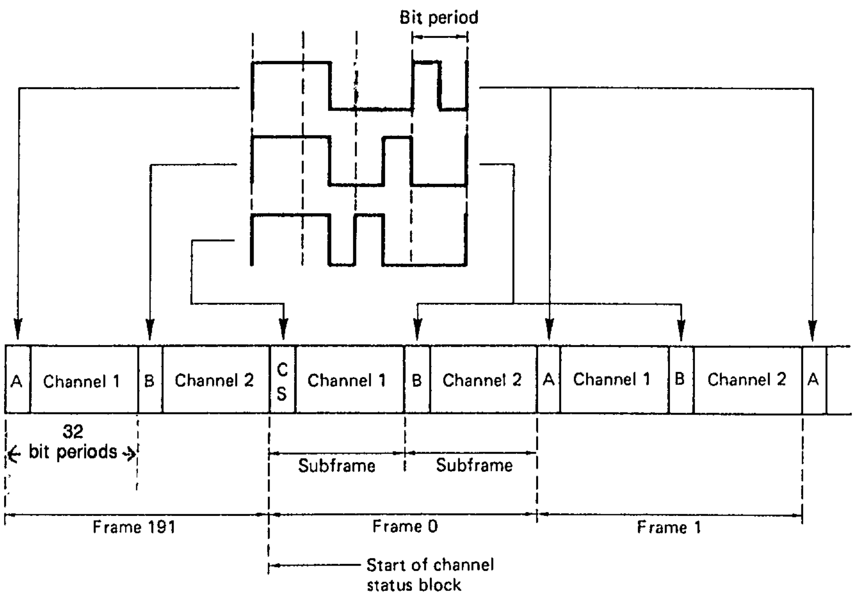

Figure 3 The A and B channels are distinguished by different sync patterns. Once every 192 blocks, the A sync is replaced by channel status block sync. Inverse of sync patterns are also valid; system is not polarity-conscious.

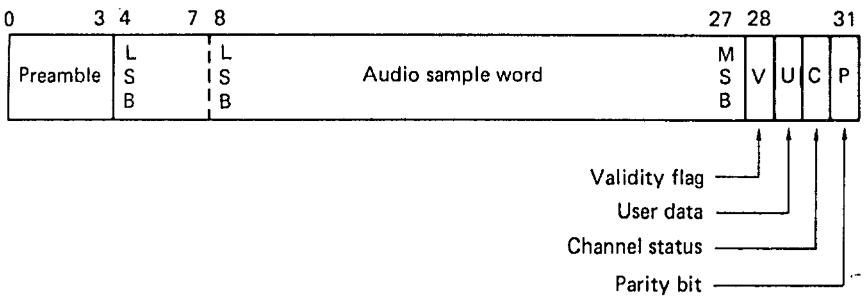

Figure 4 The basic subframe structure of the AES/EBU format. Sample can be twenty bits with four auxiliary bits, or 24 bits. LSB is transmitted first.

Byte	Content
0	Emphasis + sampling rate (see Figure 5.8)
1	Channel usage (see Figure 5.9)
2	Wordlength (see Figure 5.10)
3	Vectored target byte from byte 1
4–5	Reserved
6–9	Alphanumeric channel origin data = 4 x 7 bits ASCII + odd parity
10–13	Alphanumeric channel destination data = 4 x 7 bits ASCII + odd parity
14–17	Local sample address code: 32 bits binary address of first sample in this block
18–21	Timecode = 32 bits binary timecode of first sample in block
22	Data reliability flags (see Figure 5.11)
23	CRC $x^8 + x^4 + x^3 + 1$ on bytes 0-23

Figure 5 The content of the 24-byte sequence of channel-status data in the AES/EBU format.

