SONY®

Introduction to DIGITAL AUDIO

by Luc Theunissen

TRAINING TEXT

SONY®

Introduction to DIGITAL AUDIO

by Luc Theunissen

TRAINING TEXT

CONTENTS

INTRODUCTION

The possible application of digital signal processing in the audio field has been known and studied for many years: the first theoretical description of pulse-code modulation in telephony actually was written more than 40 years ago!

For application of PCM in High-Fi and professional audio however, even today we are not so far from the limits of what is technologically possible. Therefore, only relatively recently, it has become possible to develop commercially viable digital audio products.
As a consequence, only a small group of high-level specialists are nowadays familiar with the theory and applications of digital audio technology.

This booklet is intended as a way to familiarize a newcomer in the field with some of the theoretical and technological aspects of PCM-audio, in the simplest possible way. We hope it will encourage towards further studies in this interesting field.

SONY SERVICE CENTRE (Europe) N.V.

PART I

Analog - to - Digital Conversion

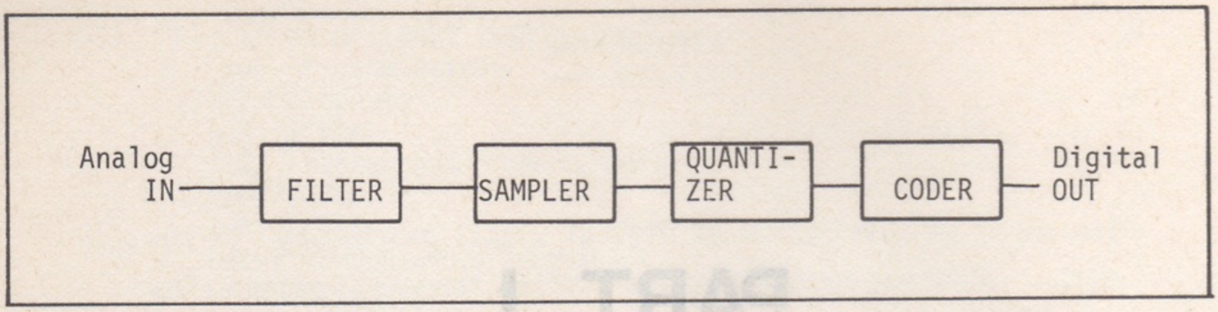


Fig. 1-1. Basic steps of analog-to-digital conversion

PART I

ANALOG-TO-DIGITAL CONVERSION

I. OUTLINE

An analog signal, be it audio or any other, has to be converted to a digital signal before it can be digitally processed. Although the principles of A/D- and D/A-conversion may seem relatively simple, in fact, this conversion between the analog and the digital domain is, technologically speaking very difficult and will unavoidably cause degradation of the original signal. Consequently, most of the time this stage will be the limiting factor that determines system performance.

Conversion from the analog to the digital domain is done in several steps, which however are not always clearly defined :

- filtering
- sampling
- quantization
- coding.

(see Fig. 1-1.)

(a)

(b)

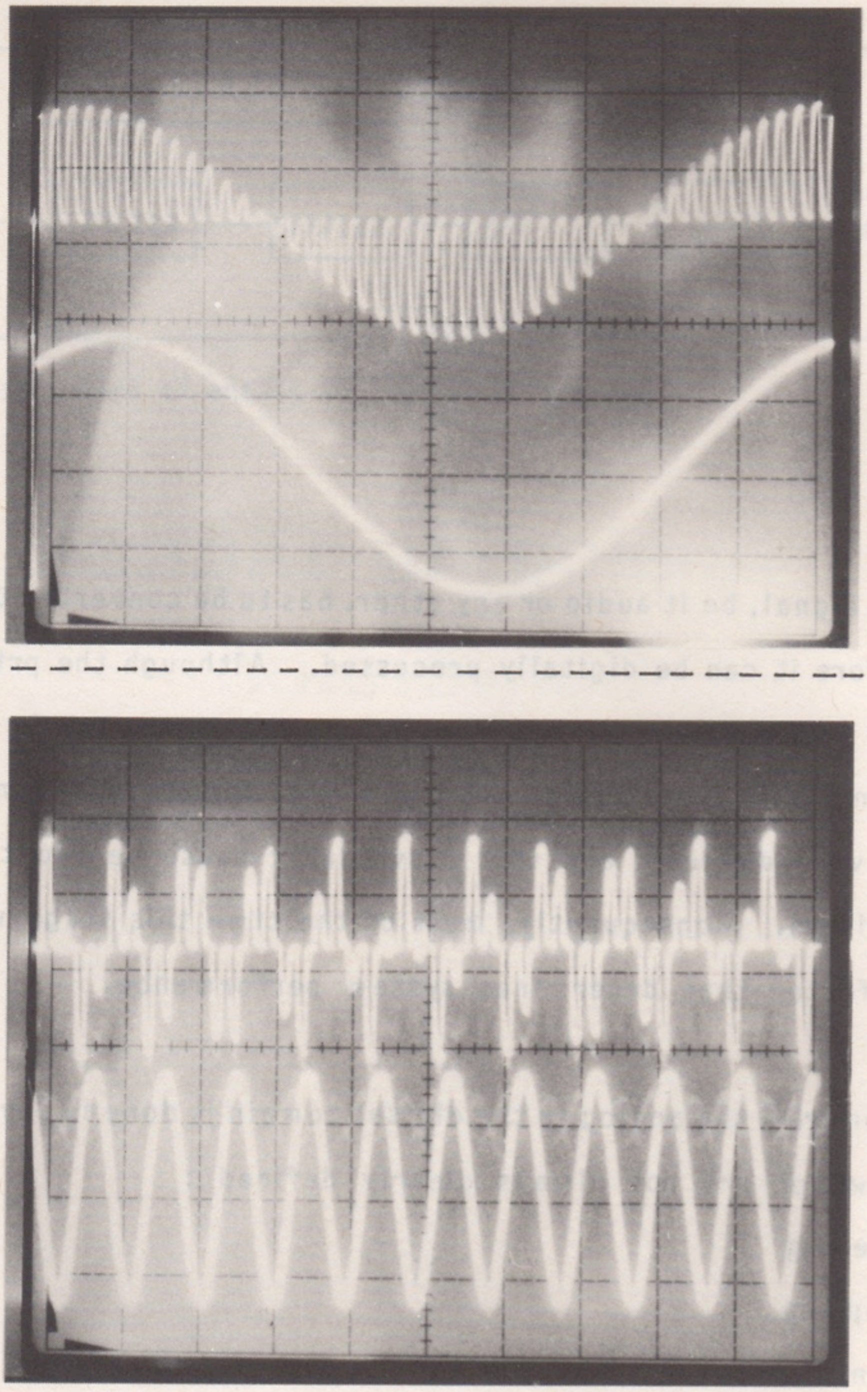


Fig. 1-2. Two examples of sine waves together with their sampled versions (f_S = 44.056kHz)

(a)1kHz sine wave

(b)10kHz sine wave

Although sampling in (b) seems much coarser than in (a), in both cases restitution of the original signal is perfectly possible.

Below, we will explain and discuss these basic steps of analog-to-digital conversion, along with the signal degradations that each conversion block is likely to introduce.

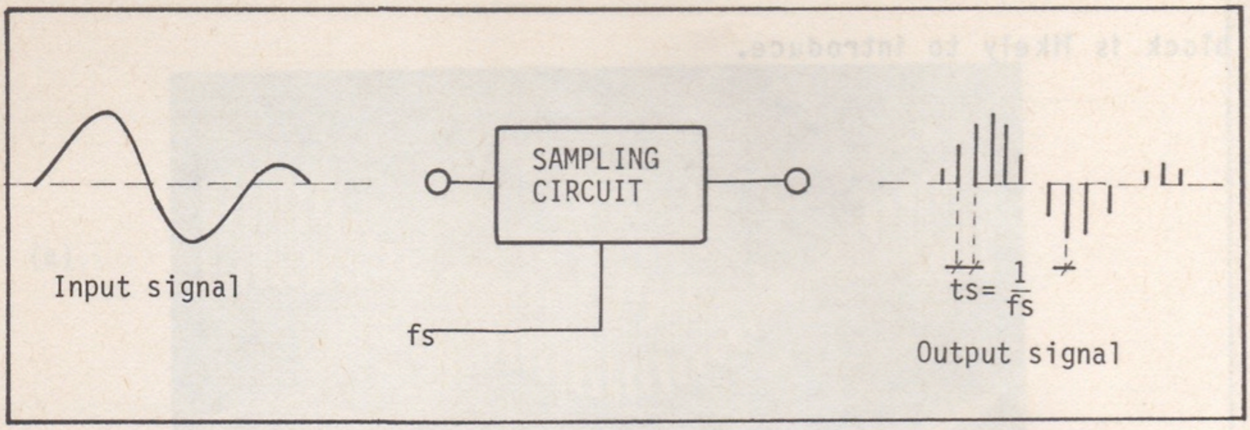


Fig. 1-3. Principle of sampling

II. SAMPLING

1. Principle - The Nyquist Theorem

By definition, an audio signal varies continuously in time. To enable it to be converted into a digital signal, it is necessary that the signal is first sampled, i.e. at certain points in time a sample must be taken of the input value, which subsequently can be converted. The fixed time intervals between each sample are called sampling intervals (t_s).

Although the sampling operation may seem to introduce a rather drastic modification of the input signal, (since it ignores all the signal changes that occur between the sampling times), it can be shown that the sampling process itself in principle removes no information whatsoever, as long as the sampling frequency is at least twice the highest frequency that is present in the input signal.

This is the famous Nyquist criterion on sampling (also called Shannon criterion or Shannon theorem).

The correctness of the Nyquist criterion can be understood when we consider the spectra of the input and output signals (Fig. 1-4.).

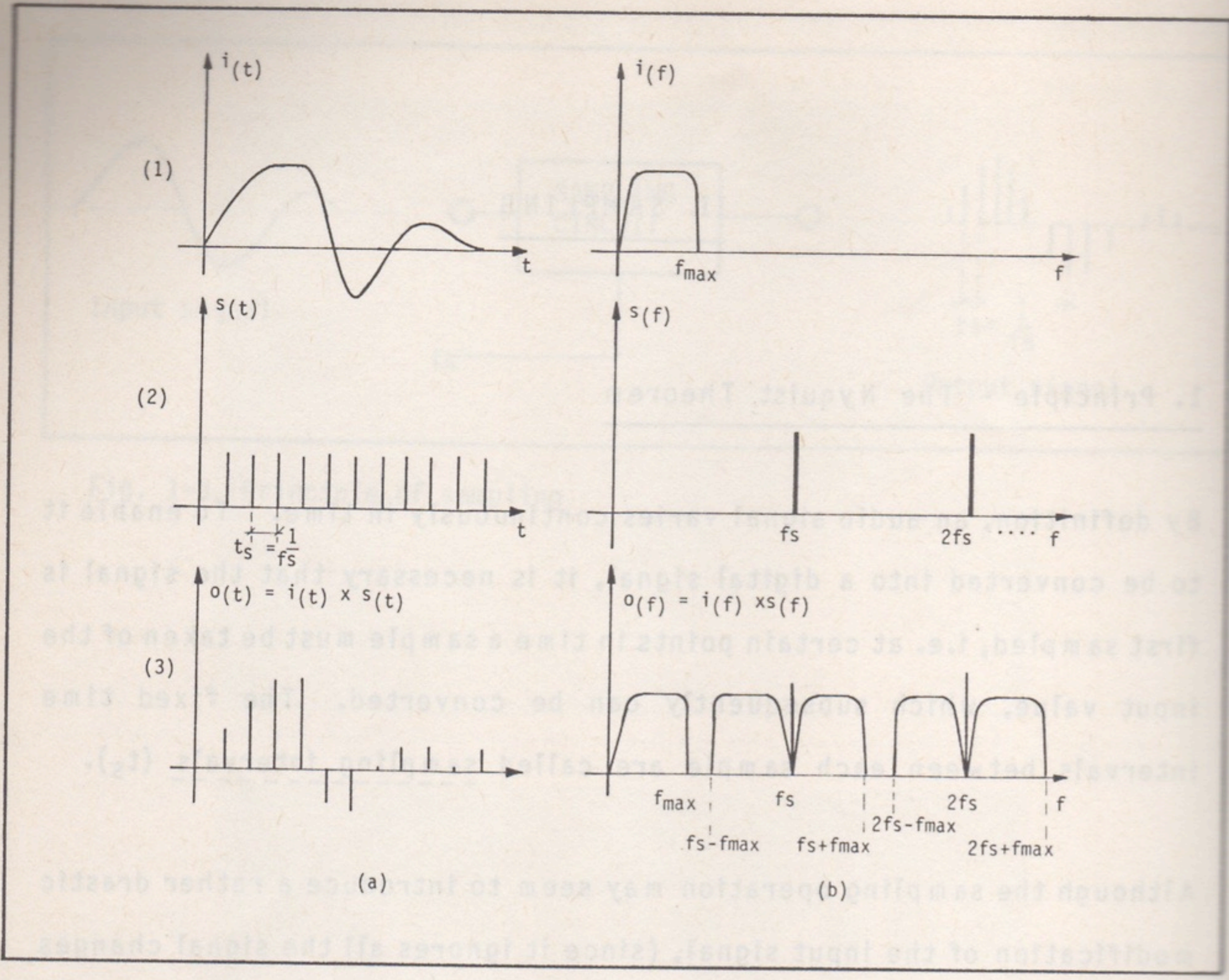


Fig. 1-4. Time domain - (a), and frequency domain in representation (b) of sampling process

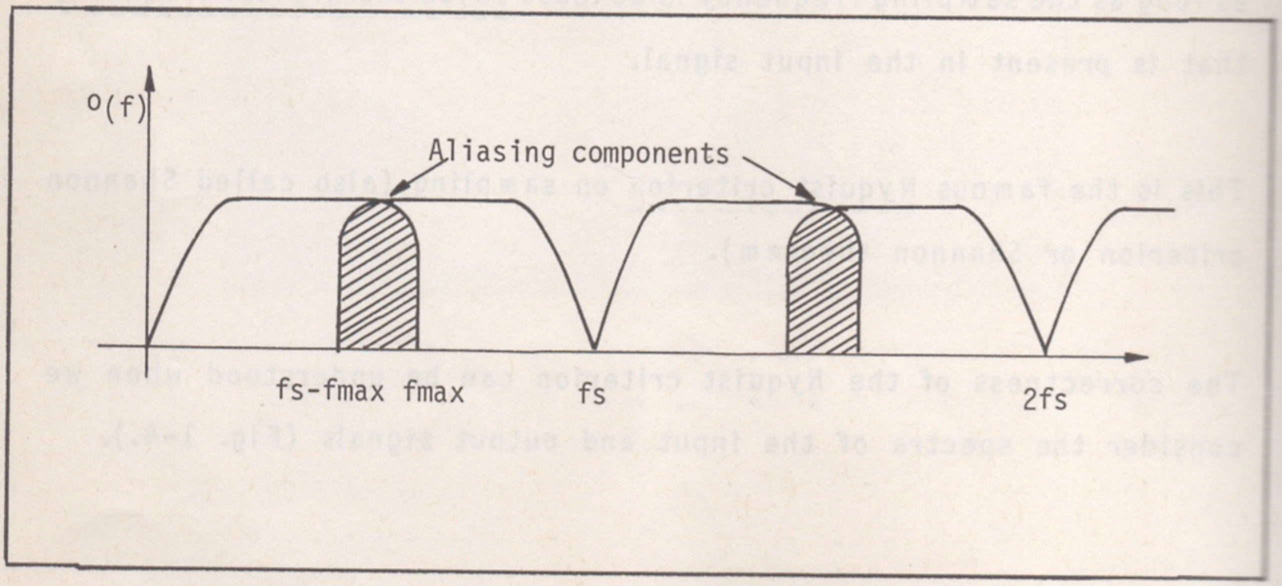


Fig. 1-5. Principle of aliasing

If we consider an input signal $i_{(t)}$ which has a maximum frequency f_{max}, its spectrum may have any form between 0 Hz and f_{max} (1); the sampling signal $s_{(t)}$, having a fixed frequency f_s, can be represented by one single line at f_s (2). The sampling process is equivalent to a multiplication of $i_{(t)}$ and $s_{(t)}$, and the spectrum of the output signal, as shown in (3), can be seen to contain the same spectrum as the input signal, together with repetition of the same spectrum, modulated around multiples of the sampling frequency.

As a consequence, proper low-pass filtering can completely isolate and thus completely recover the input signal.

From Fig. 1-4., it can also easily be understood that f_s must be greater than $2 \times f_{max}$. If this would not be the case, the original spectrum would overlap with the modulated part of the spectrum, and consequently be inseparable from it (Fig. 1-5.)

For example, a 20kHz-signal sampled at 35kHz would produce a 5kHz difference frequency. This phenomenon is known as <u>aliasing</u> and must be avoided by all means.

For this reason, a very sharp cut-off filter (anti-aliasing filter) is absolutely required in the signal path to remove all the unwanted harmonics out of the input signal <u>before</u> the sampling takes place. If this is not done, aliasing will cause unremovable distortion components.

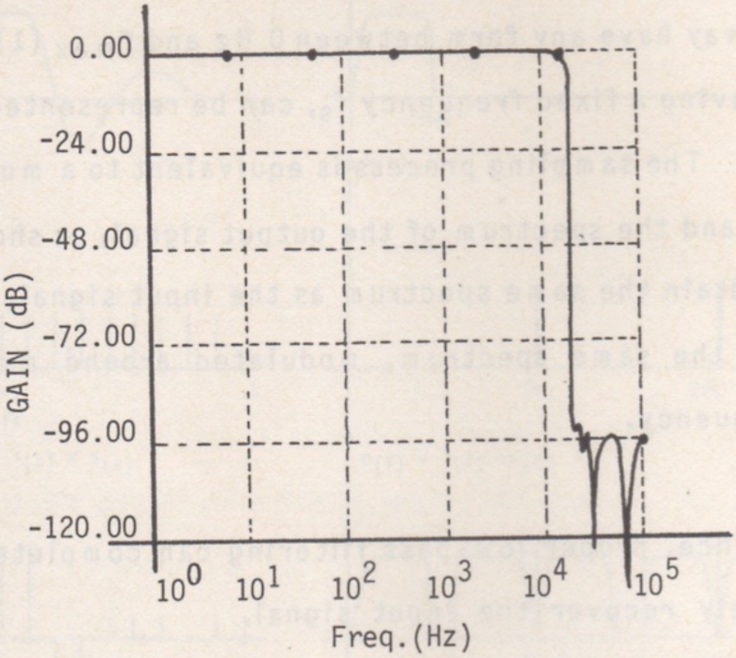


Fig. 1-6. Characteristic of anti-aliasing filter

2. The problem of the sampling frequency

2-1. Minimum value

In the digital audio field, proper selection of a convenient sampling frequency is extremely important.

It should indeed be avoided to select the sampling rate unnecessarily high, since this tends to increase the hardware costs dramatically.

On the other hand, since ideal low-pass filters do not exist, a certain safety margin must be incorporated in order to avoid any frequency higher than $0.5f_s$ from passing through the filter with insufficient attenuation.

Therefore, if we wish to reproduce a flat audio bandwidth of 20 Hz - 20.000 Hz, a sampling frequency of 44 kHz can be considered as a minimum, giving 22 kHz as extreme frequency where the "anti-aliasing filter" already gives sufficient attenuation (60 dB) to make eventual aliasing components inaudible (Fig. 1-6).

2-2. Use of 44.056/44.1 kHz

Another important criterion for selection of a sampling frequency lies in the fact that, to arrange the digital information in a video-like signal, (as it is done in all the PCM-adapters which use a standard helical-scan video recorder as a storage medium) there must be a fixed relationship between sampling frequency (f_s), the horizontal video frequency (f_h) and the vertical frequency (f_v). For this reason, these frequencies must be derived from the same master clock by frequency division, or in other words, f_s and f_h should have an as-low-as-possible Least Common Multiple.

In the NTSC system, f_h = 15.734,2657... Hz (a non-integer value, due to the necessary relationship with the NTSC chroma and audio subcarriers), whereas in the European PAL or SECAM systems, f_h = 15.625,0 Hz.

Calculations have shown that, for the NTSC system, a frequency of 44.055944... Hz would come closest to this ideal, whereas for the PAL system, a frequency of 44.100 Hz was also quite feasible.

The difference between these two frequencies is only 0,1%, which is negligible for normal use (the difference translates as a pitch difference at playback, and 0,1% is entirely imperceptible).

As a consequence, 44.056 has been adapted as sampling frequency in the EIAJ-standard for PCM-adapters for EIAJ, while 44.1 will be used by adapters for the CCIR-system, as well as for the future Compact Digital Audio Disc.

2-3. Use of 50.4 kHz

For studio recording, mostly using the multiple-track recording technique, the helical-scan recorders are not so ideal, therefore stationary-head recorders are being designed, giving up to 48-channel recording capability.

One of the aspects of such studio recorders is that the tape speed must be adjustable, in order to allow easy synchronization between several machines, correct tuning, etc. If we consider a speed tuning range of 10%, a sampling frequency of 44 kHz could become less than 40 kHz, which is insufficient to comply with the Nyquist criterion. Therefore, such machines should use a higher sampling rate, for example 50 kHz, which at the lowest speed would still give a satisfactory 45 kHz.

However, to allow direct (digital) dubbing between such stationary head recorders and helical-scan recorders (which with their two-track capability are ideal and reliable master recorders), conversion between the higher and the lower (44.1 kHz) sampling rate must be possible. To do this in an economical way, a simple mathematical relationship between both frequencies is important. A frequency that has an integer relationship (8/7) with 44.1 kHz is 50.4 kHz. Also, 50.4 kHz has a good relationship with various TV and film standards (25 frames/sec), so it is also possible to make digital data blocks coincide with frames. For these reasons, Sony as well as some other Japanese manufacturers selected 50.4 kHz for their stationary-head machines.

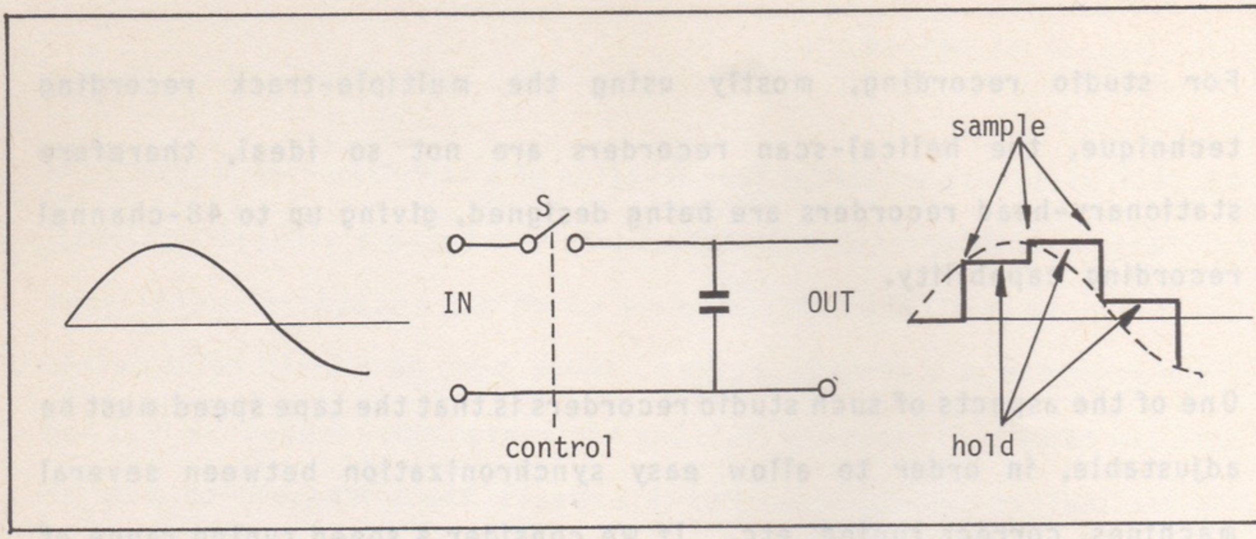


Fig. 1-7. Basic sample-hold circuit

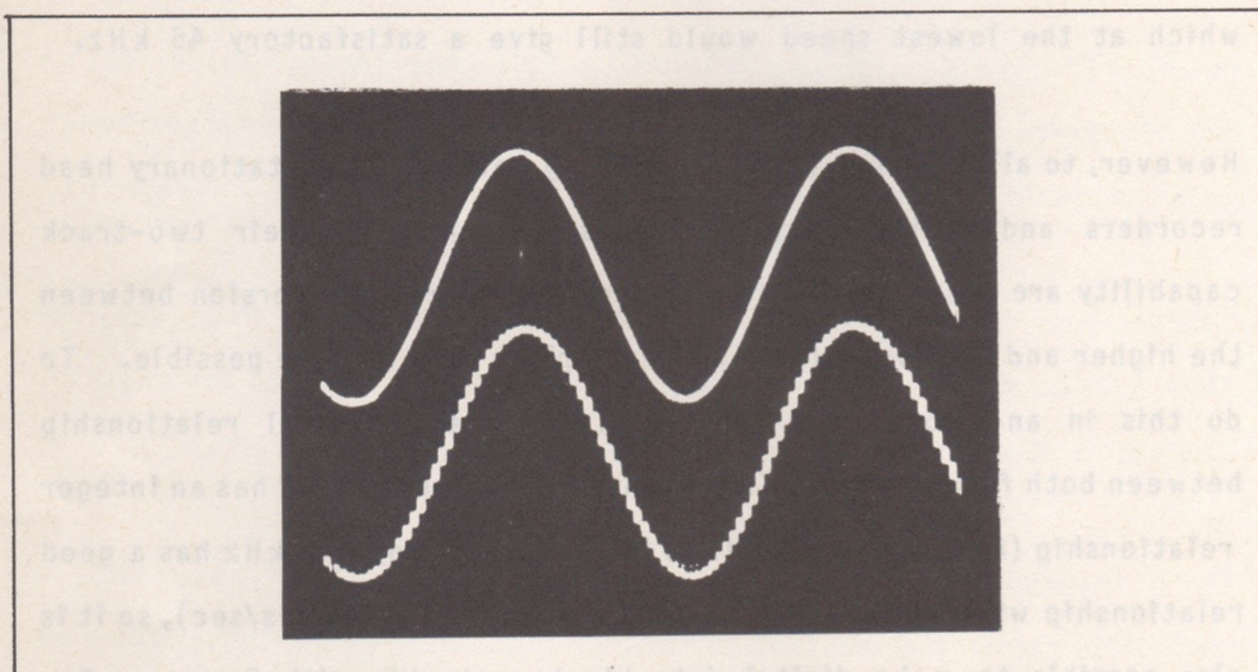


Fig. 1-8. Sine Wave before (upper) and after (lower) Sample-Hold

2-4. Use of 32 kHz

Quite a few European broadcasters are already utilizing digital transmission links between studios and transmitters. However, since the FM-broadcast frequency range is limited to 15 kHz anyhow, the EBU (European Broadcasting Union) has chosen 32 kHz as an (economical) standard sampling frequency. The same frequency will also be adopted by the Japanese P.T.T. However, this complicates the conversion problems again.

