A General Format for Data Compression

Hank JAVAN, Mike FULTON
Department of Engineering Technology, University of Memphis
Memphis, Tennessee 38152

ABSTRACT

Transmission of huge amount of data, either analog or digital requires significant amount of bandwidth and processing time. Data companding can reduce the required bandwidth and the transmission time to an acceptable level. Analog companding have been fully developed and several companding methods such as u-Law and A-Law compression methods are now in commercial use [1]. They are basically log amplifiers. But it was not until 1970 with the event of personal computers that digital companding received special attention. Although several software have been developed but there seems to be no a general format for digital data compression.

This article addresses a new general method for companding any digital data by introducing a new compression format. Specifically a method will be introduced to convert a 16 bits data to 12 bits then to an eight and finally to 6 bits, thus reducing the bandwidth and transmission time by a factor of 16/6. The recovered data, depending on resolution and dynamic range of the sampled signal may have some inevitable error, which is the fundamental drawback of every compression method. However, it will be shown mathematically that suggested method forces this error to attain a minimum possible value not to exceed allowed resolution.

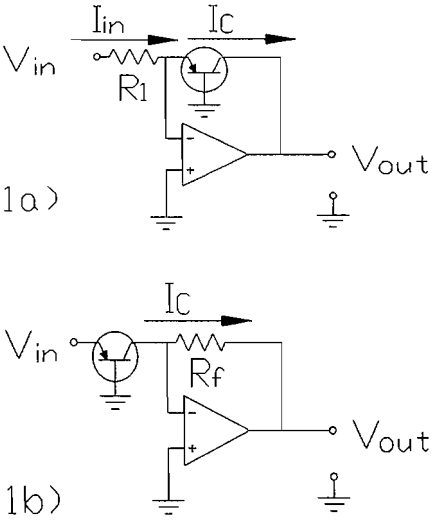
Keywords: Analog/ Digital Converter, Sample/Hold Circuit, Data Compression.

1. INTRODUCTION

Signal compression is an old method scientists used to analyze huge data or several signals. Use of log paper to plot frequency response and slide rule to compute mathematical operation (those don't know what it is should check our display) all are examples of data compression. Use of log paper makes it possible to scale x-axis from 1Hz to 100 MHz, otherwise kilometers of paper will be required to obtain few decades of frequency response.

2. ANALOG COMPRESSION

As mentioned earlier analog compression is based on logarithmic operation. Log operation not only compresses data but also convert multiplication to addition. Two standard analog compression circuits used for analog signals are shown in figure 1. Figure 1(a) is used to compress the signal and 1(b) is the reverse process, i.e. it decompresses the signal thus obtaining original signal [2]. Analog signal compression can be very precise; i.e. the recovered signal can resemble the original signal without significant error.



(a) $V_o = -(0.025)\ln(V_{in}/I_{eb}R_1)$
(b) $V_o = -R_f I_{eb}\,\text{antilog}(V_{in}/25\text{mV})$

Fig.1-Compression of Analog Signal
(a) Log amplifier (b) Antilog Amplifier

3- DIGITAL DATA

Real data are in analog format. Thus they must be converted to code before digital compression. A basic circuit shown in figure 2 will accomplish this task [3]. Op/amp 1 and 2, a single chip such as IC 741, is used as buffer, and JFET is used as a switch to sample the analog signal, capacitor will charge and hold the sampled data, after which the priority encoder will code the data for transmission. JFETs provide high input impedance and low on resistance and high off resistance. Accuracy and also the resolution depend on the frequency of sampling pulses applied to the base of JFET. According to Nyquist theory minimum sampling rate is given by;

$$f_s > 2 f_{in} \qquad (1)$$

This sets the lower limit. In practice a sampling rate 10 times the analog input frequency will be sufficient to recover the original signal. Figure 3 shows the effect of sampling frequency. As can be seen from this figure, more sampling will require more bits/sample. For example if we have only 7 sample of a given signal, we must use 3 bit in order to code all 7 samples. Number of bit/sample depends on input voltage, sampling frequency, and required resolution is given by;

$$2^n - 1 = V \qquad (2)$$

Satellite communications are using digital transmission such as FSK (frequency shift keying), QPSK (quadrature phase shift keying) all use PCM (pulse code modulation) scheme with 8, 12, or 16 bits per sample. Thus it becomes important to compress this huge amount of data in order to conserve bandwidth and transmission time. In the following section a general method for compression
16 bits to 12 bits will be discussed, then 12/8 and 8/6 bit will be demonstrated. Thus a general format will be established.