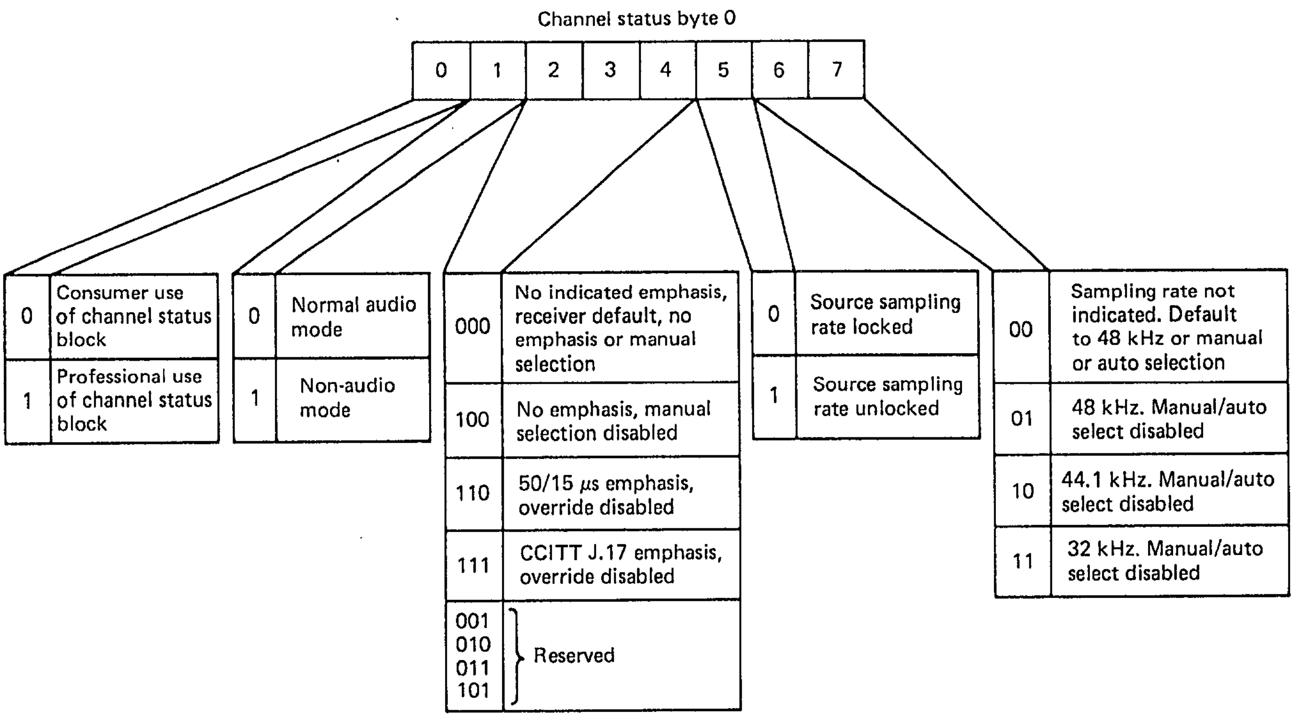

Figure 6 The first byte of the channel status information in the AES/EBU standard deals primarily with emphasis and sampling rate control.