3. Sample-Hold Circuits

In the practice of analog-to-digital conversion, the sampling operation is performed by so-called Sample-Hold Circuits, that, once the sample has been taken, store the sampled analog voltage a certain time, during which the voltage can be converted by the A/D convertor into a digital code (see further).

The principle of a Sample-Hold Circuit is relatively simple (Fig. 1-7.).

A basic Sample-Hold Circuit is in fact a 'voltage memory' device that stores a given voltage in a high-quality capacitor. To sample the input voltage, switch S closes momentarily; when S opens, capacitor C holds the voltage until C closes again and passes the next sample. Fig. 1-8. shows a sine wave at the input and the output of a S/H circuit.

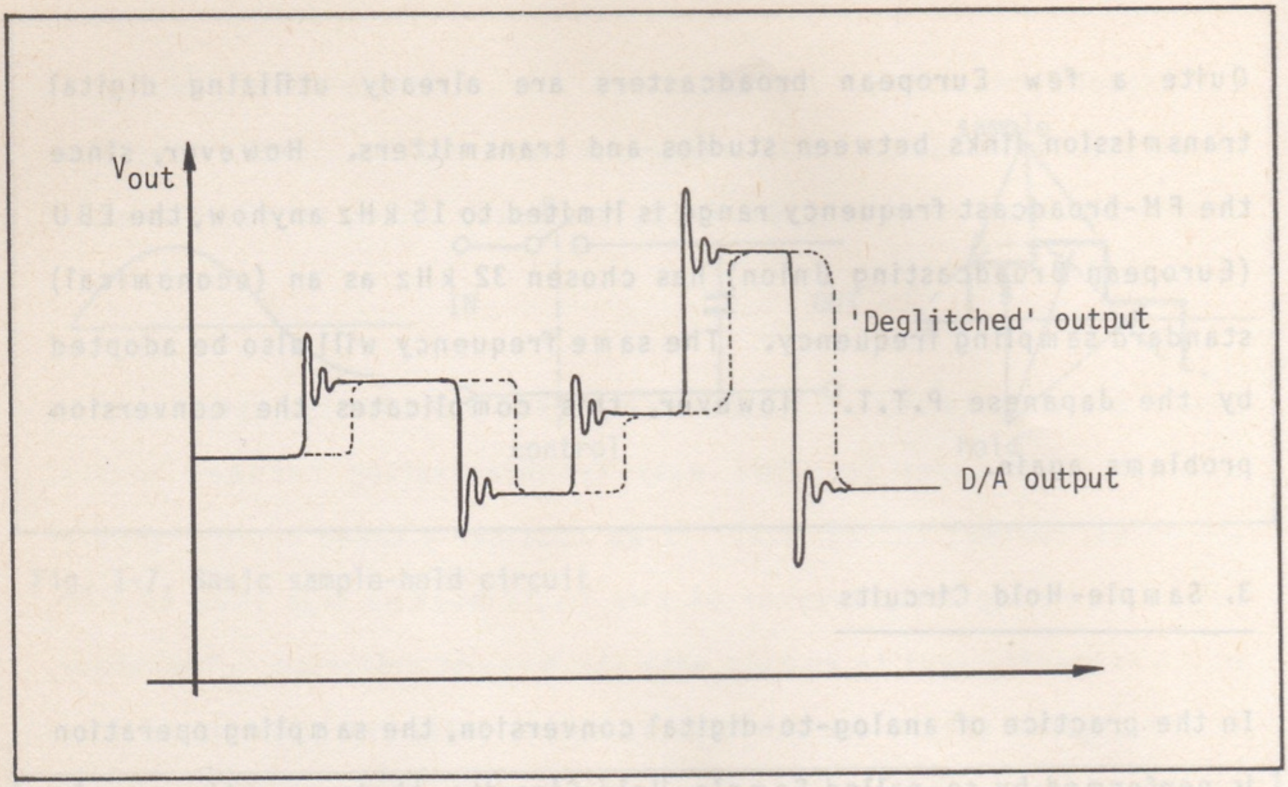


Fig. 1-9. S/H used as Deglitcher

Practical circuits obviously will have buffer amplifiers both at input and output, in order not to load the source, and to be able to drive a load such as an A/D convertor. The output buffer amplifier must have a very high input impedance, and very low bias current, so that the charge of the hold capacitor does not leak away too rapidly. Also the switch must be very fast and have very low off-stage leakage.

Sample-Hold Circuits are not only used in A/D conversion to sample an analog input signal, but also in the reverse stage, after the D/A conversion, to remove transients (glitches) from the output of the D/A convertor and thus facilitate the task of the low-pass filter. In this case, it is often called 'De- glitcher'. (Fig. 1-9.)

4. Aperture Control

The output signal of a sampling process is in fact a Pulse- Amplitude Modulated signal (PAM). It can be shown that, for sinusoidal input signals, the frequency characteristic of the sampled output is :

$$H\ (w_v) = \frac{t_o}{t_s} \cdot \frac{\sin \frac{t_o}{2} w_v}{\frac{t_o}{2} \cdot w_v}$$

In which w_v = angular velocity of input signal (= $2\pi fv$)

t_o = pulse width of the sampling pulse

t_s = sampling period

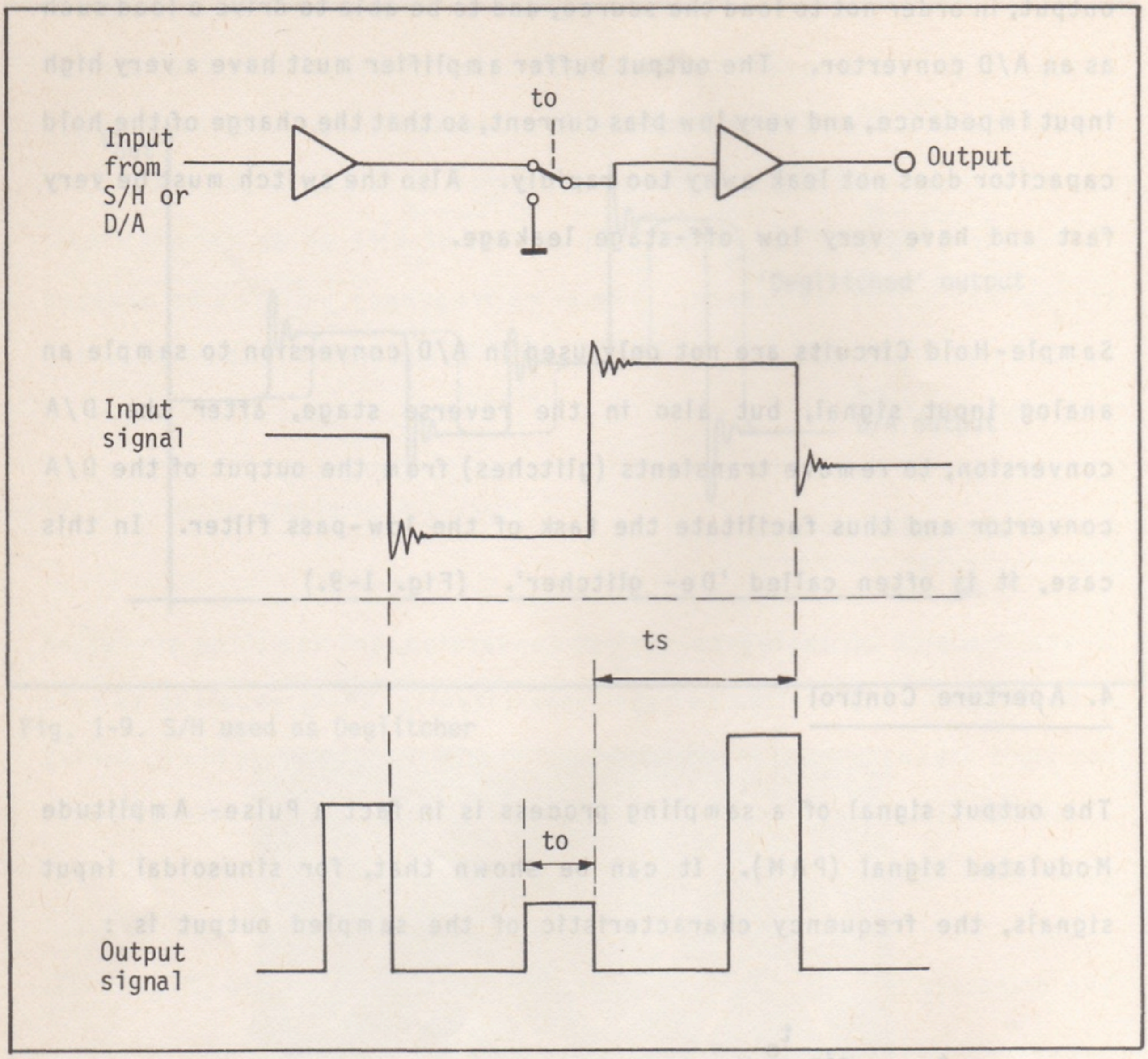


Fig. 1-10. Basic circuit and waveforms of aperture control circuit

At the output of a Sample/Hold, or a D/A convertor, however, $t_o = t_s$, consequently :

$$H(w_v)_{\,t_o = t_s} = \frac{\sin \frac{t_s}{2} w_v}{\frac{t_s}{2} \cdot w_v}$$

This means that at maximum admissible input frequency (which is half the sampling frequency), $w_v = \frac{\pi}{t_s}$, and consequently :

$$H\left(\frac{\pi}{t_s}\right)_{t_o = t_s} = \frac{\sin \frac{\pi}{2}}{\frac{\pi}{2}} \approx 0.64 \; !$$

This decreased frequency response can be corrected by the so- called Aperture Circuit, which decreases t_o and restores a noral PAM signal. (See Fig. 1-10.)

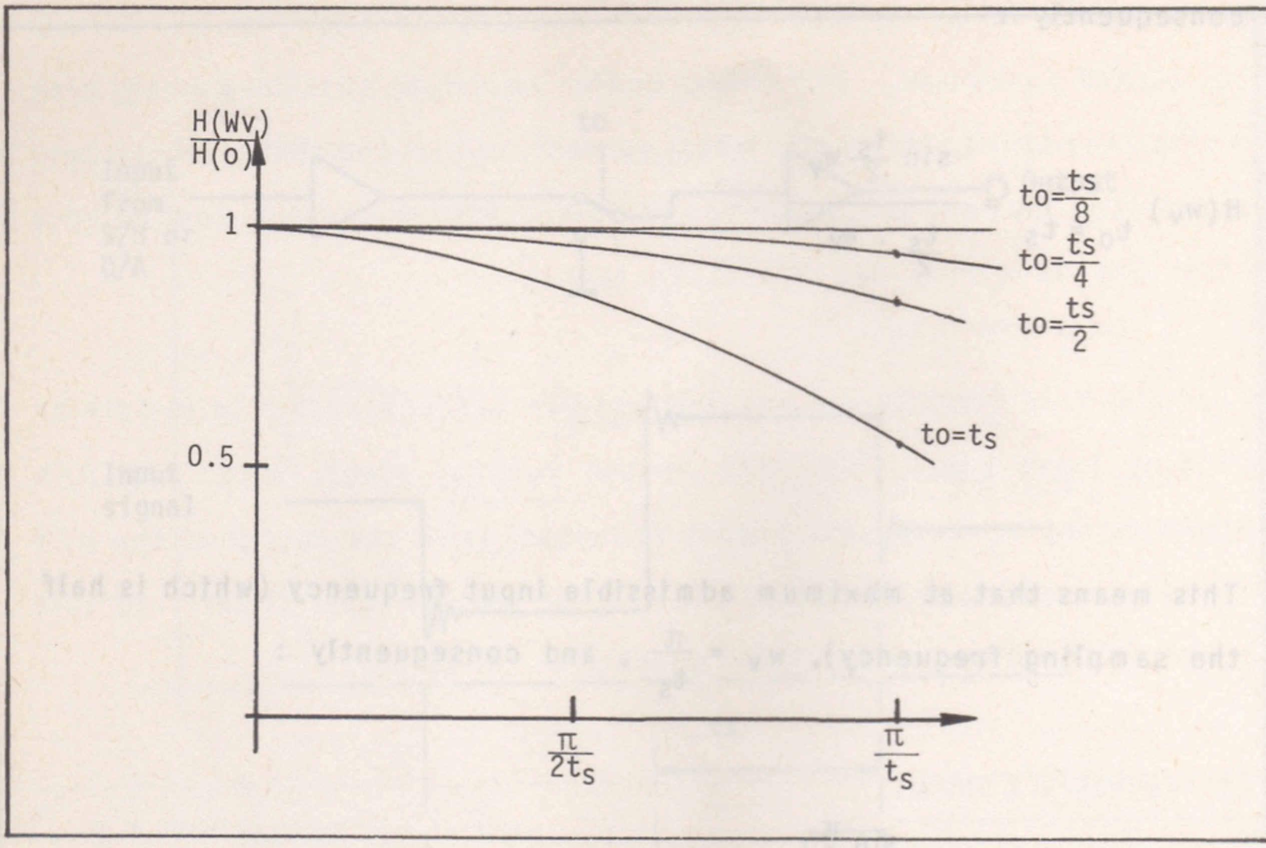


Fig. 1-11. Aperture time vs. frequency response

In practical circuits for instance, mostly $t_0 = \frac{t_s}{4}$, which leads to :

$$H\left(\frac{\pi}{t_s}\right)_{t_0 = \frac{t_s}{4}} = \frac{\sin\frac{\pi}{8}}{\frac{\pi}{8}} \approx 0.97$$

This is an acceptable value; reducing t_0 further would also reduce the average output voltage too much and thus worsen S/N- ratio.

Fig. 1-11. shows the frequency response for some values of t_0.

5. Characteristics and terminology of Sample-Hold Circuits

In a S/H circuit, the accuracy to which the 'hold' output voltage corresponds to the original input voltage obviously depends on the quality of the buffer amplifiers, on the leakage current of the holding capacitor and of the switch S (usually a FET). The unavoidable leakage causes the output voltage to decrease slightly during the 'hold'-period, which is known as 'Droop'.

In fast applications, the <u>acquisition time</u> and <u>settling time</u> are also important. The acquisition time is the time needed after the 'Hold-Sample' transition to match the input signal again within a certain error band, whereas the settling time is the time needed after the 'Sample-Hold' transition to obtain a stable output voltage. Both times obviously define the (theoretical) maximum sampling rate of the unit. The aperture time is the time interval between the beginning and the end of the Sample/Hold transition; also terms as aperture uncertainty and aperture jitter are used to indicate variations in the aperture time and consequently variations of the sample instant itself.

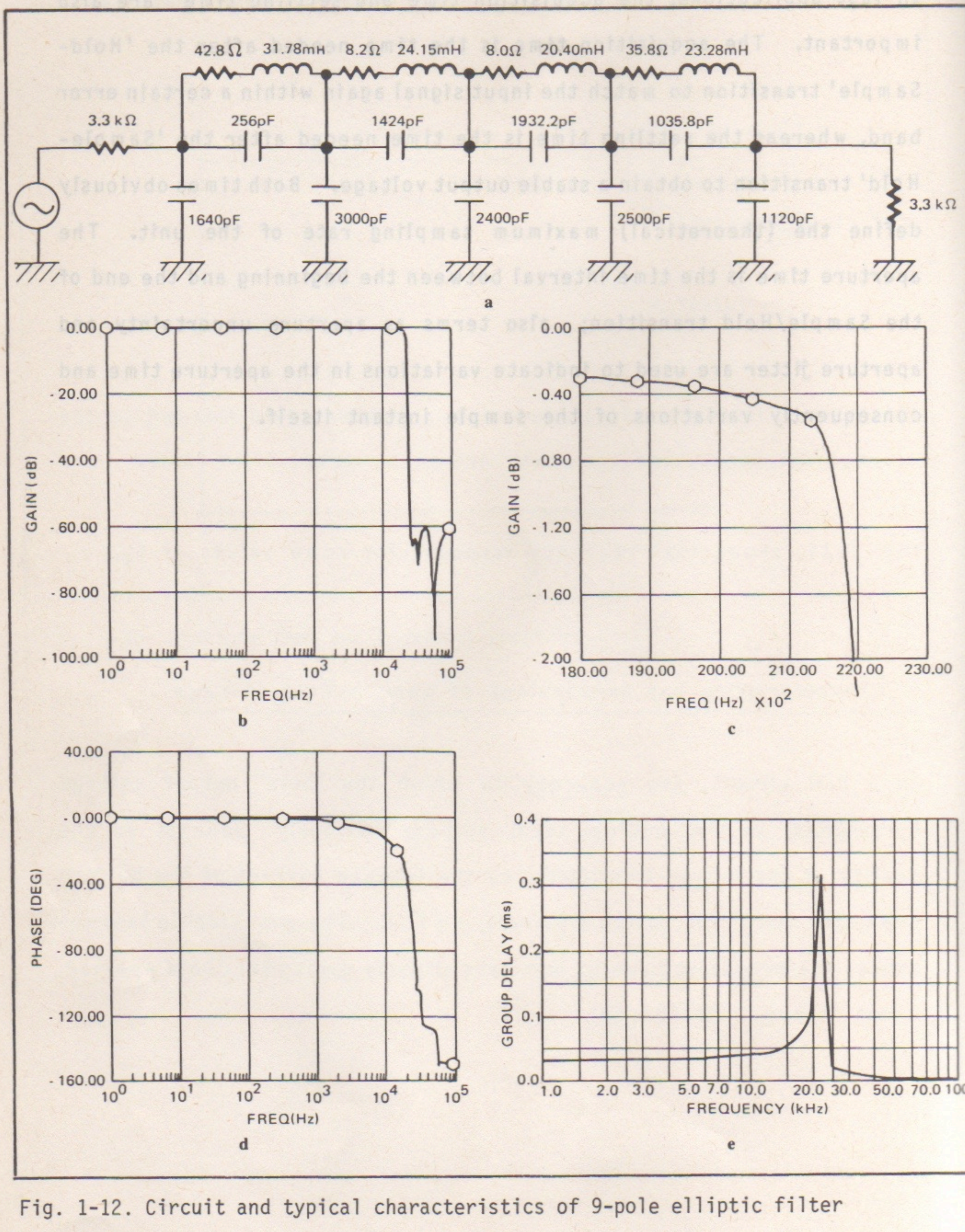


Fig. 1-12. Circuit and typical characteristics of 9-pole elliptic filter

III. FILTERING

In the foregoing chapter, we have seen that low-pass filtering is required at two places in a digital audio system :

1) In the A/D stage before the input sampler we need an anti- aliasing filter, that drastically suppresses all input frequencies higher than the Nyquist frequency.

2) We need a reconstruction filter after the Sample/Hold in the D/A stage, to separate the sampled audio signal from its images around the harmonics of the sampling frequency.

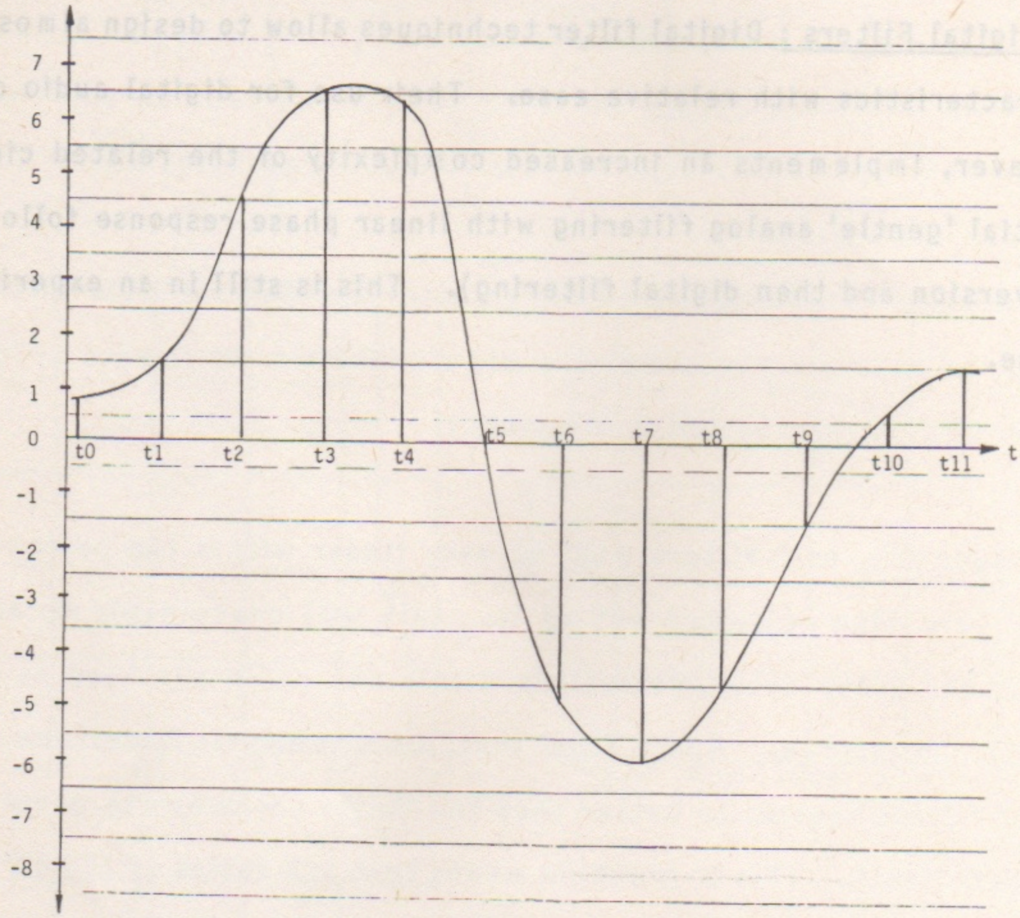
Consequently, both filters must be very linear within the passband, and have very high rejection out of it; this will unavoidably bring along phase and group delay distortions within the audio passband as well as ripple and ringing. Apart from that, Noise- and THD-performance of these filters have to be better than the limits posed by the quantization process itself. This in practice means that S/N ratios of 100 dB have to be attained.

For the practical realization of such low-pass filters, there are several approaches possible :

1) Passive Filters (LC-type) : 7th- to 9th-order filters must be used to obtain the required rejection. The coils that must be used give problems because of non-linearity, size and cost. Noise, however, is hardly a problem in this type of filters. See Fig. 1-12. for an example of a passive LPF.

2) Active Filters can also be used, but specifications are much more stringent than usually encountered in active filter designs; therefore, very careful design is necessary. Distortion can be made better than with passive filters, but noise is worse.