Compression of 16 bits to 12 bits:
A 16 bit data excluding sign bit will have 2^{15} = 32768 possible voltage level. Hopefully with this huge amount of data every possible signal level can be detected and can be encoded. In the suggested method this data will be divided in to 12 sections. As long as we cover all possible levels, i.e. 32,768

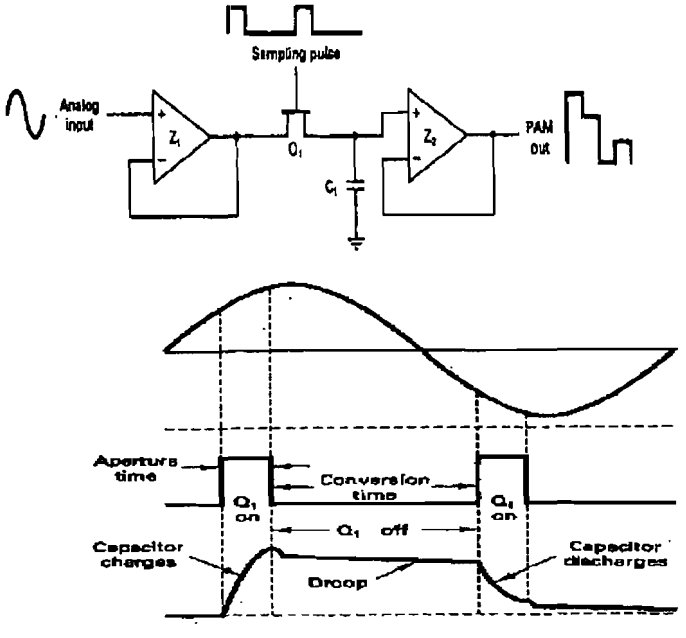


Fig.2-A basic sample/hold circuit

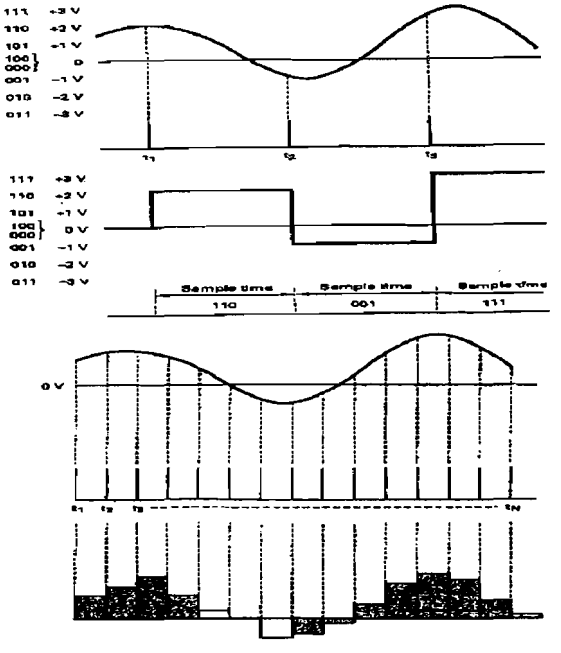


Fig.3-Effect of sampling

levels, the method should be valid one. In addition, since compressing a data, naturally will introduce error 12 bits code will be identified with 11- zeros up to, we must demonstrate that recovered code will not differ from the original code significantly. Table 1 shows a 16 bits data, section numbers with number of levels in each section, and the compressed 12 bits data. Section number in the 12 bits code will be identified with 11-zeros up to 11. Since we have 12 sections, thus we must use 4 bit to identify each section. This gives us 16 possibilities, enough for 12 sections. The 12 bit compressed data will consist of 4 bits for the section and one bit for the sign, 7 bit for the code 4bit of which will be the original code with leading zeros corresponding to original code. Table 1 shows the general format. x means drop or don't care state, but in calculating the level they should be taken in to consideration. Thus segment three for example has a total of 2^6 combinations and section five a total of 2^8. In this format the sum of all combinations is 32,768 bits in agreement with the 2^{11}. Also it is to be noted that with this format the number of leading zeros i.e. the zeros in front of 1DCBA of compressed code should not exceed the number of zeros in the original code. For this reason segments 10 and 11 have one zero and no zero respectively in front of 1DCBA. The error in this format will involve the drop of x bits. In compensating for this error we insert a 1 for the most significant bit and assign a 0's for the least significant bits. The method, depending on the resolution and voltage level, still yields minimum error as will be shown.

Compression of 12 /8 and 8/6bits

Following above method format for companding a 12/8 and 8/6bits data are shown in table 2 and 3 [3,4]. The implementation of 12/8bits is shown at the end of this manuscript. Following section will be devoted to the calculations of errors introduced in recovering original codes.