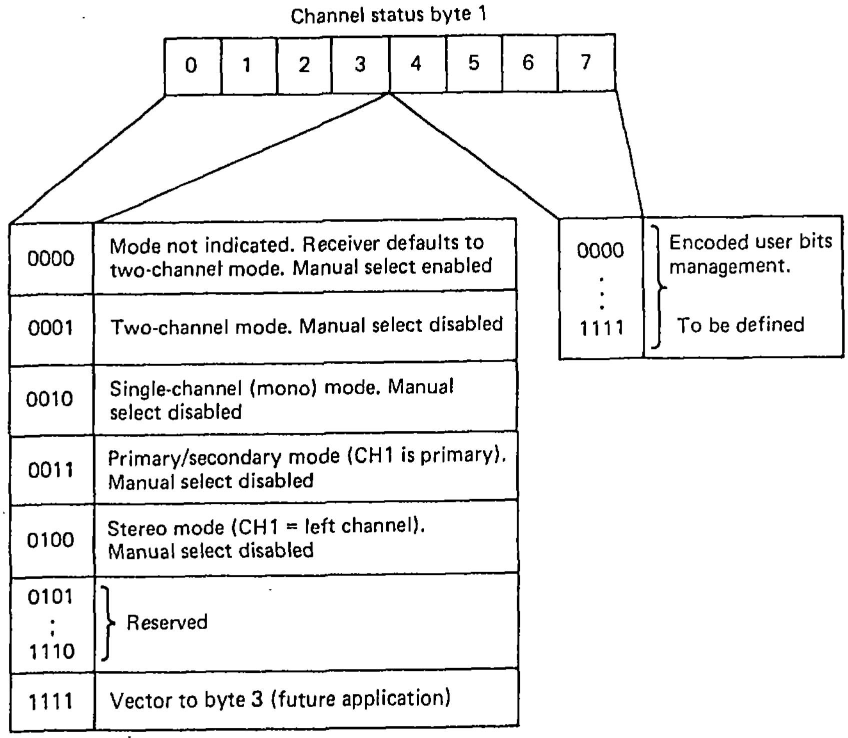

Figure 7 The secondary byte of the channel-status information currently deals with audio channel usage, but will be extended in the future to manage user bits.

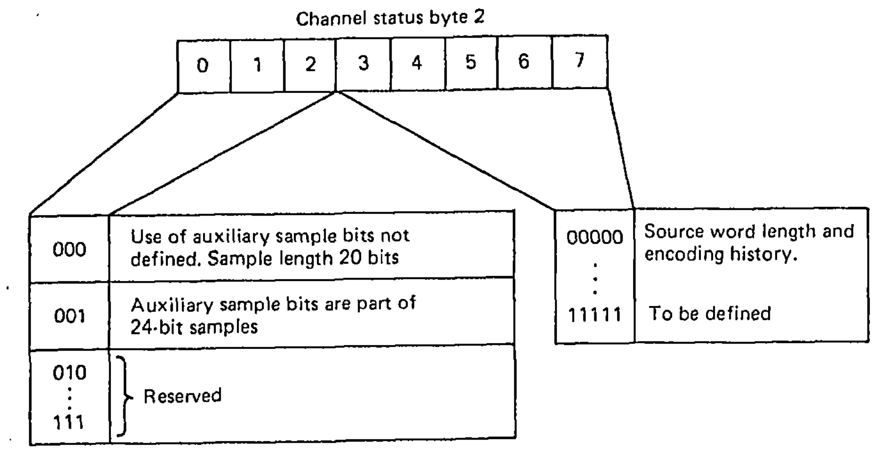

Figure 8 Byte 2 of channel status determines whether all 24 bits of the word slot are used for audio samples, or just 20 maximum.

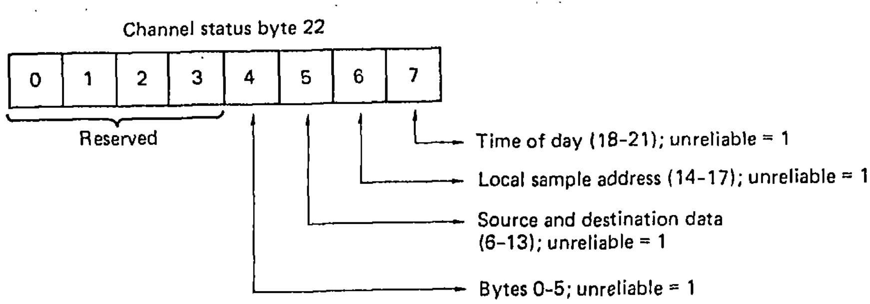

Figure 9 Byte 22 of channel status indicates if some of the information in the block is unreliable.