3) Digital Filters : Digital filter techniques allow to design almost ideal characteristics with relative ease. Their use for digital audio coding, however, implements an increased complexity of the related circuitry (initial 'gentle' analog filtering with linear phase response followed by conversion and then digital filtering). This is still in an experimental stage.



Sample	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11
Value	1	2	5	7	6	0	-5	-6	-5	-2	1	1
4-bit code (2's complement)	0001	0010	0101	0111	0110	0000	1011	1010	1011	1110	0001	0001

Fig. 1-13. Principle of quantization

IV. PRINCIPLES OF QUANTIZATION

1. Basic Principle

Even after sampling an audio signal, we are still in the analog domain : indeed, the amplitude of each sample can vary infinitely between minus and plus a certain maximum value.

The decisive step to the digital domain is now taken by quantization (see Fig. 1-13.), i.e. replacing the theoretically infinite number of samples by a finite number of values.

In a (simplified) practical system the analog signal range is divided (quantized) in a number of regions (in our example 16), and the samples of the signal are assigned a certain value (ex. -8 to +7) according to the region in which they fall. These values are then converted into digital (binary) numbers; in our example, 4 bits are needed to indicate the 16 regions ($2^4 = 16$).

The example shows a bipolar system in which the input voltage can be either positive or negative (the normal case for audio). In this case, the coding system is often the so-called 2's complement code, in which positive numbers are indicated by the natural binary code (LSB = 2^0, 1SB = 2^1, 2SB = 2^2, etc.), while negative numbers are indicated by simply complementing the positive codes (i.e. changing the state of all bits) and adding 1LSB.

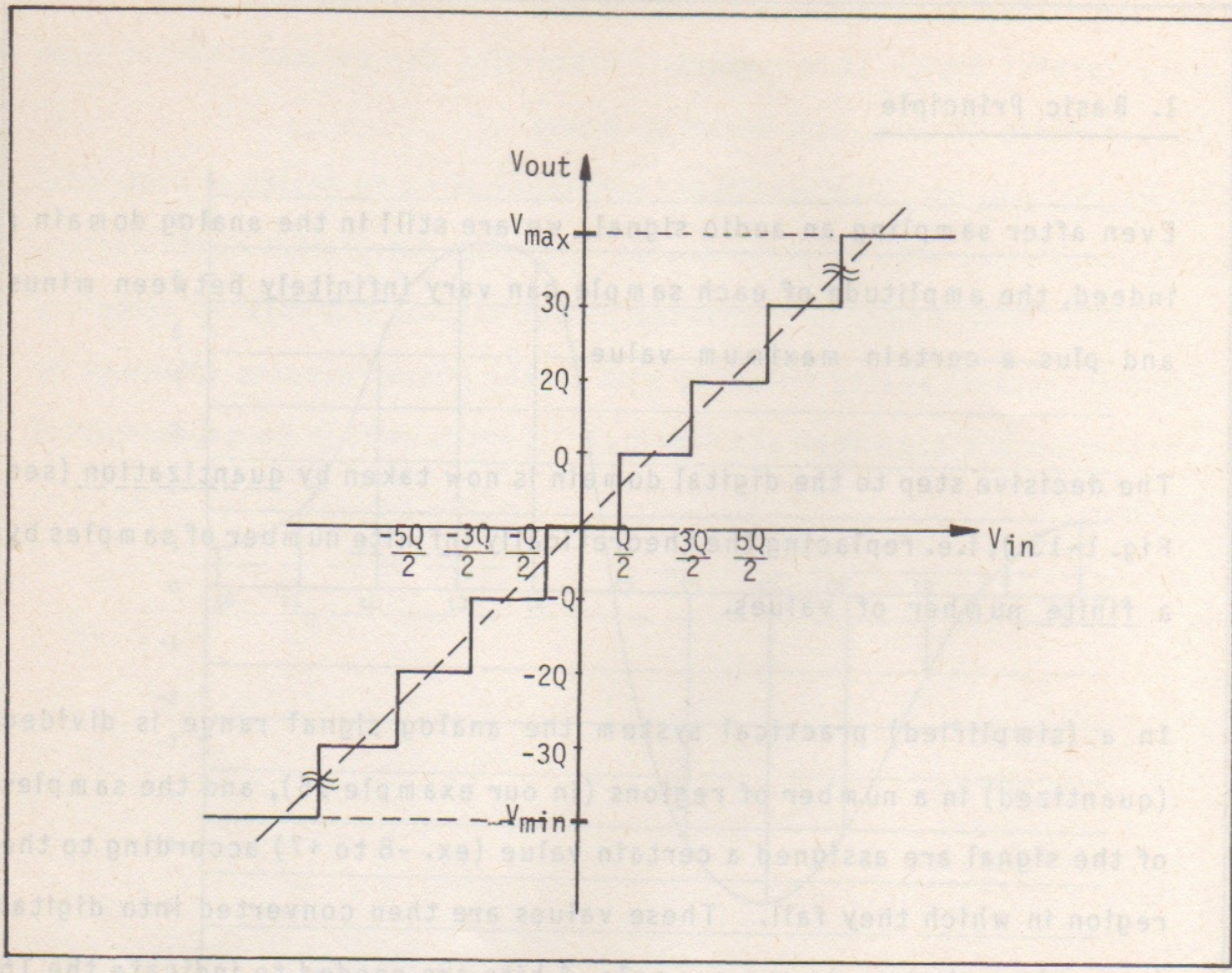


Fig. 1-14. Quantizer characteristic

In such a system, one bit (the MSB) is used as a sign bit : it is generally 'zero' for positive values but 'one' for negative values. More on the several coding systems is given in a further chapter.

The regions in which the signal range is divided are called quantization intervals, mostly represented by the letter Q. A digital word assigned to a certain quantization interval is assumed to represent the voltage at the centre of this quantization interval. Fig. 1-14. shows a typical step-wise quantizer characteristic.

In fact, quantization can be regarded as a mechanism in which some information (present in the input samples) is thrown away, keeping only as much as is required to retain a certain accuracy (or fidelity) as is needed in a certain application.

2. Quantization error

By definition, since all voltages in a certain quantization interval are represented by the voltage at the centre of this interval, the process of quantization is a non-linear process and creates an error, called quantization error (or sometimes round-off error). The maximum quantization error is obviously equal to half the quantization interval Q, except in the case that the input voltage widely exceeds the maximum quantization level (+ or - V_{max}), in which case the signal will be rounded to these values.

Generally, such overflows or underflows are avoided by careful scaling of the input signal, so in this case we can say that

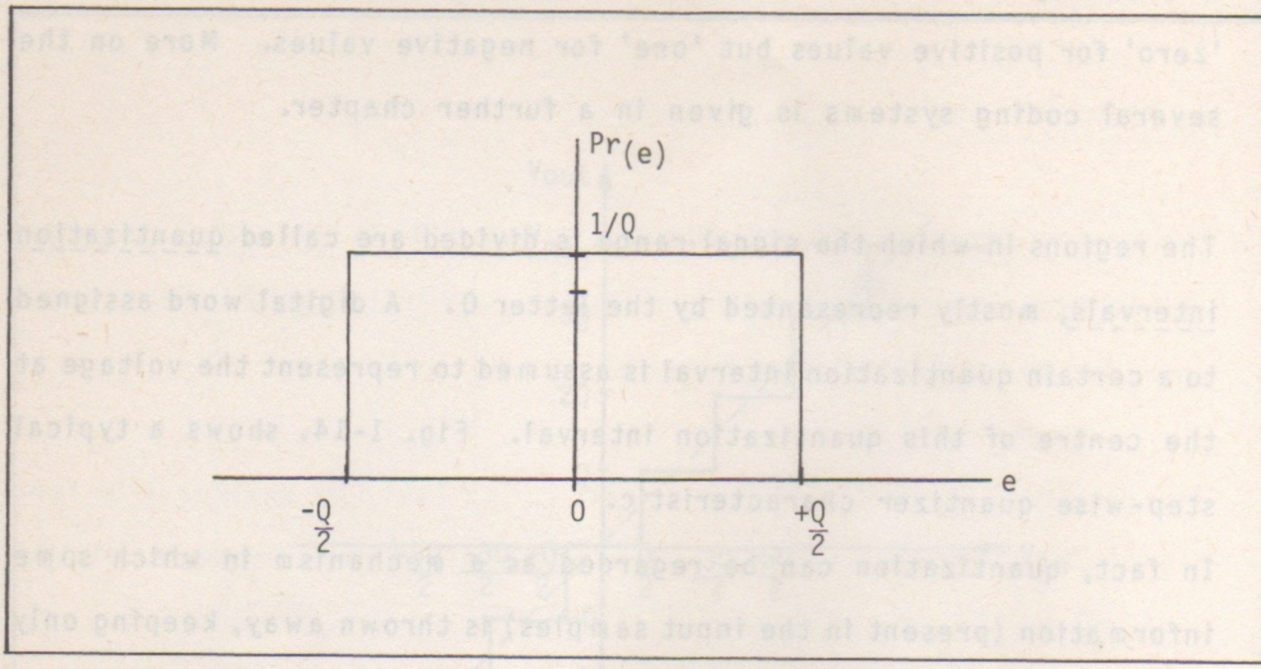


Fig. 1-15. Probability density function of quantization error

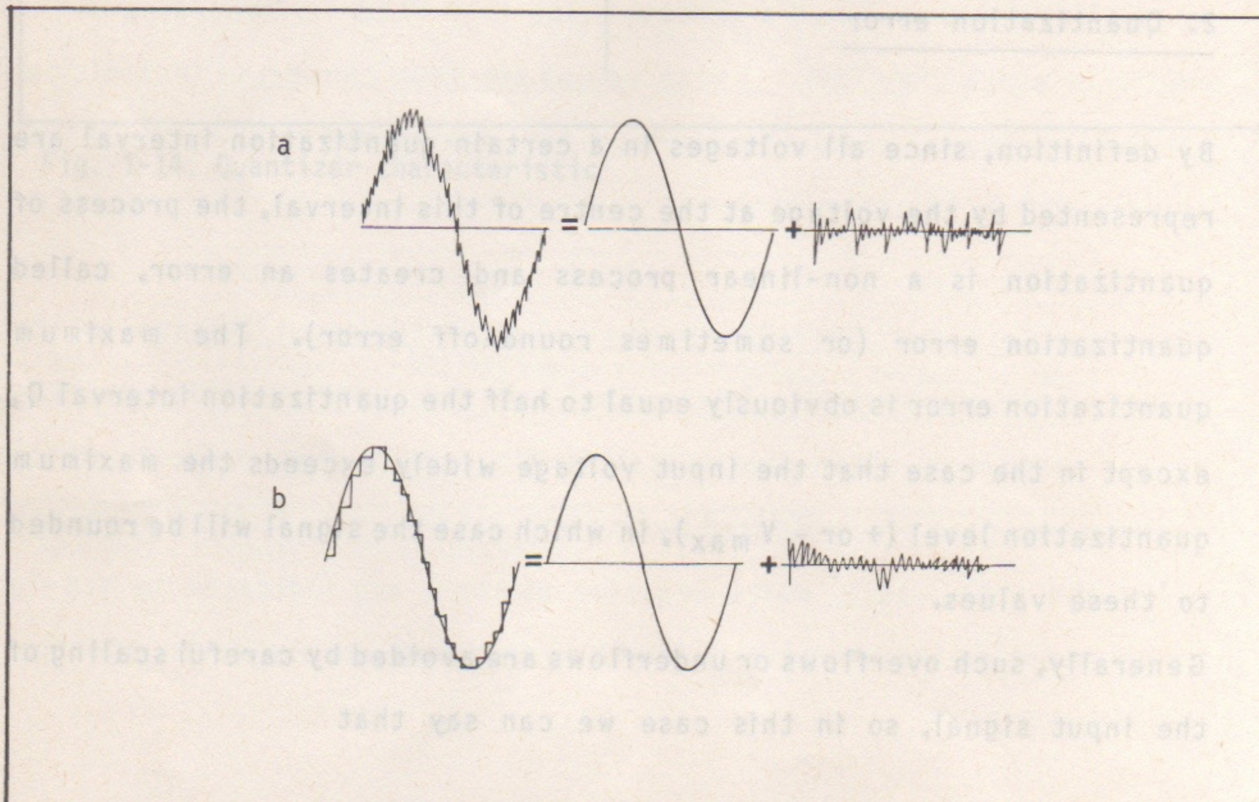


Fig. 1-16. Analogy between quantization error and noise

(a) Analog signal with noise
(b) Quantized signal with quantization error

$$- Q/2 \leqslant e_{(n)} \leqslant Q/2$$

in which $e_{(n)}$ is the quantization error for a given sample n.

It can be shown that, with most types of input signals, the quantization errors for the several samples will be randomly distributed between these two limits, or in other words, its probability density function is flat (Fig. 1-15).

There is a very good analogy between the quantization error in digital systems and 'analog' noise in analog systems : one can indeed consider the quantized signal as the infinite precision signal plus the quantization error (just like an analog signal can be considered to be the sum of the signal without noise plus a noise signal) (see Fig. 1-16). In this manner, the quantization error is often called quantization noise, and, just like in analog systems, a "signal-to-quantization noise" ratio can be calculated.

3. Calculation of the theoretical signal-to-noise ratio of a quantizer.

In an N-bit system, the number of quantization intervals N can be expressed as

$$N = 2^n \qquad (1)$$

If the maximum amplitude of the signal is V, the quantization interval Q can be expressed as

$$Q = \frac{V}{N-1} \tag{2}$$

Since the quantization noise is equally distributed within +/- Q/2, the quantization noise power Na is

$$Na = \frac{2}{Q}\int_0^{Q/2} x^2\, dx = \frac{2}{Q} \cdot \frac{(Q/2)^3}{3} = \frac{1}{12} Q^2 \tag{3}$$

If we consider a sinusoidal input signal with p-p amplitude V, the signal power is

$$S = \frac{1}{2}\int_0^{2} \left(\frac{V}{2} \sin x\right)^2 dx = \frac{1}{8} V^2 \tag{4}$$

Consequently, the power ratio signal-to-quantization noise is

$$\frac{S}{Na} = \frac{V^2/8}{Q^2/12} = \frac{V^2/8}{V^2 / (N-1)^2 \, . \, 12} \approx \frac{3}{2} N^2 \quad \text{(for } N \gg 1\text{)} \tag{5}$$

Or, by substituting by (1)

$$\frac{S}{Na} = \frac{3}{2} \cdot 2^{2n} = 3 \, . \, 2^{2n-1}$$

Expressed in decibels, this gives

$$S/N \text{ (dB)} = 10 \log (S/Na) = 10 \log 3.2^{2n-1}$$

Working this out gives :

$S/N \text{ (dB)} = 6.02 \times n + 1.76$

With a 16 bit system, a theoretic S/N ratio of 98 dB is possible; for a 14 bit system, this value is 86 dB.

4. Masking of quantization noise

Although, generally speaking (i.e. with most types of input signals), the quantization error is randomly distributed between + and - Q/2 (see Fig. 1-13.) and consequently similar to analog white noise, there are some cases in which it may become much more noticeable than the theoretical S/N figures would indicate.

The reason is mainly that, under certain conditions, quantization can create harmonics in the audio passband which are not directly related to the input signal; since in this case there is no masking effect that makes these harmonics less audible, audibility of such distortion components is much higher than in the 'classical' cases of distortion (THD etc.). Such a noise is known as 'granulation noise', or, in bad cases, it may become audible as beat tones, called 'bird singing'.

Auditory tests have shown that to make this 'digital' noise just as perceptible as 'analog' white noise, measured signal- to-noise ratio should be up to 10-12 dB higher !

To reduce this audibility, there are two possibilities :

a) To increase the number of bits sufficiently (which is very expensive)

b) To 'mask' the digital noise by a small amount of analog white noise, known as 'dither noise'.

Although such an addition of 'dither noise' actually worsens the S/N ratio slightly (several dB), the highly-audible 'granulation' effect can be very effectively masked by it.

The technique of adding 'dither' is very well known in the digital signal processing field; also in video applications, it is used to reduce the visibility of the noise in digitized video signals.

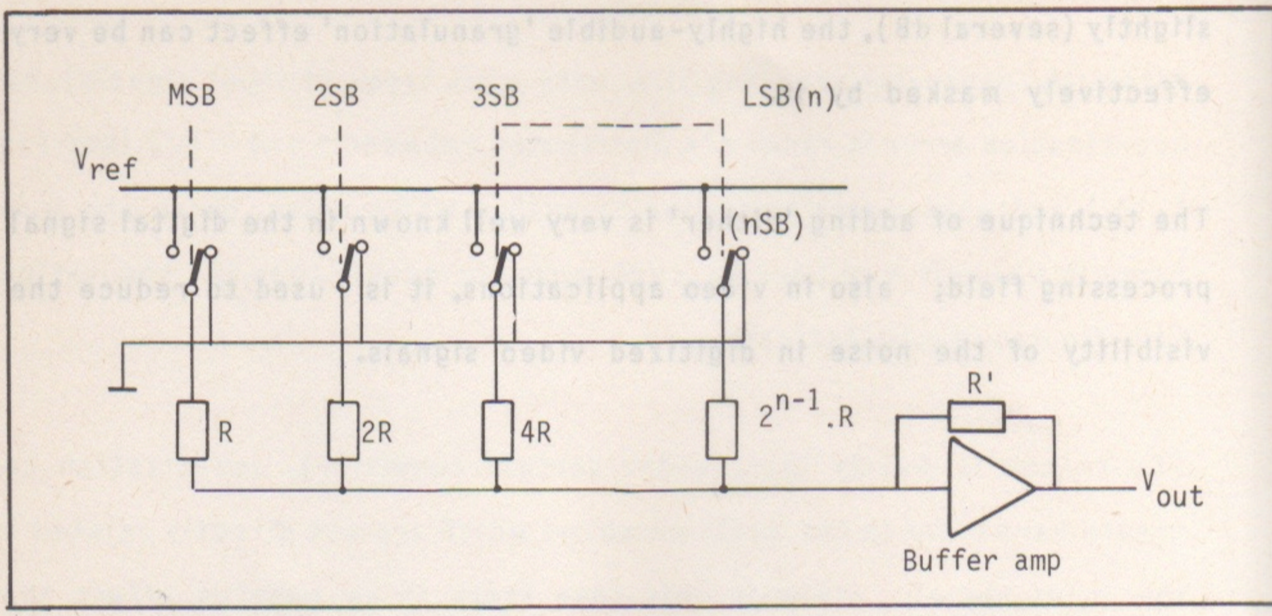


Fig. 1-17. Weighted current D/A converter

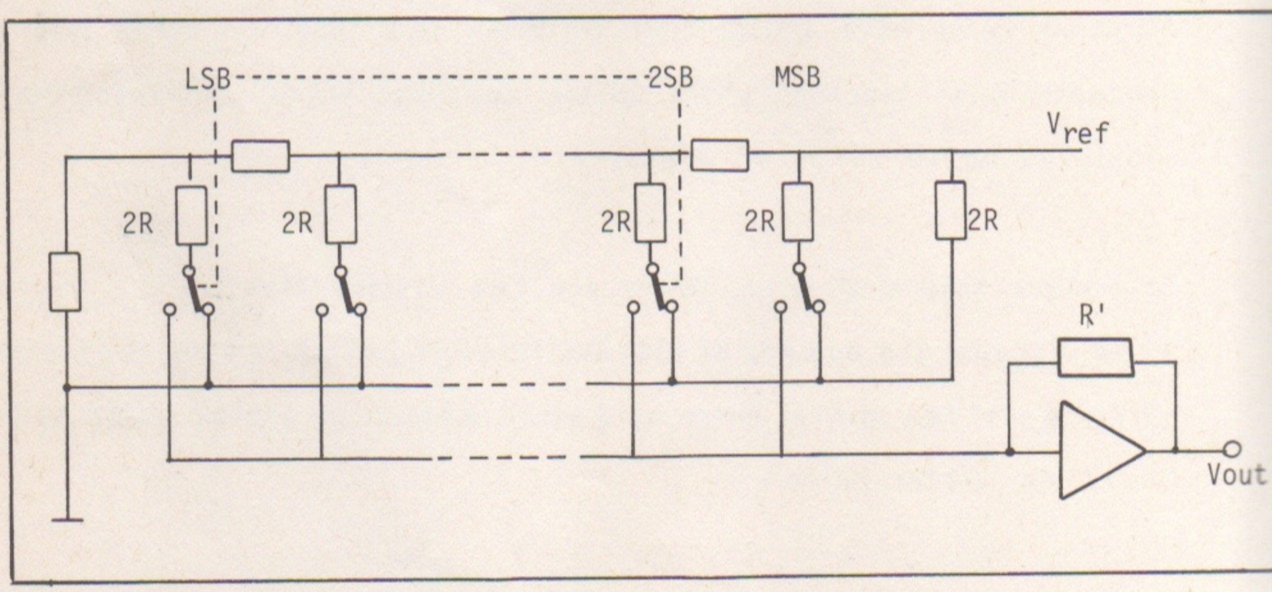


Fig. 1-18. Ladder network D/A

V. ARCHITECTURE OF A/D AND D/A CONVERTORS

1. Introduction

Obviously, one of the most important components in a digital audio system is the convertor itself. From the foregoing, it has become clear that very high-resolution is needed to obtain a satisfactory signal-to-noise ratio. At the moment, 14 bit systems (or equivalent) seem a minimum for good audio perfomance, whereas, for video applications for example, 8 bits are more than sufficient. For real professional purposes, 16 bits are required to leave some margin for further processing (e.g. filtering, mixing, etc.).

In such high-resolution, and relatively fast A/D convertors (conversion time around 20 µs), we are reaching the limits of the present-day circuit technology. This chapter will explain the basic principles generally encountered and typical specifications of A/D convertors.

2. D/A Convertors

The principle of the best known, so-called 'weighted current', D/A convertor is shown in Fig. 1-17. Each bit of the input word controls a switch in the circuit that turns On or Off a specified current. The several currents generated are added and buffered in an output amplifier.

The problem in this apparently simple circuit is the number of precision resistors needed : an n-bit convertor requires n precision resistors over a value range of 1 : 2^{n-1}. Also the required accuracy in the resistors for the highest bits is extremely high, since the accuracy of its current should be considerably smaller than the current switched by the LSB (which, in a 16 bit convertor, is 1/32.768 of the current switched by the MSB; this means an accuracy better than 0.003%!) Also this accuracy must be maintained over a long period of time and temperature range.

Therefore, in practice, other types of networks are used for high-resolution convertors, called 'ladder networks' by which some of these tremendous problems can be simplified. An example of a D/A convertor using a ladder network is shown in Fig. 1-18.

The advantages are :

- Only two values of resistors are needed, which make them easier to be matched (in value and temperature characteristic)
- Only the highest-SB's resistors must have maximum accuracy, so they can be selected from the complete batch
- 2R values can be made by using 2 times an R-value, which allows even an easier matching.