Table 1- Compression of 16bits to 12bits

16bits code	Sec. ---------- 11-zeros	Level -------	12bits code ---------- Sec., code	16 bit recovered code
S, 11 z DCBA	0	2^4	S,0000,000DCBA	S, 11z, DCBA
,10 z 1DCBA	1	2^4	,0001,001DCBA	, 10z,1DCBA
, 9 z1DCBAx	2	2^5	,0010,001DCBA	, 9z,1DCBA0
, 8 z1DCBAxx	3	2^6	,0101,001DCBA	, 8z,1DCBA10
, 7 z1DCBAxxx	4	2^7	,0100,001DCBA	, 7z,1DCBA100
, 6 z1DCBAxxxx	5	2^8	,0101,001DCBA	, 6z,1DCBA1000
S, 5z1DCBAxxxxx	6	2^9	,0110,001DCBA	,5z,1DCBA10000
, 4z1DCBAxxxxxx	7	2^{10}	,0111,001DCBA	,4z,1DCBA100000
, 3z1DCBAxxxxxxx	8	2^{11}	,1000,001DCBA	,3z,1DCBA1000000
, 2z1DCBAxxxxxxxx	9	2^{12}	,1001,001DCBA	,2z,1DCBA10000000
, 1z1DCBAxxxxxxxxx	10	2^{13}	,1010,011DCBA	,1z,1DCBA100000000
,1DCBAxxxxxxxxx	11	2^{14}	,1011,111DCBA	, 1DCBA1000000000

Table 2 – Compression of 12 bits to 8 bits

12 bits	Sec. ---------- 7-zeros	Levels --------	8 bits code ------- ------- Sec, code	12 bits recovered code
S, 0000000DCBA	0	2^4	S,000,DCBA	S, 0000000DCBA
, 0000001DCBA	1	2^4	,001,DCBA	, 0000001DCBA
, 000001DCBAx	2	2^5	,010,DCBA	, 000001DCBA1
, 00001DCBAxx	3	2^6	,011,DCBA	, 00001DCBA10
, 0001DCBAxxx	4	2^7	,100,DCBA	, 0001DCBA100
, 001DCBAxxxx	5	2^8	,101,DCBA	, 001DCBA1000
, 01DCBAxxxxx	6	2^9	,110,DCBA	, 01DCBA10000
, 1DCBAxxxxxx	7	2^{10}	,111,DCBA	, 1DCBA100000

Table3 – Compression of 8 bits to 6 bits

8 bits	Sec. ---------- 5-zeros	Levels --------	6 bits code ------- ------- Sec, code	8 bits recovered code
S, 00000BA	0	2^2	S,000,BA	S, 00000BA
, 00001BA	1	2^2	,001,BA	, 00001BA
, 0001BAx	2	2^3	,010,BA	, 0001BA1
, 001BAxx	3	2^4	,011,BA	, 001BA10
, 01BAxxx	4	2^5	,100,BA	, 01BA100
, 1BAxxxx	5	2^6	,101,BA	, 1BA1000

4- Examples

16/12 Bits

Consider Companding a 16 bits data to 12bits with analog sampled voltage of +10.234, and 0.01V resolution. The procedure follows:

10.234/0.01=1023.4~(1023) = S,000001111111111
Sect. No = 11-5z = 6
12bits : =S,000001DCBAxxxxx
=S,0110,0011111
16bits recovered =S,00000,1111110000
Error: S000001111111111
-S000001111110000

1111=15X0.01=0.15V

This error, 0.15V in +10.234 is about 1.5% of original sampled voltage.

12/8 Bits

If we use the same voltage of +10.234 to encode using 12bits and compressing to 8bit the error will be as following:

10.234/0.01=1023.4~(1023)=S, 01111111111
Sect No. =7-1z=6
8bits: =S, 01DCBAxxxxx
=S,110,DCBA
12bits recovered = S,01DCBA10000
Error: S,01111111111
-S,01111110000

1111=15X0.01=0.15V

This error is the same as 16/12bit. Thus it is seen that for this sample of 10.234V with 0.01V resolution it is not necessary to use more than 12 bits for the original sample. The number of bits

required for sampling is determined using equation (2) depends on the range of the sample as well as the resolution. With regard to this last remarks it is clear that we must use more than 8bits to represent this sample. Thus in the next example we will use a different sample to apply for 8/6bits compression.

8/6Bits

Now Consider Companding an 8 bits data to 6bits with analog sampled voltage of +0.56, and 0.01V resolution. The procedure is similar to 16/12 or 12/8,

+0.56/0.01=(56) = S0111000
Sect. No. = 5-1 = 4
6bits: = S 01BAxxx
= S 10011
8bits recovered = S 0111100
Error: S 0111100
-S 0111000

100=4X0.01=0.04V

Note that suggested method yields extremely low error for low compression.