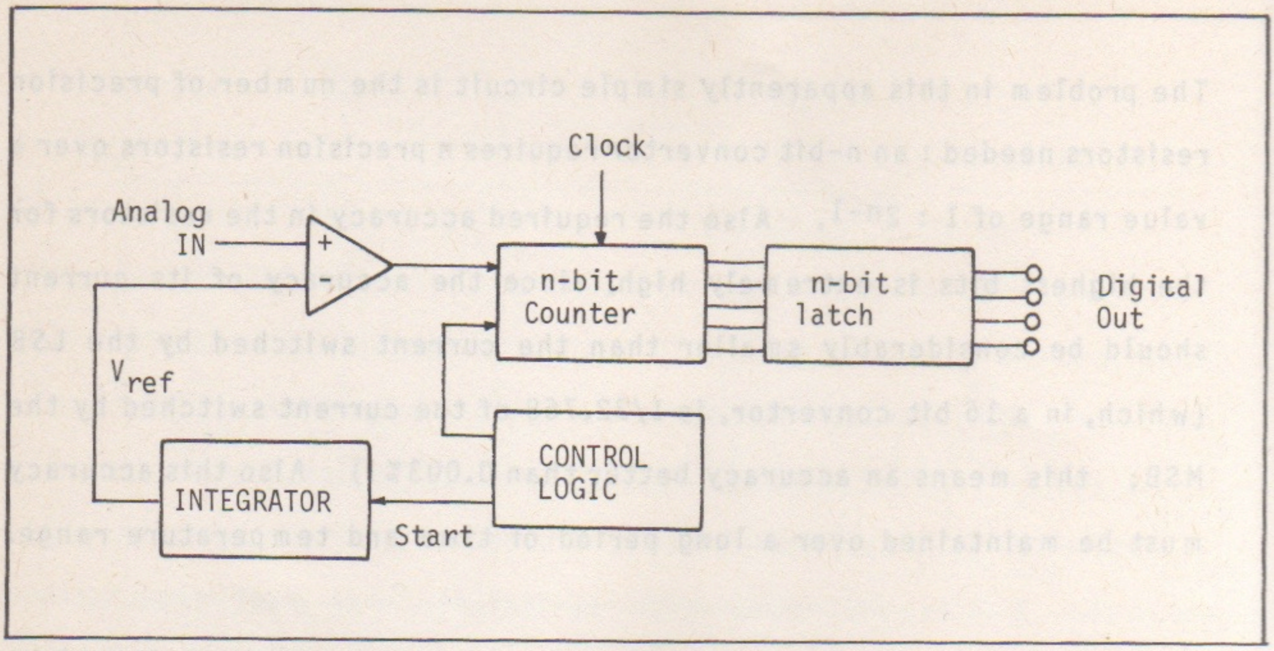


Fig. 1-19. Ramp counting A/D

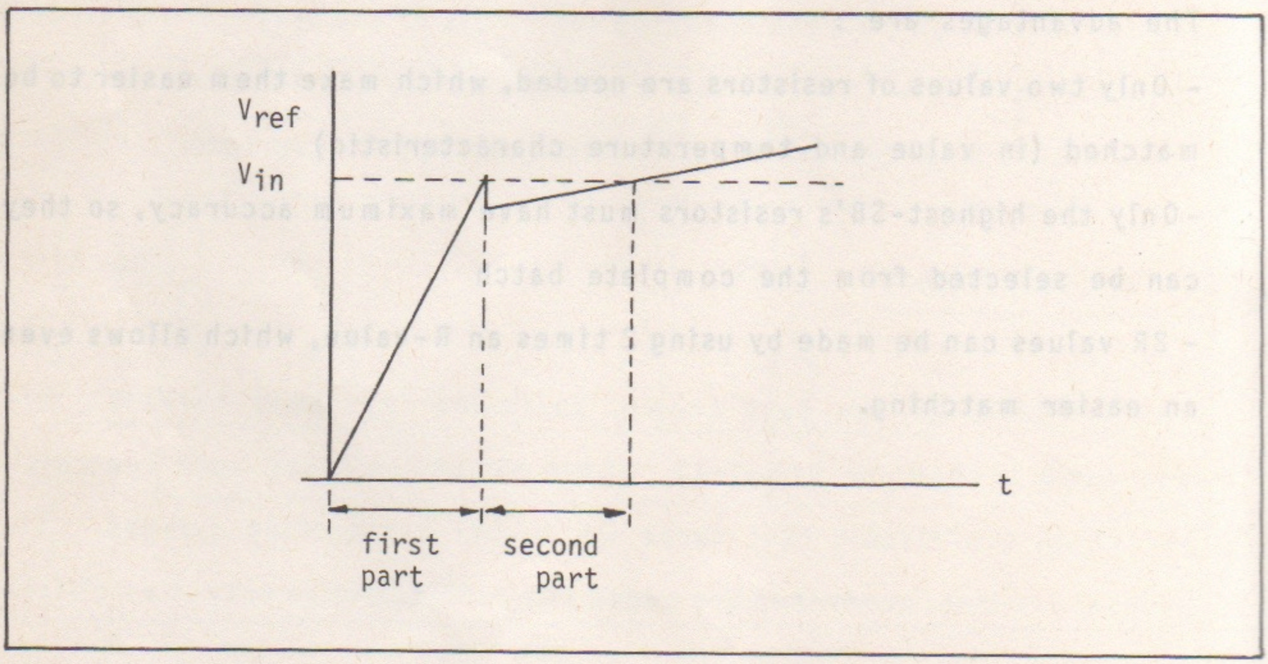


Fig. 1-20. 'Change-gear' conversion

3. A/D Convertors

3-1. Ramp Counting A/D

This type of A/D-convertor is the most typical representative of an approach, in which the analog input is converted to a number of pulses whose number is measured to obtain a digital output. (Fig. 1-19.)

A linearly rising reference voltage is generated by an integrator, and compared with the analog input voltage. At the time the integrator is started, also the counter is triggered, and starts counting, until V_{ref} becomes higher than the input signal; then the comparator changes state and stops the counting. The output of the counter is then transferred to a latch and presented at the output.

Since the time taken for V_{ref} to reach the input signal depends upon the level of the input signal, the number of pulses counted and hence the latched counter output is representative for the input voltage.

Although this type of convertor is normally too slow for audio applications, it can be used via a special technique, in which at first V_{ref} rises very fast, to produce the higher-order bits in a short time, and then much slower to determine the exact input level more accurately. This technique is commonly referred to as 'change-gear' conversion, and has been used by the BBC for 13-bit convertors. (Fig. 1-20.)

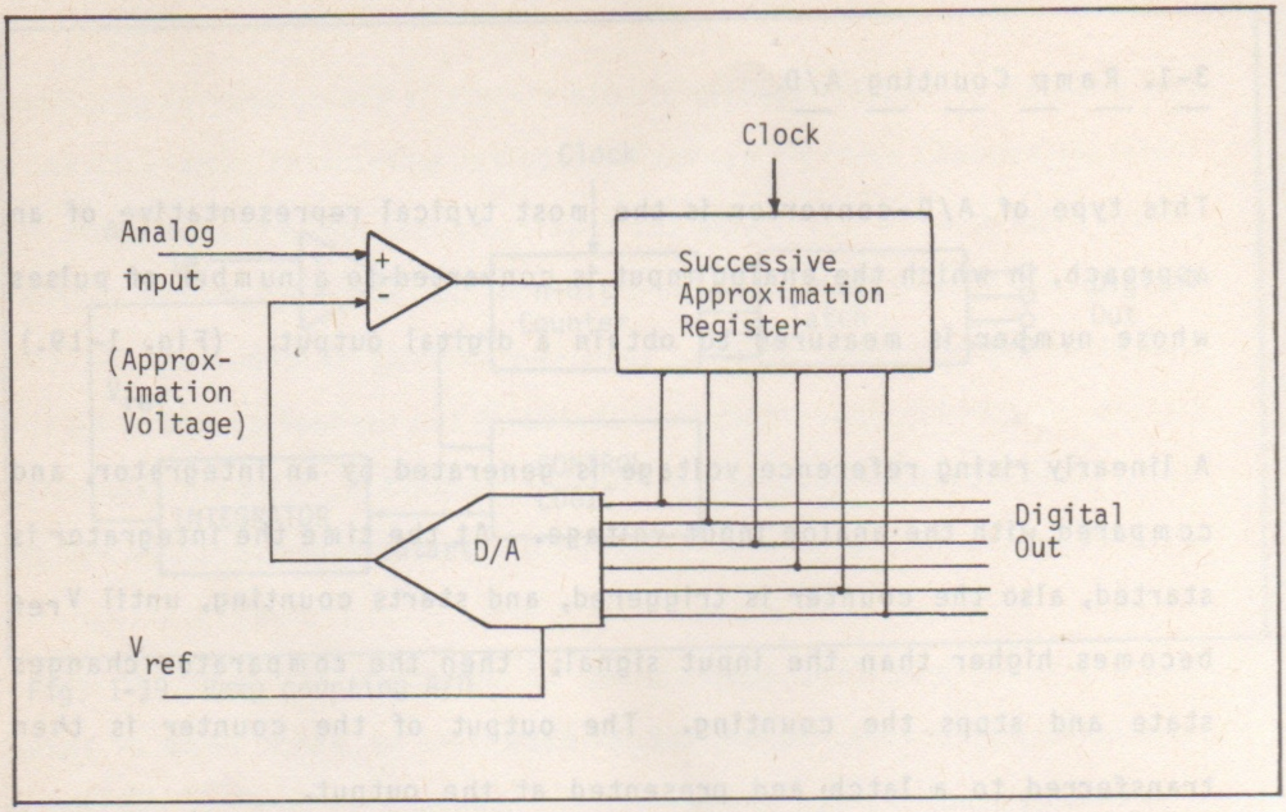


Fig. 1-21. Successive-Approximation A/D

3-2. Successive-Approximation A/D

A much faster conversion technique, and more commonly used nowadays, is "successive approximation". The block diagram of such a system is shown in Fig. 1-21.

A successive-approximation ADC consists of a D/A convertor, whose output is compared with the input voltage, and whose input is controlled by a logic circuit, called Successive- Approximation register.

At the start of conversion, only the MSB of the D/A is made '1' by the control circuit, so that its output voltage is half of full scale (see Fig. 1-20). If the input voltage is smaller than this value, the MSB is made 0, and the 2SB is made 1. If the input is now bigger, the 2SB remains 1 and the 3SB is made 1, etc.

In this way, the output from the D/A gradually comes closer to the input voltage, until the LSB is tried out and the difference is minimized.

The total conversion time obviously is equal to the clock rate times the number of bits, which can be much smaller than in a Ramp Counting ADC; obviously its accuracy is determined by the internal DAC.

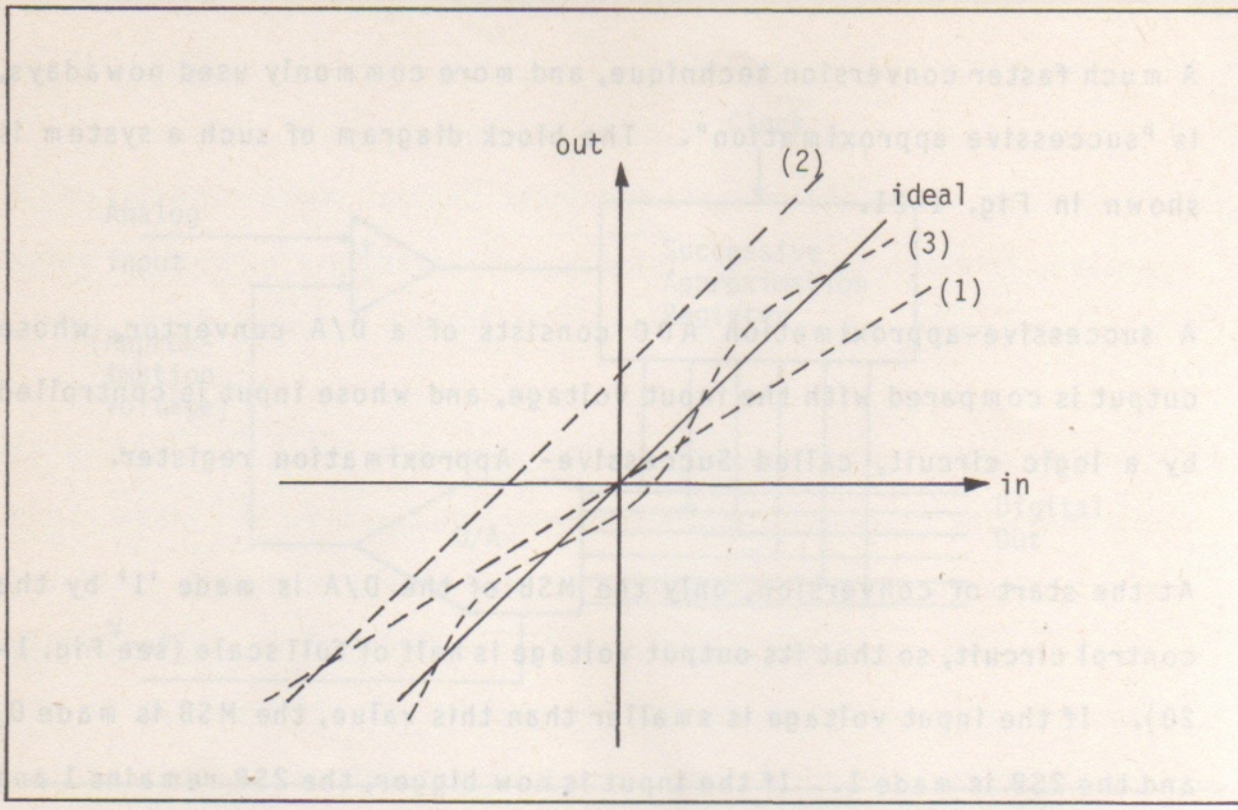


Fig. 1-22. Deviations of A/D - D/A system transfer characteristic
(1) Gain error - (2) Offset error - (3) Linearity error

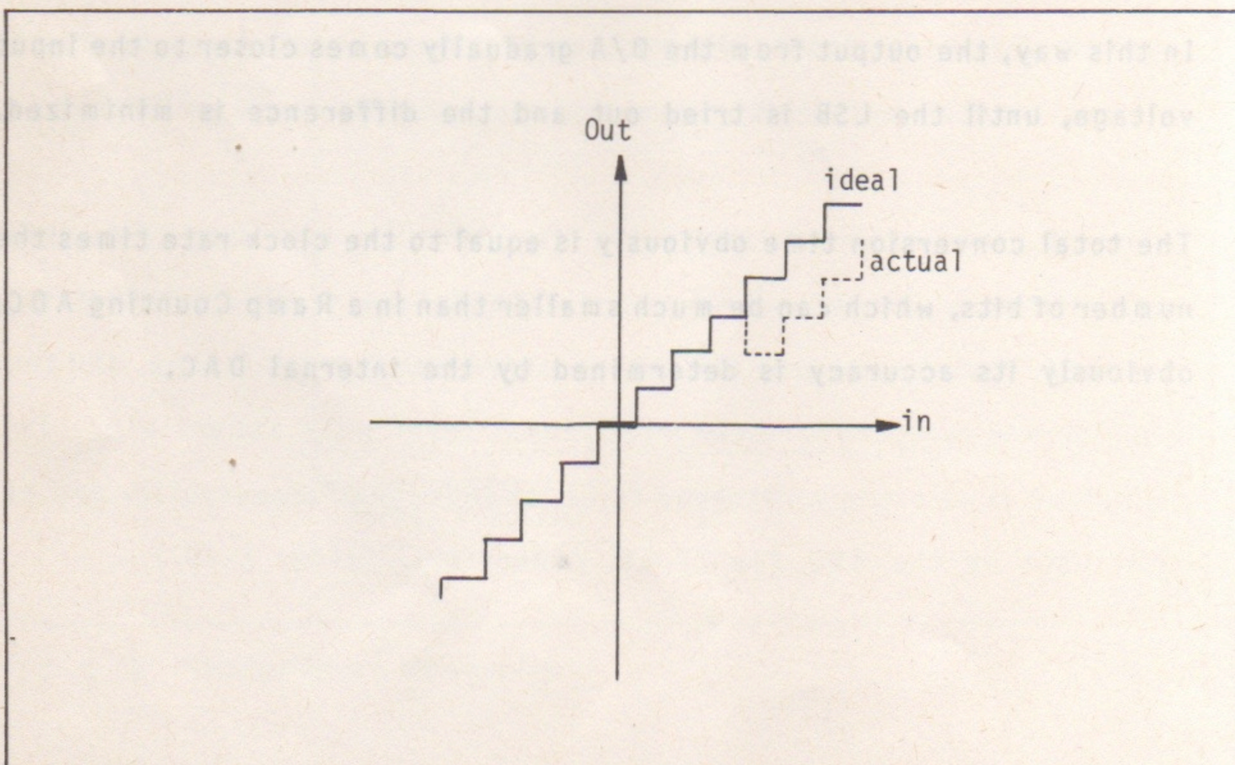


Fig. 1-23. Example of Non-monotonicity

4. Errors in Practical A/D - D/A Convertors

Practical A/D and D/A convertors will unavoidably induce deviations from the ideal input-output characteristic of an A/D - D/A combination, which is of course linear (if we do not consider the relatively small quantizing steps).

These variations can be of various sorts :

- gain error : this is not so important for audio application, since the output amplifier circuitry can compensate this
- offset error : this is a dc-shift in the characteristic; although not so serious, it shifts the output signal asymmetrically, so that the total overload margin may be (slightly) affected. However, in companding systems it can be more serious (see further)
- non-linearity : this causes similar problems as in analog circuits, and must consequently be kept to a minimum. Non- linearities can come either from the analog part of the convertor, or also from the digital circuit, e.g. if the quantization steps are not uniform, which is known as 'differential linearity error'. If this error is bigger than one LSB, the transfer characteristic can even change direction, which is called 'non-monotonicity'. (Fig. 1-23.) This must certainly be avoided in digital audio applications.

VI. OVERVIEW OF A/D CONVERSION SYSTEMS

1. Linear (or uniform) Quantization

In the examples which we have given in the foregoing part, the quantization intervals Q were all identical (see Fig. 1-12.). Such a quantization system is commonly known as 'linear quantization' (although a better term is 'uniform quantization).

From a point of view of circuit simplicity and uncompromised conversion quality, the linear system is certainly the best. However, the linear system is rather costly in terms of required bandwidth and conversion accuracy.

Indeed, one audio channel converted to 16 bits with a sampling frequency of 44.056 kHz gives a bit stream of at least 16 x 44.056 = 705 kbits/s, which requires a bandwidth of 350kHz (17.5 times the bandwidth of the original signal). In practice, even this is not sufficient because mostly more bits will be added for synchronization , error correction and other purposes.

It is obvious that, since the beginning of PCM-telephony, ways have been looked for to reduce the bandwidths that digitized audio signals require, since especially here the capacity of the transmission channels should be optimized. Most of these techniques can also be used for high-quality audio.

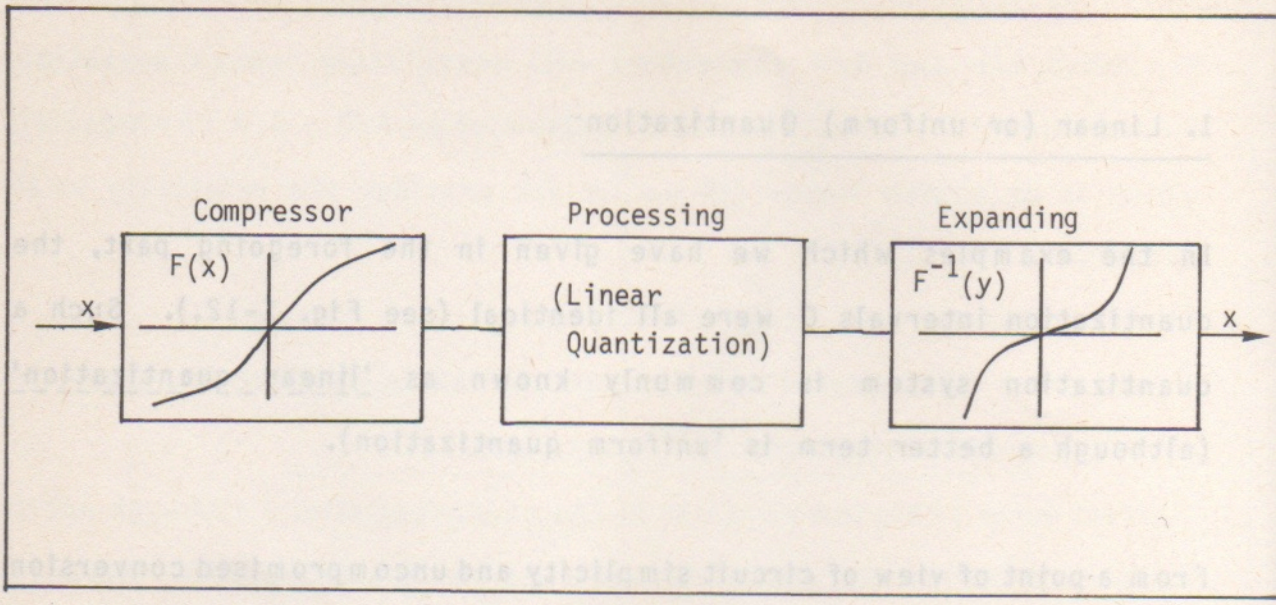


Fig. 1-24. Non-linear quantization principle

Another important aspect is that the price of A/D convertor modules does increase drastically when we go, for instance, from 12 bits to 16 bits (x 100 and more); therefore, it is interesting if for instance 16 bit performance could be obtained by using a 12 bit module. This, however, is not possible with linear quantization, which is bound by the physical limits of the quantization process.

2. Companding System

If, in a quantizer, the quantization intervals Q are not identical, we talk about 'non-linear quantization'. It is for instance perfectly possible to change the quantization intervals according to the level of the input signal. In general, in such systems, small level signals will be quantized with more closely spaced intervals, while larger incoming signals can be quantized with bigger quantization intervals.

This is possible since, in such case, the larger signals will more or less mask the unavoidably higher noise levels of the coarser quantization. In this way, it is theoretically possible to obtain, for instance, the performance of a 14 bit convertor with a 10 bit convertor module. Such a non-linear quantization can be thought to consist of a linear system, to which a compander has been added. In such a system, the input signal is first compressed following some non-linear law F(x), then linearly quantized, processed and then after reconversion, expanded by the reverse non-linearity $F^{-1}(y)$. (See Fig. 1-24.)

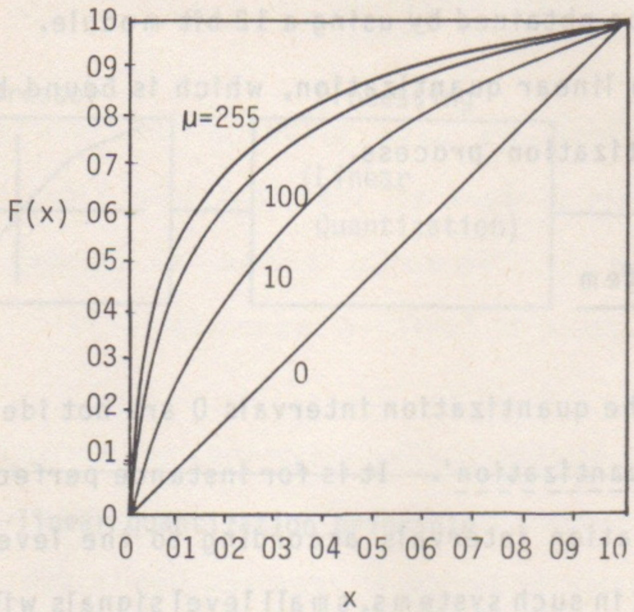


Fig. 1-25. μ-law compressor curves

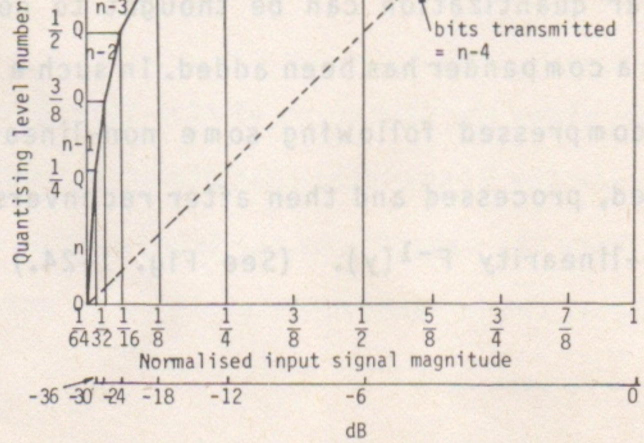


Fig. 1-26. A-low characteristic

Consequently, this is very similar to the companders used in the analog field (e.g. Dolby, DBX and many others).

A compressor curve used quite extensively in digital telephony, i.e. for digitization of speech, is the so-called "μ-law"- curve. This curve is characterized by the following formula

$$F(x) = V \frac{V \log(1 + \mu x/V)}{\log(1 + \mu)}$$

Curves for this equation are shown in Fig. 1-25., for several values of μ. In Europe, another companding law, called the "A- law" is more generally used. (Fig. 1-26.)

The (dual) formula for the "A-law" is :

$F(x) = Ax / 1 + \log A \quad \text{for } o \leqq x \leqq V/a$

$F(x) = V + V\log(Ax/V) / 1 + \log A \quad \text{for } V/a \leqq x \leqq V$

In practice, it is important that the non-linearities at the input and the output of the system are very closely matched. This is difficult to achieve with standard analog techniques; therefore, the non-linearity will mostly be built-in in the conversion system itself.

The big advantage of these companded systems is, that the S/N ratio is less dependent on the level of the input signal; the disadvantage, however, is that the noise level follows the level of the signal, which may lead to audible noise modulation.

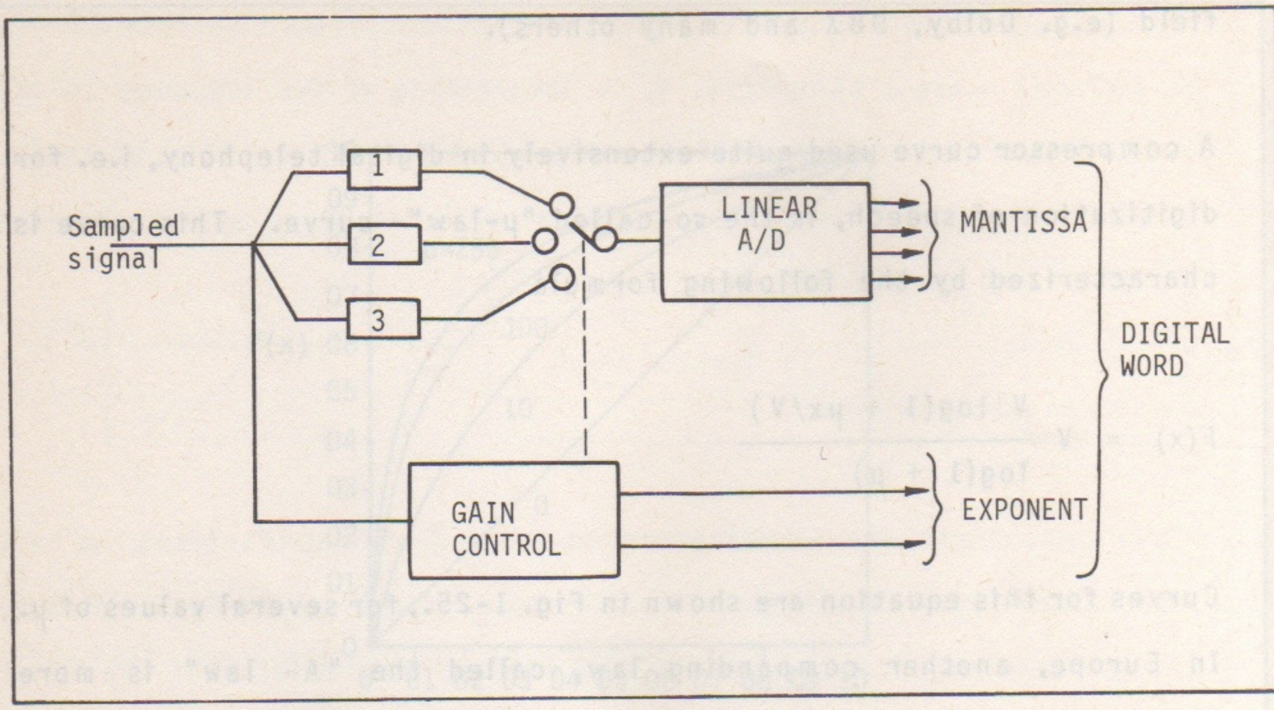


Fig. 1-27. Principle of Floating-Point Convertor

Mostly, in addition to companding, pre-emphasis is used to improve the S/N and to make the noise modulation less audible.

Systems using companding, as described above, are not generally used for quality audio, although the Deutsche Bundespost is willing to use a similar system for its radio network and (eventually) international links.

3. Floating-Point Conversion

A special case of non-linear quantization, that is presently used in professional audio, is the so-called "Floating-point convertor" (Fig. 1-27.)

In a floating-point convertor, the sampled signal is sent through several, selectable paths with a different gain; depending on the input level of the signal, the appropriate gain is selected by a logic monitor circuit, in order to make maximum use of the linear A/D convertor, without overloading it. The output from the A/D convertor (called 'mantissa', as an analogy with logarithmic annotation) is now of course meaningless without a way to indicate the gain that was originally selected. This information is provided by a logic output from the monitor circuit, called 'exponent'. Exponent and mantissa, taken together, give an unambiguous digital word, that can be reconverted to the original signal by selecting the corresponding (inverse) gains in the decoding stage.

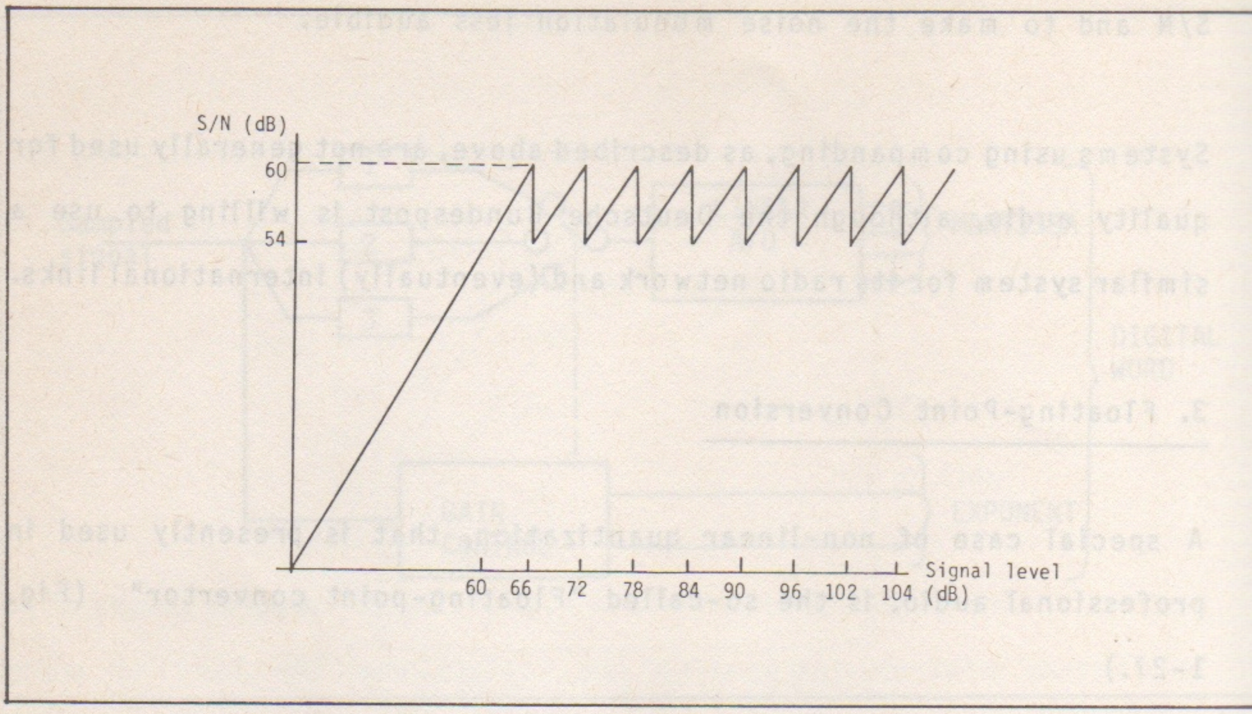


Fig. 1-28. Signal-to-noise ratio of floating-point convertor

In this way two bits of exponent can indicate four different gains. If we select these gains as 0, 6, 12 and 18 dB for instance, the two additional bits provide an increase of 18 dB of the dynamic range of the basic system.

Of course, since the signal level will determine the gain of the basic system, also here there will unavoidably occur noise modulation, which may specially become audible when for instance an high-level, low frequency signal occurs; in this case, the modulation noise will not be masked by the signal.

Because of the noise modulation, one must make a distinction between the dynamic range and the signal-to-noise ratio.

The dynamic range can be defined as :

$$\frac{\text{maximum signal level (RMS)}}{\text{RMS level of quantization noise WITHOUT signal}}$$

whereas the signal-to-noise ratio is :

$$\frac{\text{signal level (RMS)}}{\text{RMS level of quantization noise WITH signal}}$$

A curve for the signal-to-noise ratio of a typical floating- point convertor with a 10 bit mantissa, a 3-bit exponent and 6 dB gain steps is shown in Fig. 1-28.

Although, theoretically, this system provides the same dynamic range as a 17 bit linear system (over 100 dB), it can be seen that the S/N is much worse, in fact unacceptable for high- quality purposes.

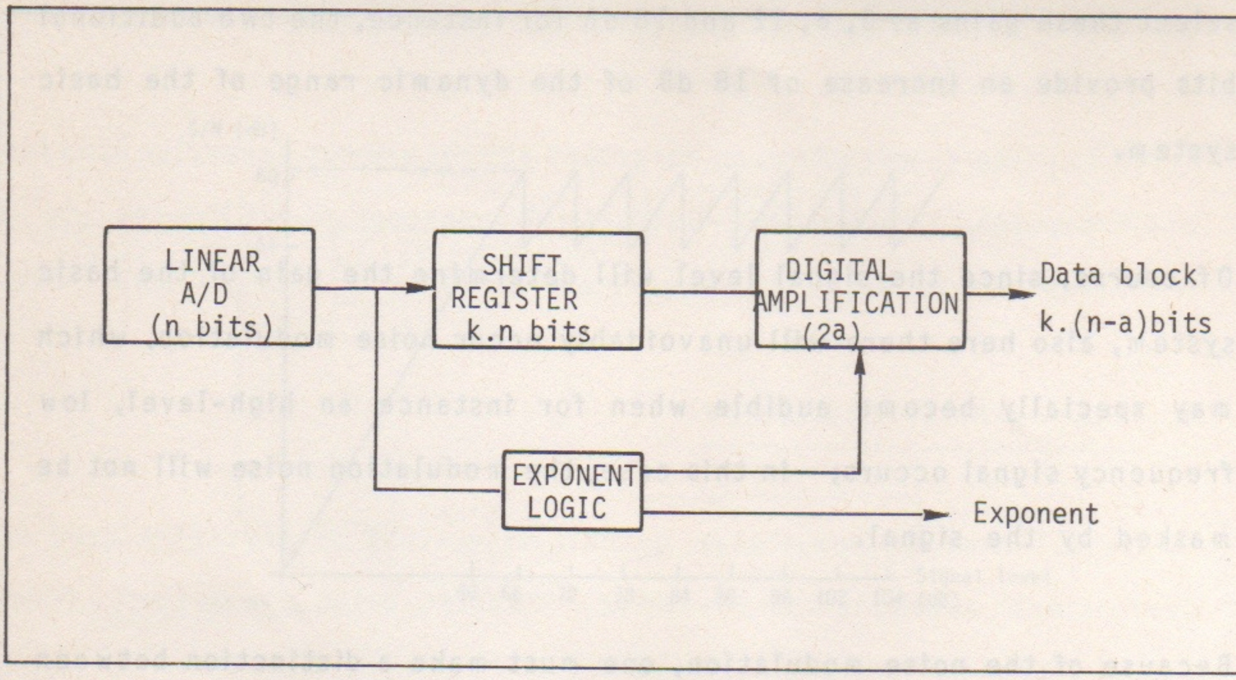


Fig. 1-29. Example of block floating-point encoder

In spite of this, high-quality floating-point convertors having e.g. a 13 bit mantissa and 3 bit exponent, are still being considered for digital audio purposes, since they are considerably cheaper than linear systems.

4. Block Floating-Point Conversion

When economy of bandwidth (i.e. bit-rate) is of utmost importance, so-called Block Conversion can be used. This technique is also known as 'Near-Instantaneous Companding' (in contrast to 'normal' floating point, or other companding systems). The term Near-Instantaneous means that not every sample is scaled by an exponent, but a number of successive samples (mostly 32). Each block of 32 samples is then followed by a scale factor word, so that, at the receiving end, each block can be correctly scaled up again.

This system is rather expensive as far as hardware is concerned, but permits significant reduction in bit rate. Consequently, a typical application is digital transmission of audio signals in radio networks.

Subjective tests have shown that an original 14 bit system compressed to 10 bits is almost indistinguishable from 13 bit linear, although the limitations mentioned in the foregoing paragraph remain valid.

An example of such a system is the BBC's NICAM-3 (Near-Instantaneous Companding Audio Multiplex) which permits to transmit six audio channels over one (standard) telephony 2048 kbit/s circuit.

5. Differential PCM and Delta Modulation

Instead of transmitting the exact binary value of each sample, it is possible to transmit only the difference between the current sample and the foregoing one.
Since this difference is generally small, a smaller number of bits can be used with no apparent degradation in performance.

The previous sample is stored during one sample period, and added to the difference signal to obtain the current sample. The current sample then becomes the reference for the next one.

Differential PCM in fact is a special type of 'Predictive Encoding'. In such encoding schemes, a prediction is generated for the current sample, based upon the past data; the correcting signal is then the difference between the prediction and the actual signal.

It is clear that, when the sampling rate is increased, the differences between the foregoing and the present samples become smaller, so that, for very high sampling rates, only 1 bit is needed for the error signal to indicate the sign of the error; in this case we talk about 'Delta Modulation'.

In a further step, the transmitted data can be used not only to indicate the sign of the error, but also to indicate the step size (some kind of floating-point encoding). For example, a continuous series of ones means that the signal is quickly increasing, so the step-size can be increased; if ones and zeros are alternating, step size can be reduced.

Such strategies are called 'Adaptive Differential PCM' (the quantization interval is changed) or 'Adaptive Delta Modulation' (the step size is changed).

Although these techniques have some interesting theoretical and practical properties, it is presently difficult to use them for high-quality applications.

VII. CONVERSION CODES

In principle, any digital coding system can be adopted to indicate the different analog levels in A/D or D/A - conversion, as long as they are properly defined. Some, however, are better for certain applications than others. Two main groups exist : Unipolar codes and Bipolar codes. The Bipolar codes give information on both the magnitude and the sign of the signal, which makes them preferable for audio applications.

1. Unipolar codes

Depending upon the application, the following codes are mostly used :

a) Natural Binary Code

The MSB has a weight of 0.5 (2^{-2}), the 2SB has 0.25 (2^{-2}) etc. until the LSB (nth SB) which has a weight of 2^{-n}.

Consequently, the maximum value that can be expressed (when all bits are one) is $1 - 2^{-n}$, or in other words Full-scale minus one LSB.

b) BCD CODE

The well-known 4 bit code in which the maximum value is 1001 (decimal 9), after which we reset to 0000. A number of such 4 bit codes is combined in case we want, for instance, a direct read-out on a numeric scale such as in Digital Voltmeters. This code is consequently not used for audio.

	Decimal Fraction					
Number	Positive Reference	Negative Reference	Sign + Magnitude	Two's Complement	Offset Binary	One's Complement
+ 7	$+\frac{7}{8}$	$-\frac{7}{8}$	0 1 1 1	0 1 1 1	1 1 1 1	0 1 1 1
+ 6	$+\frac{6}{8}$	$-\frac{6}{8}$	0 1 1 0	0 1 1 0	1 1 1 0	0 1 1 0
+ 5	$+\frac{5}{8}$	$-\frac{5}{8}$	0 1 0 1	0 1 0 1	1 1 0 1	0 1 0 1
+ 4	$+\frac{4}{8}$	$-\frac{4}{8}$	0 1 0 0	0 1 0 0	1 1 0 0	0 1 0 0
+ 3	$+\frac{3}{8}$	$-\frac{3}{8}$	0 0 1 1	0 0 1 1	1 0 1 1	0 0 1 1
+ 2	$+\frac{2}{8}$	$-\frac{2}{8}$	0 0 1 0	0 0 1 0	1 0 1 0	0 0 1 0
+ 1	$+\frac{1}{8}$	$-\frac{1}{8}$	0 0 0 1	0 0 0 1	1 0 0 1	0 0 0 1
0	0 +	0 -	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0
0	0 -	0 +	1 0 0 0	(0 0 0 0)	(1 0 0 0)	1 1 1 1
- 1	$-\frac{1}{8}$	$+\frac{1}{8}$	1 0 0 1	1 1 1 1	0 1 1 1	1 1 1 0
- 2	$-\frac{2}{8}$	$+\frac{2}{8}$	1 0 1 0	1 1 1 0	0 1 1 0	1 1 0 1
- 3	$-\frac{3}{8}$	$+\frac{3}{8}$	1 0 1 1	1 1 0 1	0 1 0 1	1 1 0 0
- 4	$-\frac{4}{8}$	$+\frac{4}{8}$	1 1 0 0	1 1 0 0	0 1 0 0	1 0 1 1
- 5	$-\frac{5}{8}$	$+\frac{5}{8}$	1 1 0 1	1 0 1 1	0 0 1 1	1 0 1 0
- 6	$-\frac{6}{8}$	$+\frac{6}{8}$	1 1 1 0	1 0 1 0	0 0 1 0	1 0 0 1
- 7	$-\frac{7}{8}$	$+\frac{7}{8}$	1 1 1 1	1 0 0 1	0 0 0 1	1 0 0 0
- 8	$-\frac{8}{8}$	$+\frac{8}{8}$		(1 0 0 0)	(0 0 0 0)	

Table 1-1. Commonly used Bipolar codes

C) Gray Code

Is used when the advantage of changing only one bit per transition is important, for instance in position encoders where inaccuracies might otherwise give completely erroneous codes; is easily convertible to binary. Not used for audio.

2. Bipolar Codes.

These codes are similar to the unipolar natural binary code, but one additional bit, the sign bit, has been added.

The best-known codes are sign-magnitude, offset binary, two's complement and one's complement. The structure of each of these codes is compared in table 1-1.

a) Sign-Magnitude

The magnitude of the voltage is simply expressed by its normal (unipolar) binary code, and a sign bit is simply added to express the polarity.

An advantage is that the transition around zero is rather simple, in contrast to the other codes; however, it is more difficult to process (computation) and there are two codes for zero.

b) Offset Binary

This is a natural binary code, but with zero at minus full scale; this makes it relatively easy to implement and for computation use.

---

c) Two's Complement

Is very similar to offset binary, but with the sign bit inverted. Arithmetically speaking, two's complement is taken by complementing the positive value and adding 1 LSB.

e.g. +2 = 0010

-2 = 1101 + 1 = 1110

It is very useful for computation, since for instance, positive and negative numbers added together always give zero (disregarding the extra carry).

```
e.g.  0010
     +1110
     ------
      0000
```

It is the code almost universally used for digital audio; there is however, (as with offset binary) a rather big transition at zero (all bits change from '1 to '0)

d) One's complement

Here negative values are full complements of positive values. This code is not so commonly used as the foregoing ones.

Furthermore, there exist variations on the foregoing codes such as modified sign-magnitude, complementary offset binary, etc. These will not be discussed here.

PART II

Registration Methods and Formats

PART II

REGISTRATION METHODS AND FORMATS

I. OUTLINE

The field in which the advantages of using digital techniques are mostly immediately recognizable, is the field of magnetic recording. In the analog world, it is this magnetic recording that creates the biggest deterioration of the original sound : it is difficult, if not impossible, to obtain a flat frequency response at all signal levels, signal-to-noise ratio is limited to some 70 dB, the sound is deteriorated by speed variations of the recorder mechanism, there exist crosstalk and print-through problems, and each additional copying, e.g. during mixdown, deteriorates the characteristics even further. In addition to this, to keep the equipment within close specifications, as required in a professional environment, frequent and costly realignment and maintenance are required.

Recording of sound under the form of digital data, on the contrary, virtually solves all of these drawbacks :

- It is perfectly possible to obtain a very flat frequency response at any signal level.

- Signal-to-noise ratio depends only on the conversion equipment and can theoretically be made arbitrarily high.

- Wow and flutter is virtually non-existent

- Crosstalk and print-through do not deteriorate the sound (in fact do not exist as such)

- Theoretically, an infinite number of digital copies is possible without degradation of the sound.

Recording of digital data, however, presents some specific problems :

- As mentioned already, the required bandwidth is increased dramatically as compared to the original signal.

- For this and other reasons, specific codes must be used for recording (in contrast to the simple data codes mentioned before).

- Contrary to analog recordings, even very small drop-outs on the tape can have disastrous audible effects, and, consequently, they must be electronically corrected. For that effet, additional data must be recorded on the tape which further increase the necessary bandwidth.

- Synchronization of the recorded data stream is necessary to allow for reconstruction of the recorded words. The additional synchronization bits again increase the bandwidth.

- In contrast to analog recordings, editing is very complicated and requires additional complex circuitry. For tape-cut editing, common practice in the analog recording field, a very strong error correction scheme together with interleaving are needed. Even then, very careful handling is a must; for instance, the tape cannot be touched with bare fingers.

II. CODES FOR DIGITAL MAGNETIC RECORDING

1. Introduction

The binary codes, representing the original audio signal, can be recorded on tape in two ways : either directly, which is the case in stationary-head recorders, or after frequency- modulation, which is the case in helical-scan recorders.

In case frequency-modulation is used, the data can be modulated as they are, which is mostly so in so-called 'Non-return-to- zero' format (see further). If they are recorded directly, however, they have to be transformed to some new code to obtain a recording signal which matches as well as possible the properties of the magnetic (or eventually, in case of discs, optical) recording channel.

This code should have a format so as to obtain the highest bit density permitted by the limiting characteristics of the recording channel (frequency response, drop-out rate etc.). Also, its low-frequency (or DC-) content should be decreased or eliminated, since as is commonly known, magnetic recorders cannot reproduce very low frequencies or DC.

The coding of binary data in order to comply with the demands of magnetic recording is often referred to as 'channel coding'.

2. Non-return to zero (NRZ)

This code is one of the oldest and best known of all 'channel codes'. Basically, 'one's are represented by positive magnetization, and 'zero's by negative magnetization. In fact, the 'normal' serial data in digital equipment are mostly in NRZ. A succession of the same codes presents no change in the signal, so that there can be a significant low-frequency content, which is undesirable for stationary-head recording. We say that simple NRZ is not a 'Run-Length Limited' Code.

For helical-scan (video-) recording however, the data are FM-converted before being recorded, so this property is not so important. NRZ is therefore commonly used in such formats as PCM-1600 and the EIAJ-format (PCM-100, -10).

Furthermore, there exist several variations on NRZ for optimizing various applications, but we will not discuss them here.

3. Bi-phase

Similar to NRZ, but extra transitions have been added at the beginning of every data bit interval. As a result, DC-content is eliminated and synchronization becomes easier, but the density of data transitions has been doubled.

This code (and its variants) is also known as Manchester Code, Frequency-shift-keying (FSK) and Frequency-modulation (FM).

4. Modified-frequency-modulation (MFM)

Also called Miller Code or Delay-modulation. Ones are coded with transitions in the middle of the bit cell, isolated zeros are ignored, and between pairs of zeros, a transition is inserted.

This code requires almost the same, relatively low bandwidth as NRZ, but has a reduced DC-content. The logic needed for decoding is more complicated.

A variation is the so-called Modified-modified-frequency- modulation (M^2FM).

5, 3-Position Modulation (3PM)

This is a code, which permits to obtain very high packing densities, but which requires a rather complicated hardware.

In principle, 3PM-code is obtained by dividing the original NRZ-data in blocks of 3; each block is then converted to a 6- bit 3PM-code, which is especially designed to optimize the maximum and minimum run lengths. In this way, minimum time between transitions is two times the original (NRZ-) clock rate, whereas the maximum is six times the original rate.

For detection, however, a clock signal twice as high as the original signal is needed; consequently, this reduces the jitter margin of the system. This clock is normally recovered from the data itself, which have an high harmonic content around the clock frequency.

6. High Density Modulation - 1 (HDM-1)

This a variation upon the 3PM-system; Density ratio is the same as 3PM, but clock recovery is easier and required hardware simpler. It is proposed by Sony for stationary-head recording.

7. Eight-to-Fourteen Modulation (EFM)

This code is used for the proposed Compact Disc Digital Audio System. The principle is again similar to 3PM but each block of 8 data bits is converted into 14 channel bits, to which 3 extra bits are added for merging (synchronization) and low-frequency suppression; in this way, a good compromise is obtained between clock accuracy (and possible detection errors), minimum DC-current (in disc systems, low-frequencies in the signal give noise in the servo systems), and hardware complexity. Also this modulation system is very well suitable for combination with the error-correction system proposed for the same format (see further).

III. PRINCIPLES OF ERROR CORRECTION

1. Type of Code Errors

When digital signals are recorded and subsequently played back, many types of code errors can occur. Now, in the digital field, even small errors can cause disastrous audible effects : suppose for instance that only the most-significant bit is erroneously detected as '1' instead of '0' : this will make the signal rise (for a short time) by half its maximum value, which will give a quite audible 'click'! Since this is entirely unacceptable in the very-high-quality context that can be expected from digital audio, a lot of efforts must be made to detect, and subsequently correct, as many errors as possible without making the circuitry extremely complicated (or the recorded bandwidth extremely high).

We can basically distinguish the following causes for code errors :

1-1. Dropouts

Dropouts are caused by dust or scratches on the magnetic tape (or record surface). They mostly cause relatively long-time errors, in which long 'bursts' of related data are all lost together.

In digital magnetic recording, dropouts consequently cause a very difficult situation for the decoding section.