5. ACKNOWLEDGEMENT

Authors would like to thank all the students and friends especially Keng Tay, Mark Farrar, Charlie Hale, and Tallant Justin for helping with this project. Their expertise with computer software and their assistances has made this article more meaning full.

6. REFRENCES

[1]. Young, Paul H., **Electronic Communication Techniques"**, 5th ed. 2004, Prentice Hall.

[2]. Floyd, Thomas L., **Electronic Devices**, 5th ed, 1999, Prentice Hall.

[3]. Tomasi, Wayne, Electronic **Communications Systems**, 5th ed. 2004, Prentice Hall.

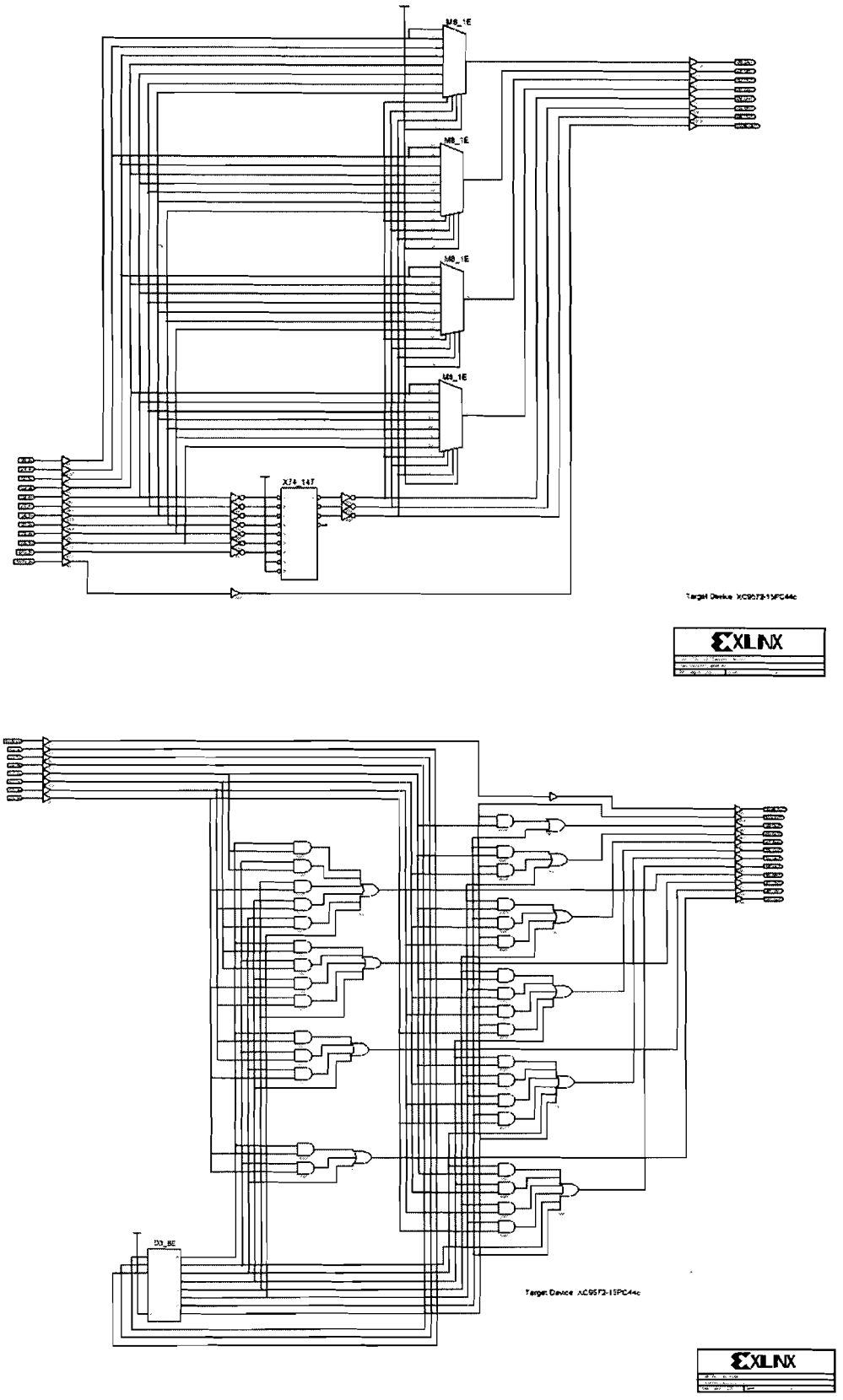


Fig.3-Implementation of 12/8bits