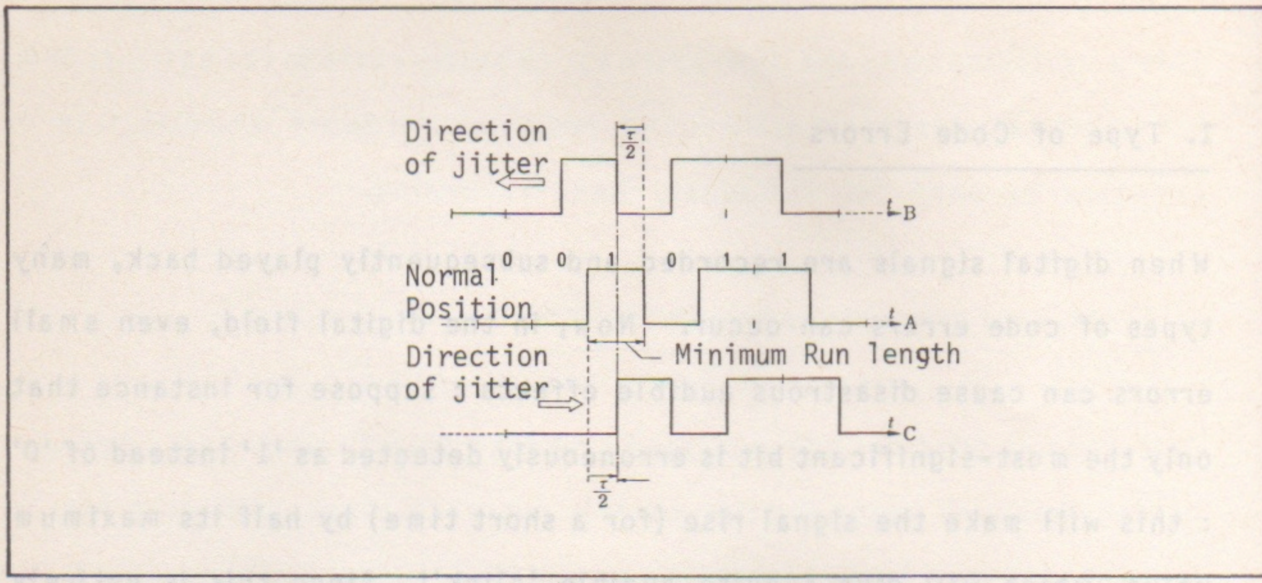


Fig. 2-1. Definition of jitter margin

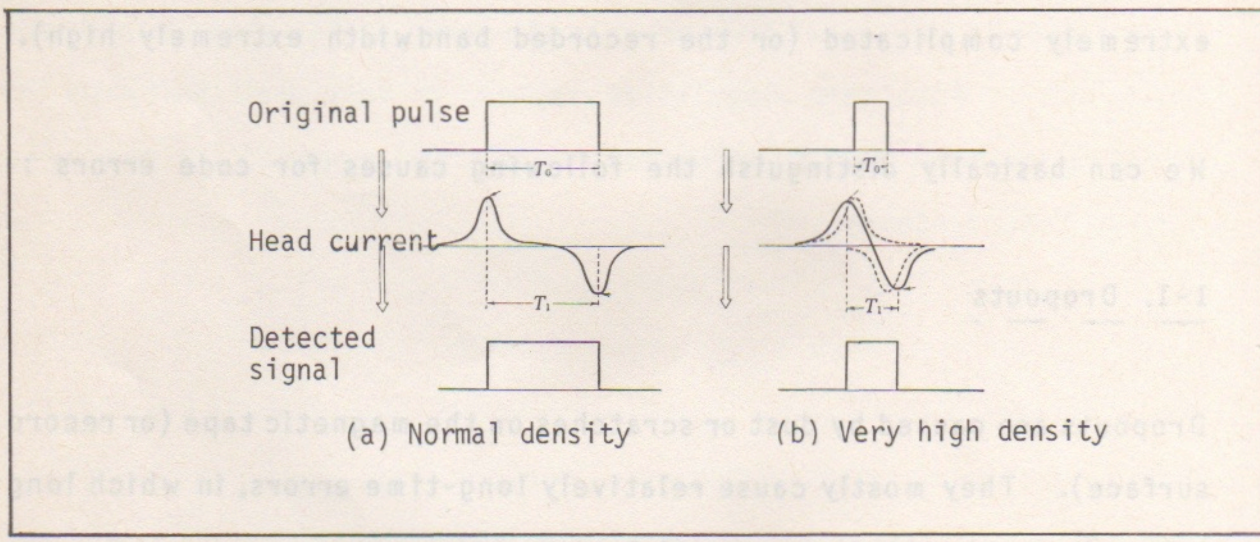


Fig. 2-2. Cause of intersymbol interference

1-2. Jitter

In magnetic recorders, a certain amount of jitter is unavoidable, due to mechanical improperties of the tape transportation mechanism. Jitter can cause random errors of isolated data. We can define jitter margin as the maximum amount of jitter which still permits to detect the data correctly; if the minimum run length of the signal is τ , then the jitter margin will be $\tau/2$. Fig. 2-1. shows this with an NRZ-signal as an example.

1-3. Intersymbol Interference

In direct (stationary head-) recording, a pulse is not recorded as a pulse, but as a positive current followed by a negative current (see Fig. 2-2.).

This causes the actual period of the signal that is read on the tape (T1) to be higher than the bit period itself (T0). Consequently, if the bit rate is very high, the detected pulse will be wider than the original pulse (b) in the figure, and interference may occur between adjacent bits, which is known as Intersymbol Interference or Time Crosstalk. Intersymbol Interference also causes random errors, depending upon the bit situation.

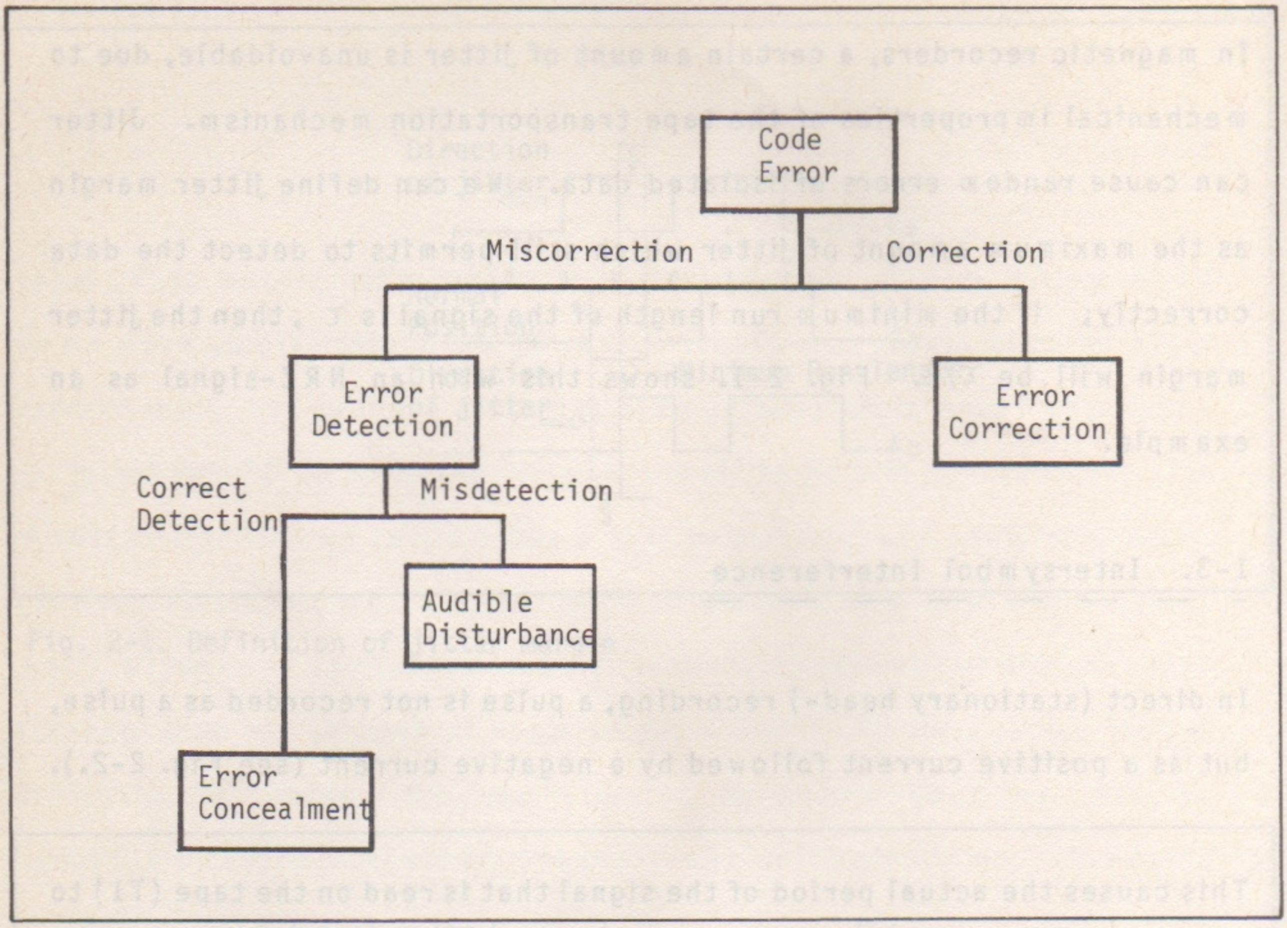


Fig. 2-3. Mechanism of error compensation

1-4. Noise

Noise may have a similar effect as dropouts (differentiation between both is often difficult), but also random errors may occur in case of pulse noise.

1-5. Editing

Tape editing always destroys some information on the tape, which consequently must be corrected. Whereas with electric editing, the errors can be kept to a minimum, tape-cut editing will always cause very long and serious errors.

2. Error Compensation Mechanism

The possible mechanisms for compensation are shown in Fig. 2-3. If the errors can be corrected by the error correction system (the principles of which are explained below), there is no problem since all data that are passed to the D/A stage will be OK. Errors, which cannot be corrected, must first be detected, after which an error concealment mechanism can be switched in. If the error detection system fails however, entirely faulty data may be sent to the D/A convertor which will give an audible disturbance in the signal.

3. Error Detection

3-1. Simple Parity Checking

To detect whether a certain data word contains an error, a very simple way is to add one extra bit to it, which is given the value 0 or 1 depending upon the number of 1's in the word itself.

Two systems are possible :

- Odd parity : here we will make the number of 1's odd.

  Example (4 bit data) :

1110	0
data	parity

1001	1
data	parity

- Even parity : we will make the number of 1's even.

  Example :

1110	1
data	parity

1001	0
data	parity

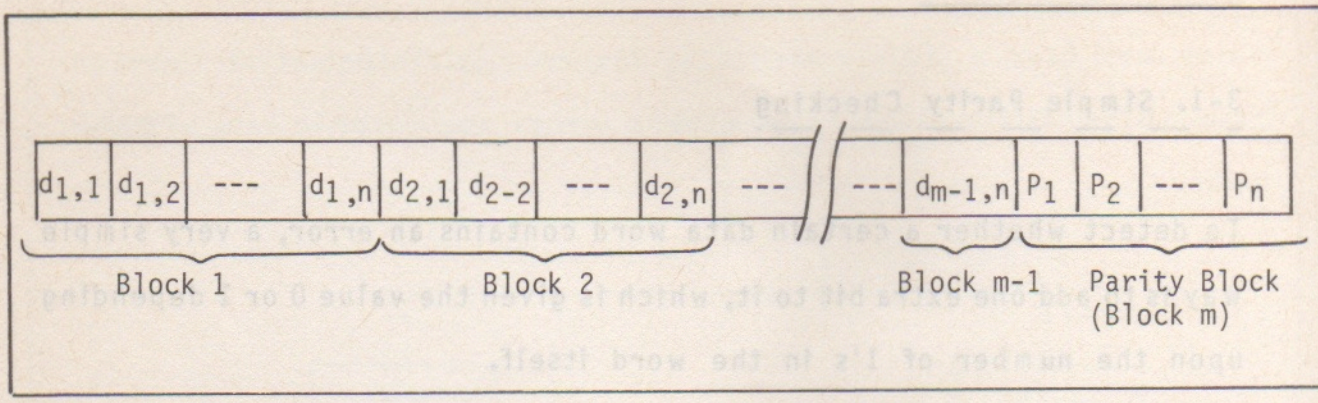
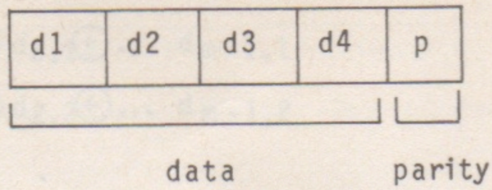


Fig. 2-4. Extended parity checking

General form :



$$p = d1 \oplus d2 \oplus d3 \oplus d4 \oplus a$$

(Even parity : a=0; odd parity : a=1)

($\oplus$ = symbol for modulo-2 addition)

When we now receive the incoming 'data + parity' blocks, we know that the sum of the '1's in the message must always be odd (or even). If this is not the case, we know that there has been a transmission error.

This rather elementary system has several obvious disadvantages :

- Even if we know there has been an error, we have no way to know which bit was faulty, so we cannot correct the message.
- If two bits of the same word are faulty, the errors may compensate each other and we do not detect any error.

3-2. Extended Parity Checking

To increase the probability of detecting errors, we can add more than one parity bits to each block of data.

If we consider the construction of Fig. 2-4., consisting of (m - 1) blocks of n data bits each. To these blocks, a mth block of parity bits, called the 'parity block', will be added by modulo-2 addition of all subsequent data bits of each block :

$$p1 = d_{1,1} \oplus d_{2,1} \oplus \dots d_{m-1,1}$$
$$p2 = d_{1,2} \oplus d_{2,2} \oplus \dots d_{m-1,2}$$

.

.

.

$$pn = d_{1,n} \oplus d_{2,n} \oplus \dots d_{m-1,n}$$

If the number of parity bits is n, it can be shown that (for a reasonably high value of n) the misdetection probability is $1/2^n$.

3-3. Cyclic Redundancy Check Code (CRCC)

There exists another checking scheme called Polynomial or Cyclic coding that can be designed to perform with higher efficiencies than traditional parities. The principles behind CRCC codes are as follows :

A very convenient way to express a bit stream (or word or message) of n bits is to think of it as an algebraic polynomial in a variable x with n terms.

For example, the word 10011011 may be written as follows :

$$M(x) = 1.x^7 + 0.x^6 + 0.x^5 + 1.x^4 + 1.x^3 + 0.x^2 + 1.x^1 + 1.x^0$$
$$= x^7 + x^4 + x^3 + x + 1$$

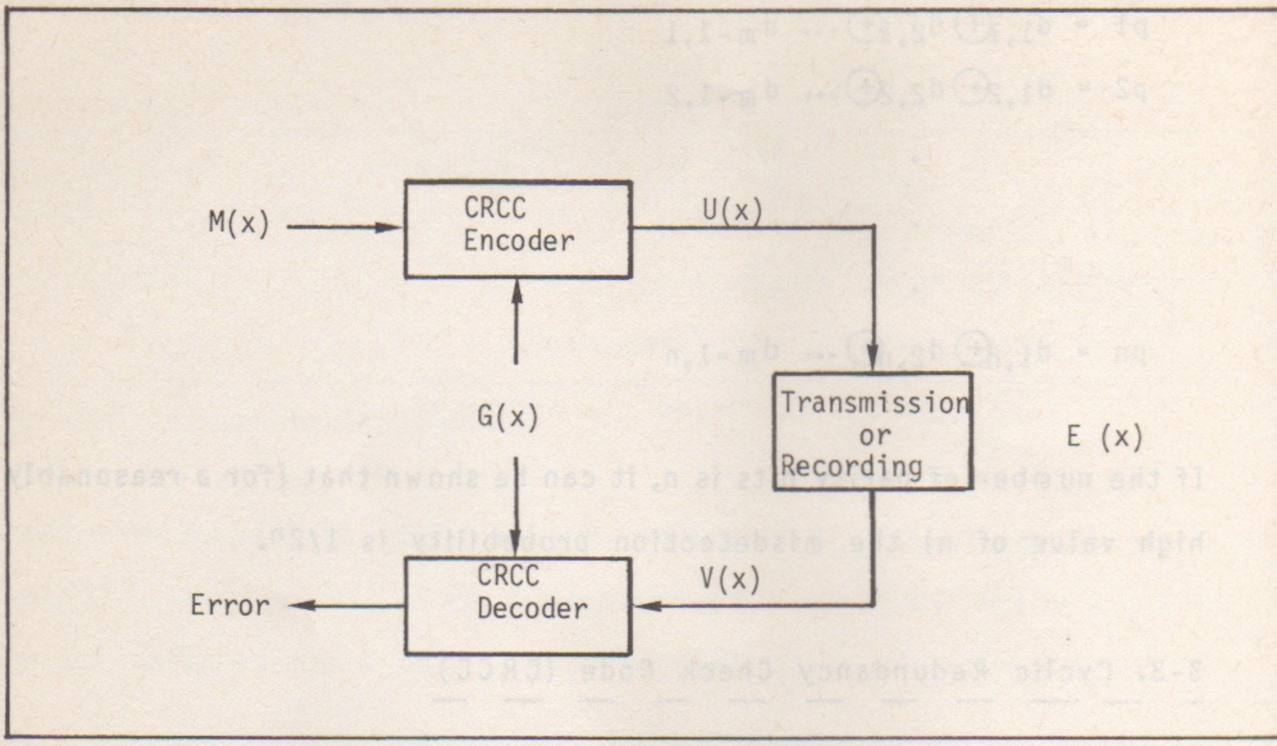


Fig. 2-5. CRCC Checking principle

Now to compute the cyclic check on M(x), another polynomial G(x) is chosen, with a degree r which is less than the degree of M(x), but greater than 0 of course; furthermore, the x^0 term must be 1.

Then, in the CRCC encoder, M(x) and G(x) can be divided

$$M(x)/G(x) = Q(x) + R(x)$$

In which Q(x) is the quotient of the division

R(x) is the rest.

Then, a new message U(x) can be generated as follows :

$$U(x) = M(x) + R(x)$$

This U(x) can always be divided by G(x).

It is this message U(x) that is recorded (or transmitted). When in the playback (or at the receiving end), an error E(x) occurs, we receive the message V(x) instead of U(x) :

$$V(x) = U(x) + E(x)$$

In the CRCC decoder, V(x) is again divided by G(x), and if the rest of the division is zero, we can decide that no error has occurred; if there is a rest, however, there has been an error.

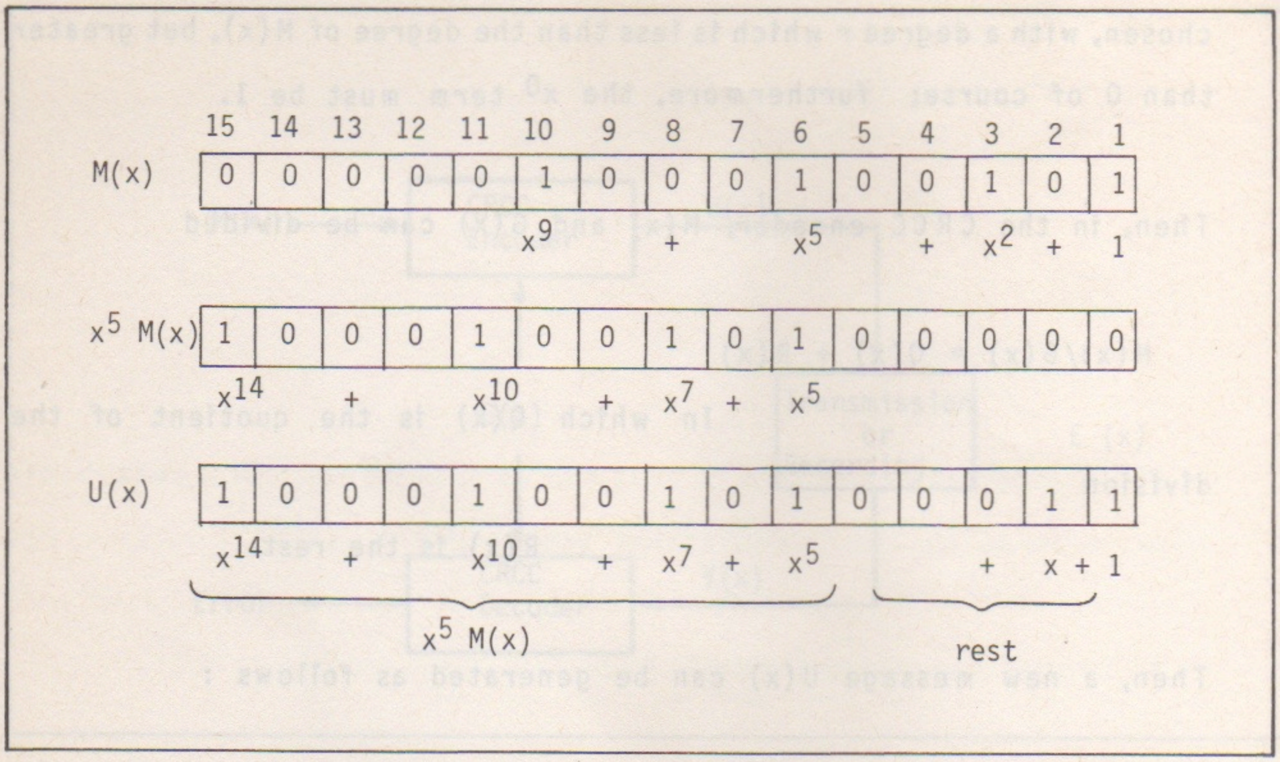


Fig. 2-6. Example of the generation of transmission polynomial U (x)

This process is schematically shown in Fig. 2-5.

Example :

The message is

$M(x) = x^9 + x^5 + w^2 + 1$

The Check polynomial is

$G(x) = x^5 + x^4 + x^2 + 1$

Now, before dividing by G(x), which is a 5th-order polynomial, we multiplicate M(x) by x^5; or in other words, we shift M(x) 5 places to te left. This is done in preparation of the five check bits that will be added to the message.

$x^5.M(x) = x^{14} + x^{10} + x^7 + x^5$

Then the division is made :

$$x^5.M(x) \;/\; G(x) = \underbrace{(x^9 + x^8 + x^7 + x^3 + x^2 + x + 1)}_{\text{quotient}} + \underbrace{(x + 1)}_{\text{rest}}$$

$U(x) = x^5.M(x) + (x + 1)$

$= x^{14} + w^{10} + x^7 + x^5 + x + 1$ which can be divided by G(x)

Fig. 2-6. shows the bit patterns for these examples. It can be seen that in fact, the data are unmodified (only shifted), and that the check bits follow at the end.

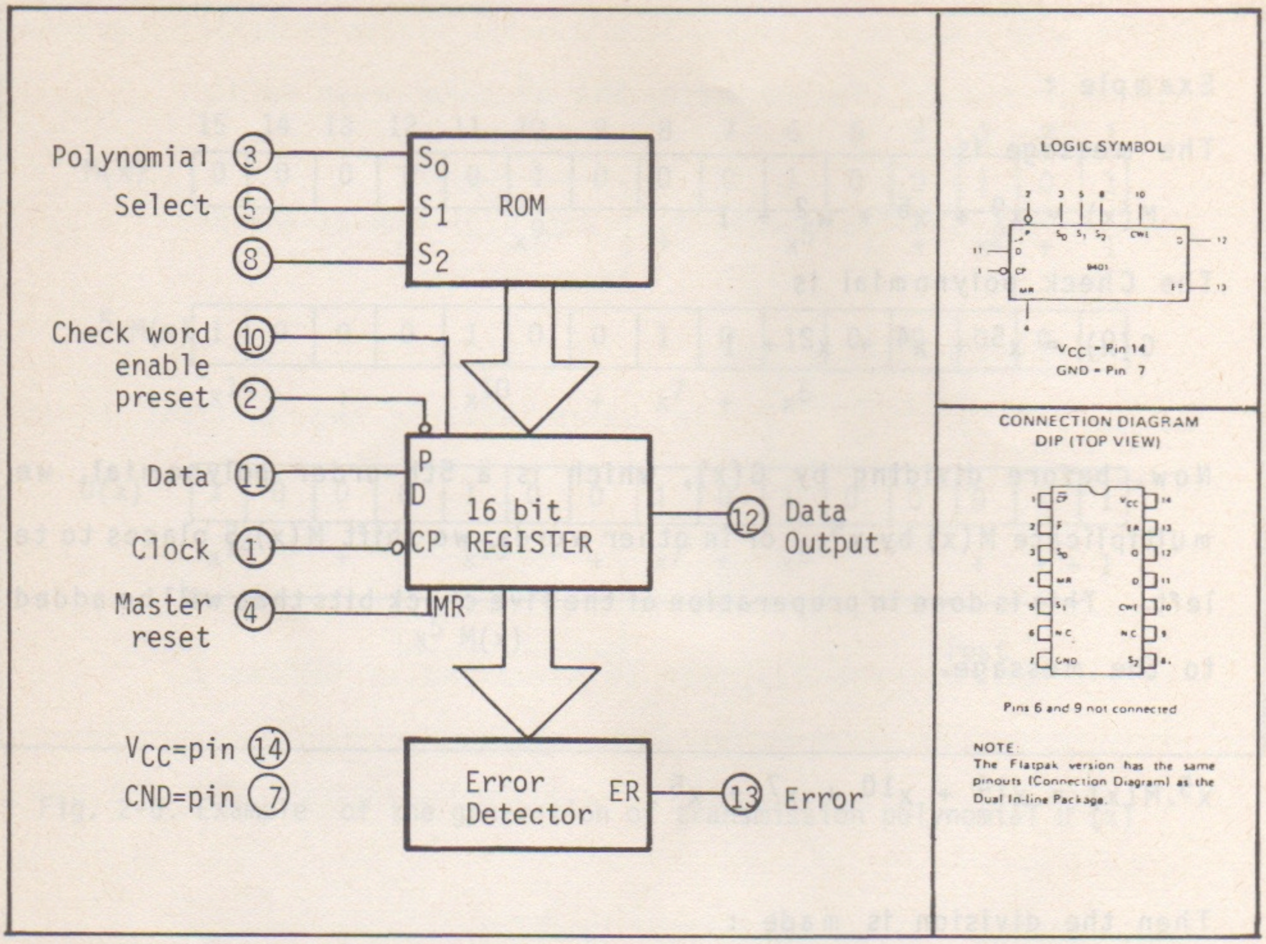


Fig. 2-7. Block diagram and layouts of Fairchild 9401 CRC checker

CRCC checking is very effective to detect any transmission errors; only if E(x) by coincidence is exactly dividable by G(x), no error will be detected. This obviously will occur only seldom, and, knowing the characteristics of the transmission (or storage) medium, polynomial G(x) can be chosen such, that this possibility is further minimized.

Although this error-checking system seems rather far-sought, involving complicated arithmetic operations, in fact the division can be done relatively simply using modulo-2 arithmetic, or in other words using exclusive OR's. There exist complete 1-chip CRCC generator/checkers, such as the Fairchild 9401; these i.c.'s generate some selectable standard polynomials, agreed by such organizations as CCITT, and automatically add the necessary check bits to the data. When used as a checker, data and check bits are entered and verified, and an Error signal generated whenever necessary.

Fig. 2-7. gives the lay-out and the Block Diagram of the Fairchild 9401.

4. Error Concealment Methods

When errors are too big to allow correction of the data by the Error Correction Code (see further), 'Error Concealment' must be adopted. Error Concealment is a way of reconstructing the samples, of which the data have been irrepairably lost, in such a way that the audible effect is minimized.

The following concealment methods are theoretically possible :

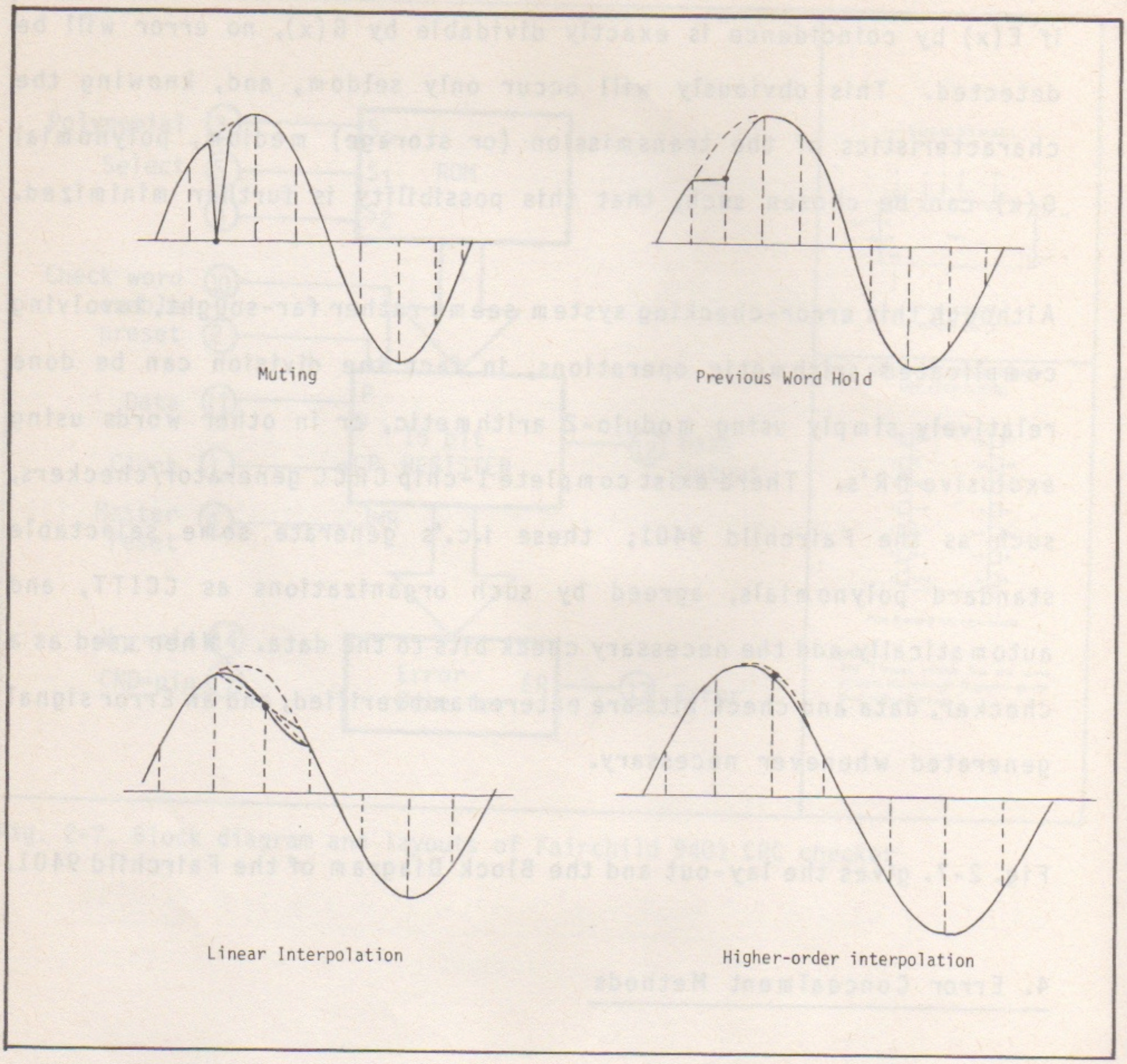


Fig. 2-8. Examples of error concealment

4-1. Muting.

When an error is detected, the signal is shortly muted; this system is hardly better than doing nothing, and is consequently not in use.

4-2. Previous word hold.

When word is detected as erroneous, it is replaced by the value of the previous sample. For low frequencies, this is an acceptable method; for high frequencies, however, it may give quite audible disturbances in the signal, which is understandable since in the extreme case the sampling frequency is only twice the signal frequency !

4-3. Linear Interpolation.

Also called averaging; when a word is missed, it is given the mean value of the preceding and following words. This is a very satisfactory method which gives almost inaudible concealment in most cases.

4-4. Higher-order polynomial interpolation.

Gives a better estimation of the missing sample by mathematically taking into account more preceding and following words. Although much more complicated than the foregoing system, it may be worthwile to use it in very critical applications.

5. Error Correction

5-1. Introduction

As we have seen before, when errors are detected, concealment is possible; concealment only, however, is not so satisfactory, especially at higher frequencies. Therefore, we must correct the majority of errors instead. When errors are to be corrected, we must not only know that an error has occurred, but also exactly which bit or bits are wrong. Since there exist only two possible states (0 or 1), correction is then just a matter of reversing the state of the erroneous bits.

Basically, correction (and detection) of code errors can be achieved by adding to the data bits an additional number of check bits, that are known as 'redundant' information. This redundancy information has a certain connection with the actual data, so that, when data get lost, they can be reconstructed again from the 'redundancy'. The "redundant" information is known as the error correction code. As a basic example, all data could be transmitted (and for instance recorded) twice (Double Writing Method), which would give a redundancy of 100%. By comparing both versions or by CRCC, errors could easily be detected, and if some word were erroneous, its counterpart could be used to obtain the correct data. It could even be considered to record everything three times, and upon playback select two-out-of-three, if there were errors; this would be still more secure. These are however rather wasteful systems, and much more efficient error correction systems can be constructed.

Presently, the development of strong and efficient Error Correction Codes is one of the key points of Digital Audio Technology, which differentiate between the advanced companies and the followers !

A lot of experience can be used from computer technology, where the correction of code errors is equally important, and where a lot of research has been spent in order to optimize correction capabilities of error correction codes. Designing strong codes is a very complex matter, however, which requires thorough study and use of higher mathematics. (Algebraic structure has been the basis of the most important codes.)

Some codes are very strong against 'burst' errors, i.e. when entire clusters of bits are erroneous together (such as during tape drop-outs), whereas others are better against 'random' errors, i.e. when single bits are faulty.

Error Correction Codes can be organized in two ways :

- Data bits and error correction bits can be organized in blocks; in this case we talk about 'Block Codes'. Redundancy that follows a block is only generated by the data in that particular block.
- Data and error correction may be mixed in one continuous data stream; in this case we talk about 'Convolutional Codes'. Redundancy within a certain time unit does not only depend upon the data in that same time unit, but also upon data occurring a certain time before. They are more complicated, and often superior in performance to block codes.

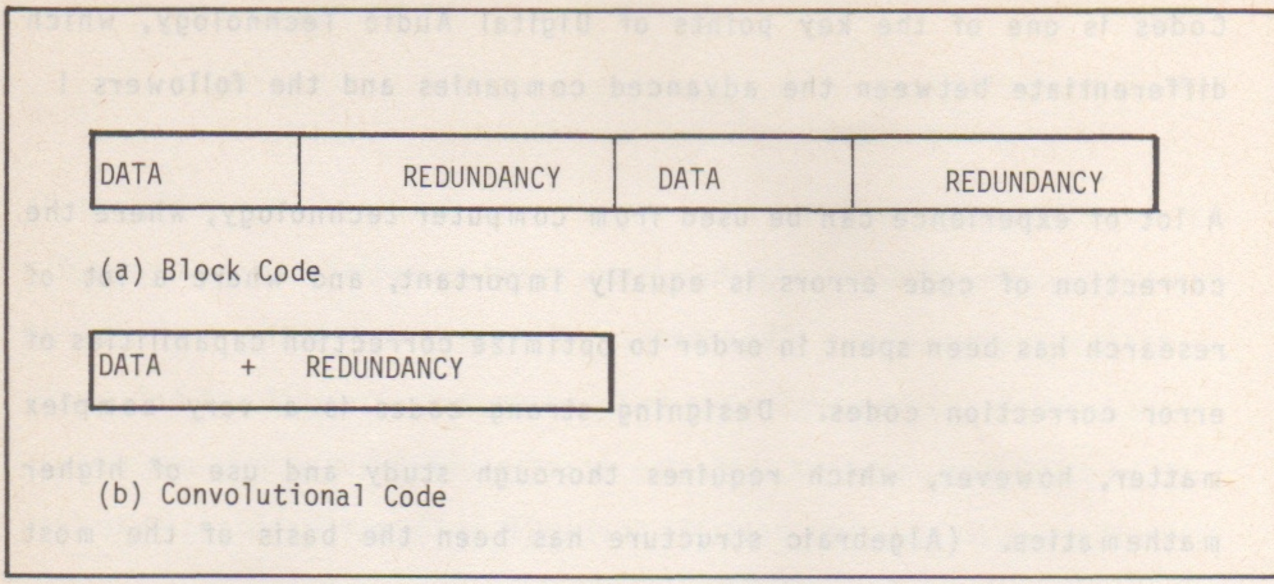


Fig. 2-9. Difference between block and convolutional codes

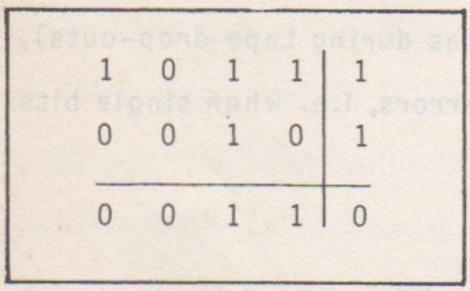


Fig. 2-10. Example of combinational parity checking

5-2. Combinational (Horizontal/Vertical) Parity Checking

If, for example, we consider a binary word or message consisting of 12 bits, these bits could be arranged in a 3 x 4 matrix as shown in Fig. 2-10.

Then, to each row or column one more bit can be added to make parity for that row or column even (or odd). Then, in the lower right-hand corner, a final bit can be added that will give the last column an even parity as well; it will then show that also the last row will have even parity.

If now this entire array is transmitted in sequence (row by row or column by column), and if during transmission one bit is changed, parity check on one row and on one column will fail, and the error will be found at the intersection; consequently, it can be corrected.

The entire array, in the example consisting of 20 bits, of which 12 are data bits, form a code word, which is referred to as a (20,12) code. There are 20 - 12 = 8 redundant digits.

All the so-called Error-correcting codes are more or less based on the same idea, although by studying them better in a mathematical way, much better codes than the one given in our example can be constructed. For example, it is possible to correct all single errors in the same 12 bits as we have shown with fewer redundant bits than 8, for instance by using so- called Hamming codes.

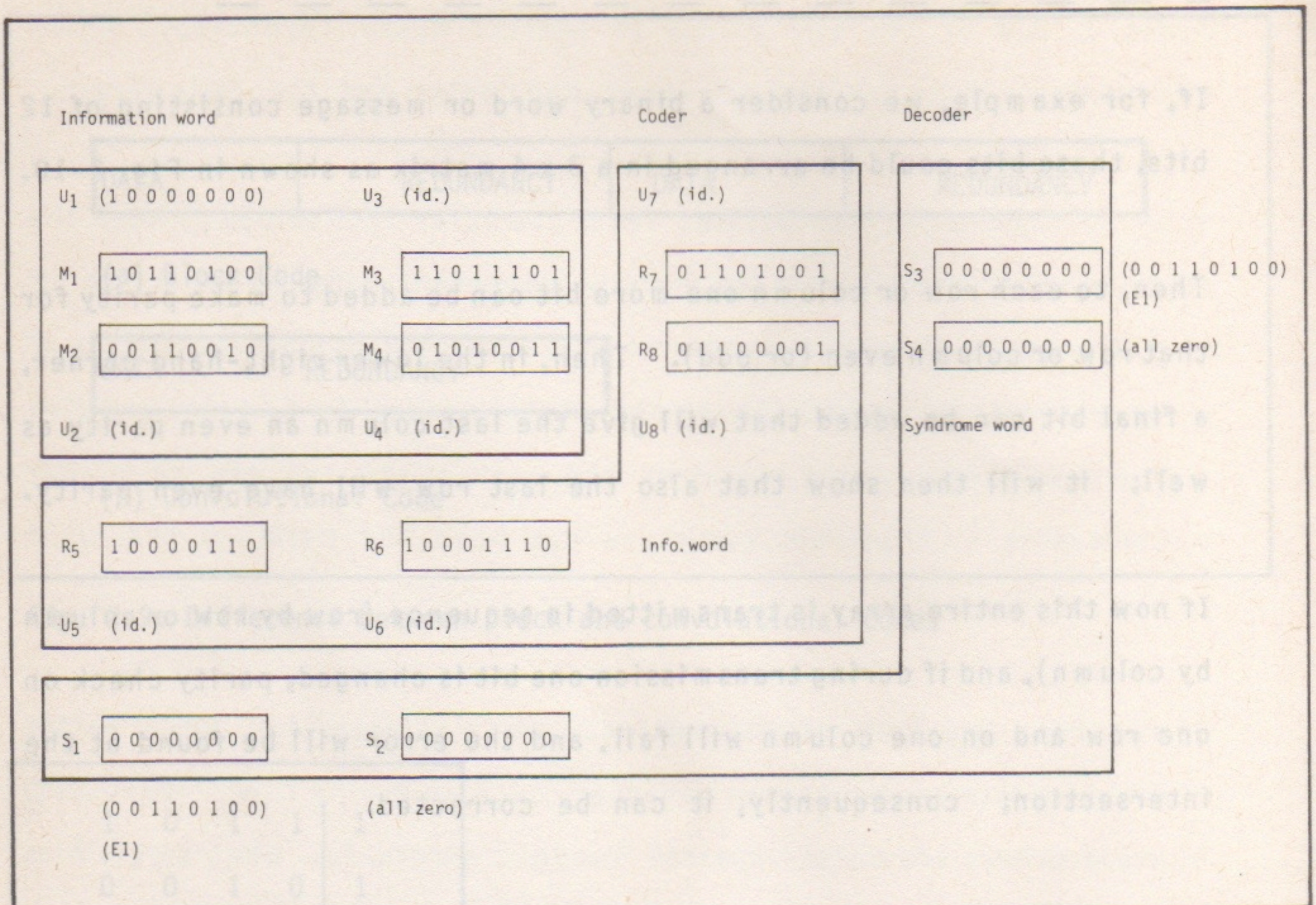


Fig. 2-11. Example of simple crossword code
Between brackets the situation of M_1, received as a U_1 with an error.

5-3. Crossword Code

In digital recording, not only single bits may become erroneous, caused by Intersymbol Interference, jitter, etc., but very often as already mentioned errors will come in bursts, i.e. complete clusters of bits may be faulty due to tape drop- outs. It will be obvious that, in view of the many possible combinations, the correction of such bursts or errors is very complicated and demands powerful error-correcting schemes.

One such a code, which has been developed by Sony for use in its PCM-1600, is Crossword Code.

Crossword Code uses a matrix scheme as basically explained above, but it carries out its operations with words instead of bits as units. This gives as an advantage that large block code constructions can be easily realized, that burst error correction capability is very high, and that also random errors can be very well corrected. Basically, it can be said that all types of errors with a high probability can be detected and often corrected, and that only errors with a low probability of occurring may pass undetected.

Fig. 2-11. shows a simple example of Crossword Code, in which 4 words $M_1 \sim M_4$ are complemented by 4 parity or survey words $R_5 \sim R_8$, so that :

$$R_5 = M_1 \oplus M_2$$

$$R_6 = M_3 \oplus M_4$$

$$R_7 = M_1 \oplus M_3$$

$$R_8 = M_2 \oplus M_4$$

$M_1 \sim R_8$ are then recorded, and at playback received as $U_1 \sim U_8$.

Now in the decoder, additional words can be constructed, called the Syndromes, as follows :

$$S_1 = U_1 \oplus U_2 \oplus U_5$$
$$S_2 = U_3 \oplus U_4 \oplus U_6$$
$$S_3 = U_1 \oplus U_3 \oplus U_7$$
$$S_4 = U_2 \oplus U_4 \oplus U_8$$

By virtue of this procedure, we can show that if all received words U_1 ~ U_8 are correct, all syndromes must be zero.

If an error E occurs in one or more words, we can say that :

$$U_i = M_i \oplus E_i \quad \text{for } i = 1 - 4$$
$$U_i = R_i \oplus E_i \quad \text{for } i = 5 - 8$$

Now since

$$S_i = U_1 \oplus U_2 \oplus U_5$$
$$= M_1 \oplus E_1 \oplus M_2 \oplus E_2 \oplus R_5 \oplus E_5$$
$$= E_1 \oplus E_2 \oplus E_5$$

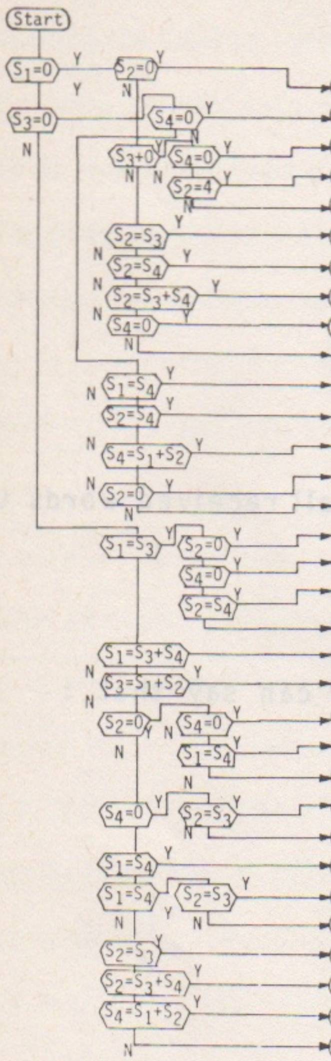
(since we know that $M_1 \oplus M_2 \oplus R_5 = 0$)

Similarly

$$S_2 = E_3 \oplus E_4 \oplus E_6$$
$$S_3 = E_1 \oplus E_3 \oplus E_7$$
$$S_4 = E_2 \oplus E_4 \oplus E_8$$

In our example, correction can be made by the following calculation

$$U_1 \oplus S1 \text{ or}$$
$$U_1 \oplus S_3 = M_1 \oplus E_1 \oplus E_1 = M_1$$



No.	Indication for decoding U_1	U_2	U_3	U_4	Cases of errors vs number of error words 0	1	2	3	4	5	6	7	8	Probability (p:word error rate)
(1)	0	0	0	0	1	2	1							$(1-P)^2+2p(1-p)^7+p^2(1-p)^6$
(2)	0	0	0	0		1	1							$p(1-p)^7+p^2(1-p)^6$
(3)	0	0	0	0		1								$p(1-p)^7$
(4)	0	0	0	1		1								$p(1-p)^7$
(5)	0	0	0	X			3	1						$3p^2(1-p)^6+p^3(1-p)^5$
(6)	0	0	1	0		1	1							$p(1-p)^7+p^2(1-p)^6$
(7)	0	0	0	1		1								$p^2(1-p)^6$
(8)	0	0	2	2		1								$p^2(1-p)^6$
(9)	0	0	X	0			3	1						$3p^2(1-p)^6+p^3(1-p)^5$
(10)	0	0	X	X				8	5	1				$8p^3(1-p)^5+5p^4(1-p)^4(1-p)^4+p^5(1-p)^3$
(11)	0	1	0	0		1	1							$p(1-p)^7+p^2(1-p)^6$
(12)	0	0	0	1			1							$p^2(1-p)^6$
(13)	0	1	0	1			1							$p^2(1-p)^6$
(14)	0	X	0	0			3	1						$3p^2(1-p)^6+p^3(1-p)^5$
(15)	0	X	0	X				8	5	1				$8p^3(1-p)^5+5p^4(1-p)^4+p^5(1-p)^3$
(16)	1	0	0	0		1	1							$p(1-p)^7+p^2(1-p)^6$
(17)	1	0	0	0			1							$p^2(1-p)^6$
(18)	1	0	0	1			1							$p^2(1-p)^6$
(19)	1	0	0	X				3	1					$3p^3(1-p)^5+p^4(1-p)^4$
(20)	2	2	0	0			1	1						$p^2(1-p)^6+p^3(1-p)^5$
(21)	1	0	1	0			1	1						$p^2(1-p)^6+p^3(1-p)^5$
(22)	X	0	0	0			3	1						$3p^2(1-p)^6=P^3(1-p)^5$
(23)	0	1	0	0			1							$p^2(1-p)^6$
(24)	X	X	0	0				8	5	1				$8p^3(1-p)^5+5p^4(1-p)^4+p^5(1-p)^3$
(25)	0	0	1	0			1							$p^2(1-p)^6$
(26)	X	0	X	0				8	5	1				$8p^3(1-p)^5+5p^4(1-P)^4+p^5(1-p)^3$
(27)	X	0	0	1				3	1					$3p^3(1-p)^5+p^4(1-p)^4$
(28)	0	1	1	0			1							$p^2(1-p)^6$
(29)	0	1	X	0				3	1					$3p^3(1-p)^5+p^4(1-p)^4$
(30)	0	X	1	0				3	1					$3p^3(1-p)^5+p^4(1-p)^4$
(31)	0	0	2	2				1						$p^3(1-p)^5$
(32)	0	1	0	1				1						$p^3(1-p)^5$
(33)	X	X	X	X				4	46	52	28	8	1	$4p^3(1-p)^5+46p^4(1-p)^4+52p^5(1-p)^3+28p^6(1-p)^2+8p^7(1-p)+p^8$
Total 8 C_i					1	8	28	56	70	56	28	8	1	

Fig. 2-12. Decoding algorithm for crossword code

Of course, there is still a possibility that simultaneous errors in all words compensate each other to give the same syndrome patterns as in our example. The probability of this occurring, however, is extremely low, and can be disregarded

In practical decoding now, when errors occur, the combination of the values of the syndromes are investigated, and a decision is made whether there is a good probability of successful correction. If the answer is yes, correction is carried out; if the answer is no, a concealment method will be used (either average value interpolation or previous word hold). The algorithm along which this decision-making occurs must be decided via probability calculations, but once it is fixed, it can easily be implemented, for instance in a P-ROM.

Fig. 2-12. shows the decoding algorithm for the Crossword Code we discussed. As will be seen, depending upon the value of the syndrome(s), decisions are made for correction according to the probability of miscorrection; the right column shows the probability for each situation to occur.

The example does not show a code which is actually used in a recorder such as PCM-1600; the practical system will be discussed later on when the recorder itself is described.

5-4. b-Adjacent Code

A different code which is very useful for correcting random and burst errors at the same time has been described by D.C. Bossen of IBM, and called b-Adjacent code. The b-Adjacent error correction system is used in the EIAJ-format for home-use helical-scan digital audio recorders.

In this format, two Parity words, called P and Q are constructed as follows:

$$P_n = L_n \oplus R_n \oplus L_{n+1} \oplus R_{n+1} \oplus L_{n+2} \oplus R_{n+2}$$

$$Q_n = T^6 . L_n \oplus T^5 . R_n \oplus T^4 . L_{n+1} \oplus T^3 . R_{n+1} \oplus T^2 . L_{n+2} \oplus T . R_{n+2}$$

in which T is a specific matrix of 14 words of 14 bits. L_n, R_n etc. are data words from respectively the left and the right channel. (We neglect the interleaving for simplicity.)

In addition, CRC is used as error pointer.

The coding scheme of the EIAJ-format leaves possibility for combining several decoding systems, according to the cost and desired reliability of the system.

5-5. Other codes

Many other codes exist since almost each manufacturer of professional audio equipment has designed his own preferred error correction system. Most of them, however, are variations on the best known codes, that have names such as Reed-Solomon code, BCH Code (Bose-Chaudhuri-Hocquenghem), etc.

Sony is now proposing codes called Cross Interleave Code for stationary-head recorders and Cross Interleave Reed Solomon Code (CIRC) for the Compact Audio Disc System (developed together with Philips).

Also in the field of error correction, standardization between manufacturers will be very important to allow world-wide distribution of recordings.

5-6. Interleaving

In magnetic tape recording, in view of the high recording density used to record digital audio, dropouts will frequently destroy many subsequent words all together

If this were allowed to happen, error correction that could cope with such situation would be prohibitively complicated, and, if it failed, concealment would not be possible since methods like interpolation demand that only one sample of a series is wrong.

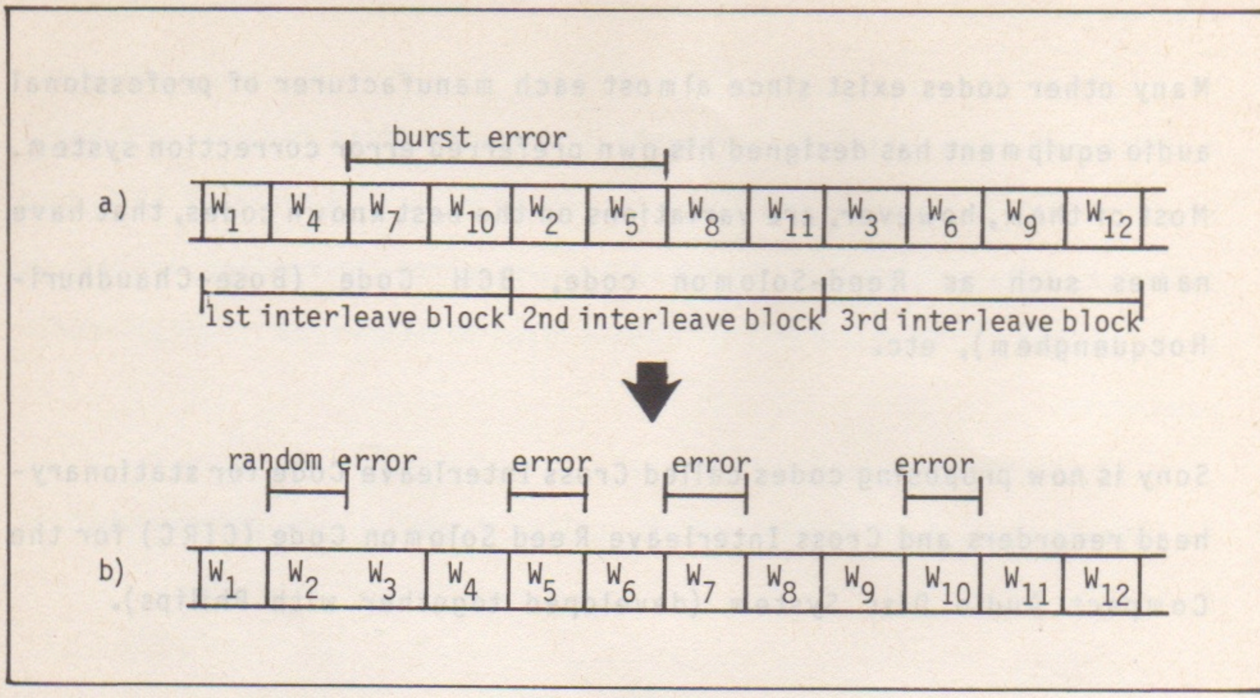


Fig. 2-13. Simple example of interleaving

(a) Interleaving signals on tape; a burst error destroys four subsequent words.

(b) Same signals after de-interleaving: errors are now limited to random words and correction and/or concealment is simplified.

For that reason, subsequent words from the A/D convertor are not written next to each other on the tape, but at locations, a sufficient distance apart to make sure that they cannot suffer from the same dropout.

In fact, this consequently is a way to convert long burst errors into a series of random errors. Fig. 2-13 demonstrates this in a simplified example. It can be seen that words are arranged in ´interleave blocks´ (which are as long as the distance between two subsequent words).

Practical interleaving blocks will be much more complicated than our example, as we will see when discussing different formats.

6. Recording formats in actual use

6-1. The PCM-100 & PCM-F1 format

1) This format uses the EIAJ-standard which is summarized in the following table.

Item	Specification
Number of channels	2 (CH-1 = left, CH-2 = right)
Number of bits	14 bits/word (pro channel)
Quantization	linear
Digital code	2's complement
Modulation	NRZ (Non-Return to Zero)
Sampling frequency	44,056 kHz
Bit Transmission rate	2,634 Mbit/sec
Video Signal	NTSC-standard

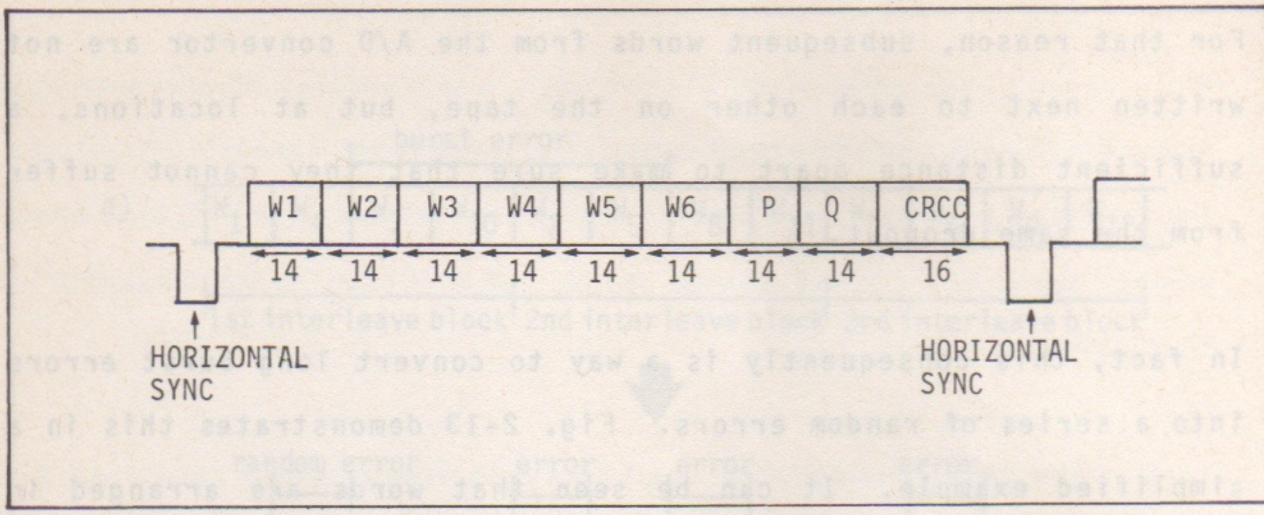

W1
W2
W3
W4
W5
W6
P
Q
CRCC
14
14
14
14
14
14
14
14
16
HORIZONTAL
SYNC
HORIZONTAL
SYNC

Fig. 2-14.

2) A horizontal pseudo-video line stores 128 bits as shown in figure 2-14.

The words W1 to W6 are INFORMATION words (14 bit each).

The words W1, W3 and W5 are used for the digital audio-data of the left channel.

The words W2, W4 and W6 are used for the digital audio-data of the right channel.

The words P and Q are ERROR CORRECTION words (14 bits each)

They are calculated as follows:

$$P = W1 \oplus W2 \oplus W3 \oplus W4 \oplus W5 \oplus W6$$

($\oplus$ denotes the EXCLUSIVE OR - function or modulo - 2 addition)

$$Q = T^6W1 \oplus T^5W2 \oplus T^4W3 \oplus T^3W4 \oplus T^2W5 \oplus TW6$$

(T is the "Q generation matrix" and uses the polynomial $X^{14} \oplus X^8 \oplus 1$)

The CRCC word finally is a 16 bit ERROR DETECTION word.

(It is generated by the polynomial $X^{16} \oplus X^{12} \oplus X^5 \oplus 1$)

3) The REDUNDANCY of this format is as follows:

We have 6 x 14 = 84 audio data bits and

(2 x 14) + 16 = 44 error correction & detection bits

$$R = \frac{44}{44 + 84} = \frac{44}{128} = 34{,}4\%$$

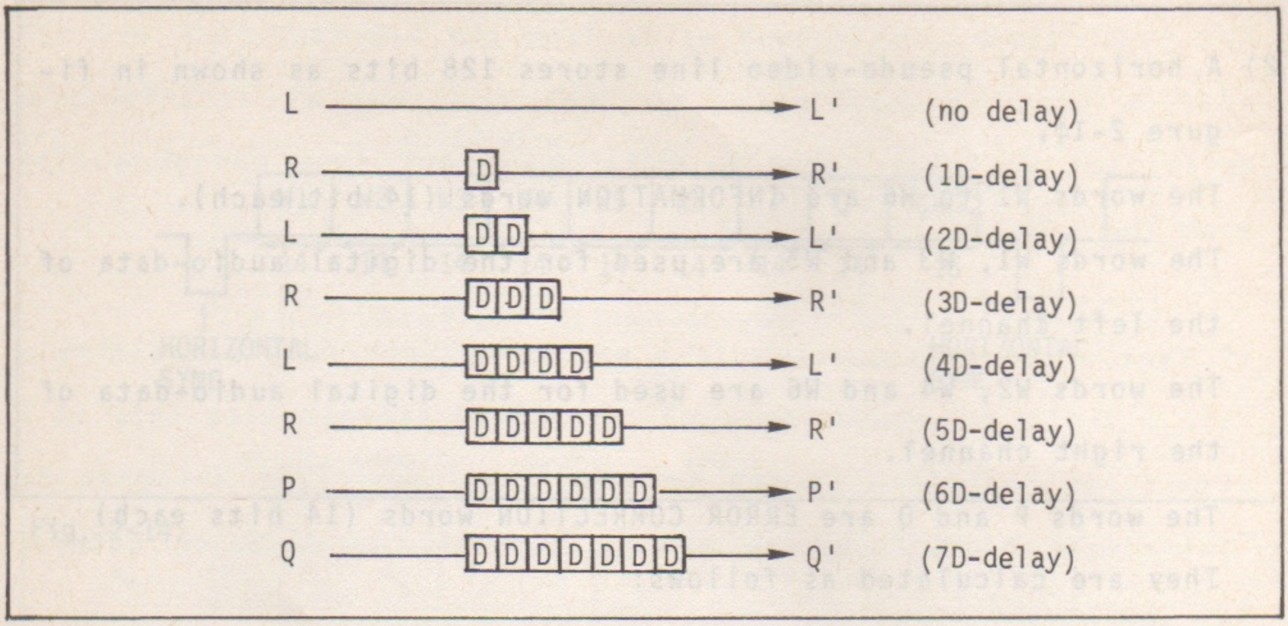


Fig. 2-15.

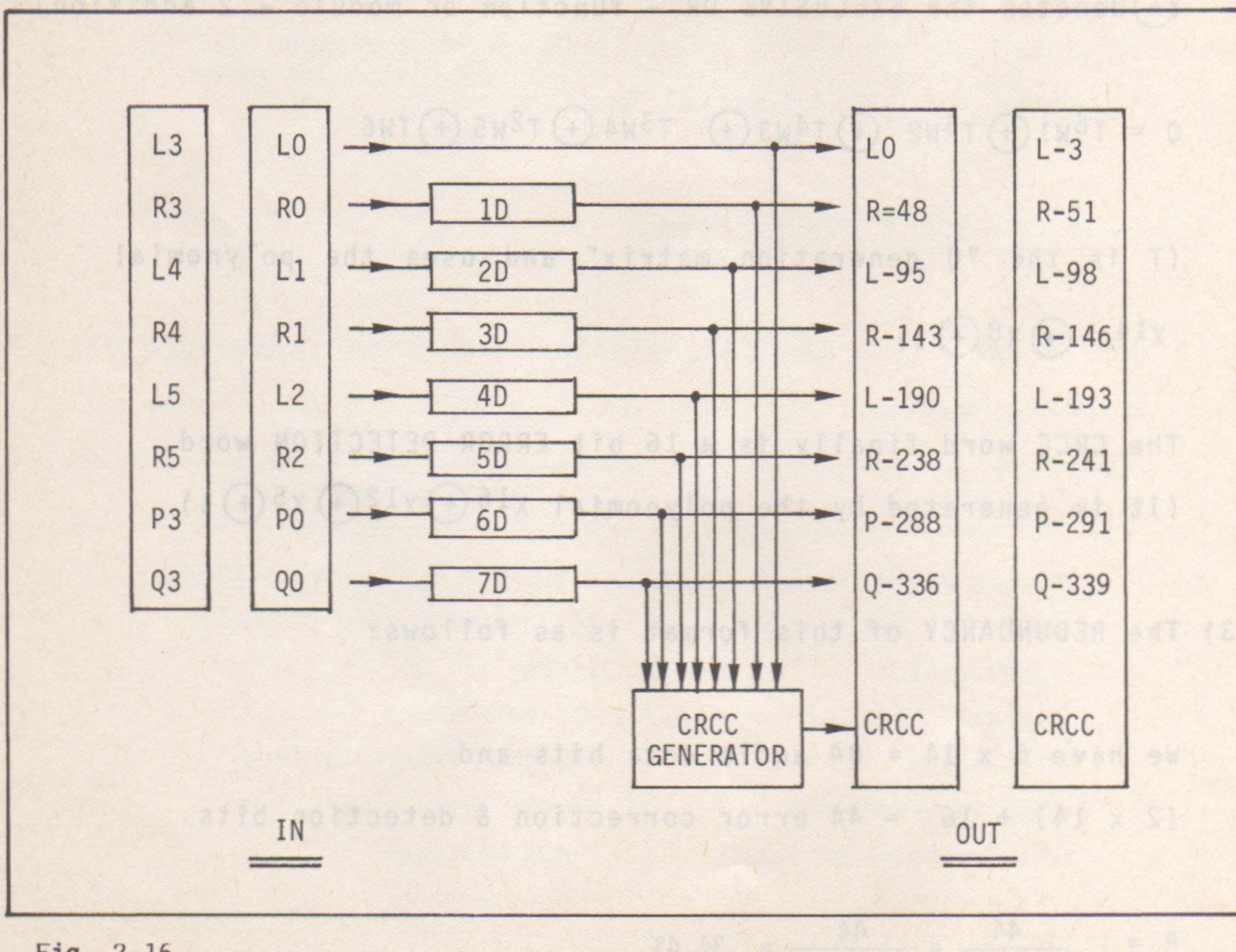


Fig. 2-16.

4) The INTERLEAVING process is as shown in figure 2-15.
"D" is the INTERLEAVE DELAY" and equals 16 words.

Figure 2-16. shows the interleaving in more detail.

6-2. The PCM-1600/-1610 format

1) The following table summarizes the main points of this format.

Item	Specification
Number of channels	2 (CH-1 = left, CH-2 = right)
Number of bits	16 bits/word (pro channel)
Quantization	linear
Digital Code	2's complement
Sampling frequencies	44,056 kHz or 44,1 kHz
Bit transmission rate	3,5795 Mbit/sec or 3,5831 Mbit/sec
Video signal	NTSC - standard

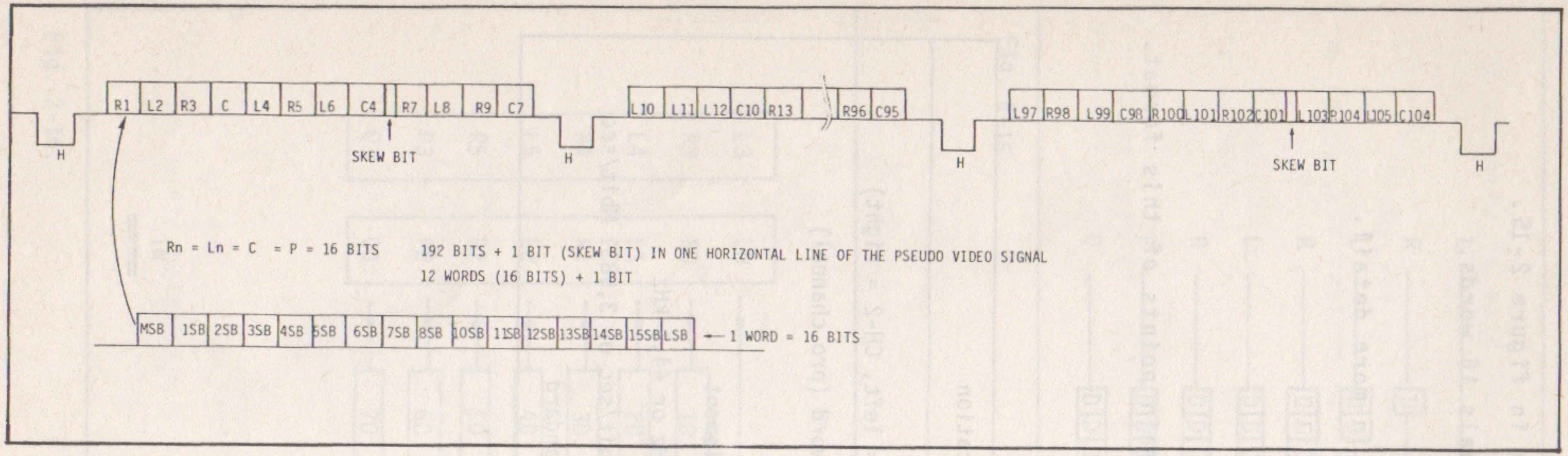


Fig. 2-17.

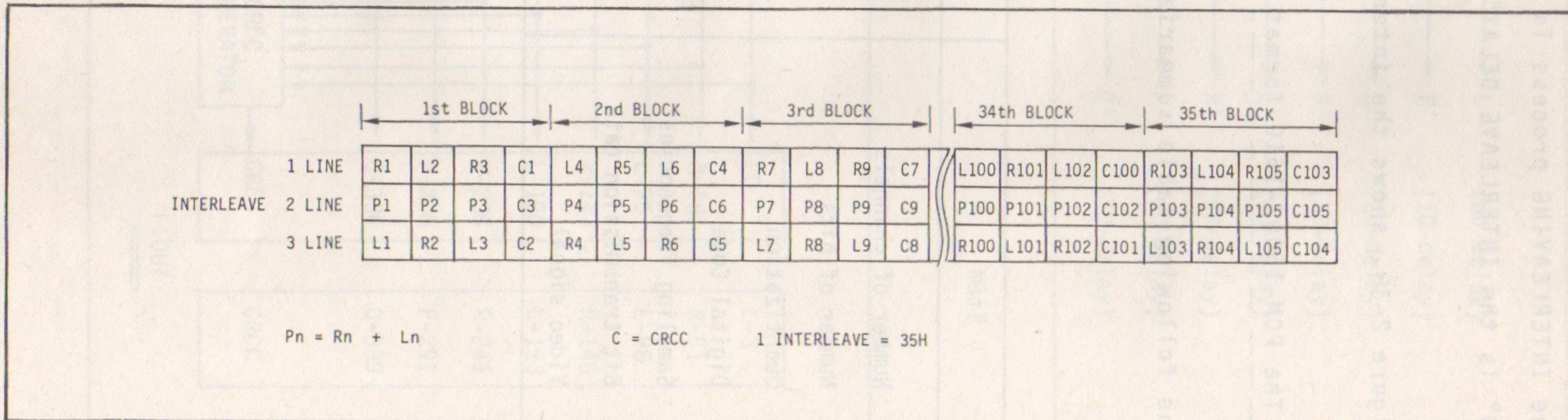


Fig. 2-18.

2) A horizontal pseudo-video line stores 193 bits as shown in figure 2-17.

3) Error-correction and - detection words are added as shown in the table below.
A block of 6 audio-data words (3 left-channel words and 3 right-channel words) is linked with 3 PARITY words and 3 CRCC words.

R1	L2	R3	C1
P1	P2	P3	C3
L1	R2	L3	C2

The PARITY word Pn is generated by the exclusive function of Ln and Rn

$Pn = Ln \oplus Rn$

The CRCC words Cn are generated by the polynomial

$X^{16} \oplus X^{12} \oplus X^{5} \oplus 1$

4) Figure 2-18. shows the INTERLEAVING of the data.
The 105 L-data-words, the 105 R-data-words, the 105 P-data-words and the 105 CRCC-words make 1 interleave.
These 420 words are stored in 35 horizontal lines. A video-field stores 7 interleaves (or 245 data-lines).

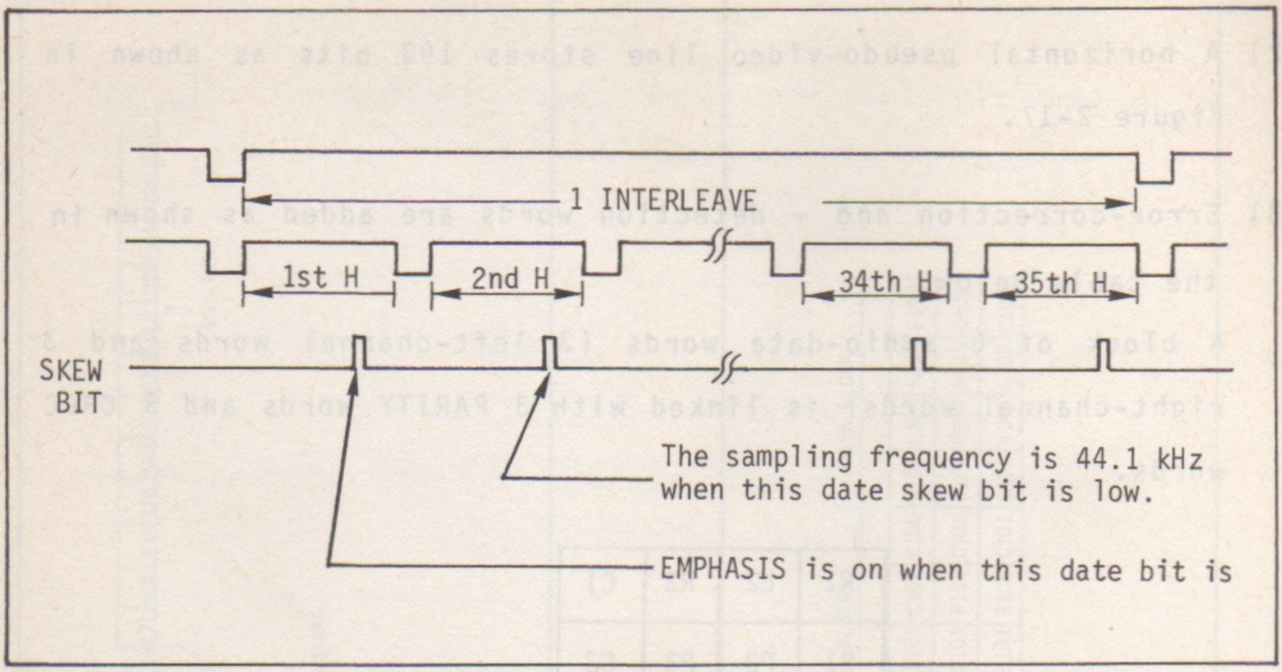

1 INTERLEAVE
1st H
2nd H
34th H
35th H
SKEW BIT
The sampling frequency is 44.1 kHz when this date skew bit is low.
EMPHASIS is on when this date bit is

Fig. 2-19.

5) The REDUNDANCY of this format is 50%, as a basic block (12 words) is composed of 6 audio-data words (3 X L, 3 X R) and 6 error-correction and -detection words (3 X P, 3 X CRCC).

6) The 129th bit of each horizontal line is a SKEW-bit. The skew-bit of the 1st and 2nd horizontal lines can be set as shown in the table.

Skew-bit 1st H	Skew-bit 2nd H	Sampling frequency	Emphasis on/off
0	0	44.1	ON
0	1	44.056	ON
1	0	44.1	OFF
1	1	44.056	OFF

The skew-bits of the other horizontal lines are always "1" (high).

This is shown in figure 2-19.

SONY SERVICE CENTRE (Europe) N.V.
Halfstraat 80,
2621 SCHELLE (Antwerp)
Belgium

Printed in Belgium
P/N S-790-093-01