EDITED BY BILL TRAVIS & ANNE WATSON SWAGER

Circuit translates A law to μ law

ROLANDO HERRERO, INSTITUTO TECNOLÓGICO DE BUENOS AIRES, ARGENTINA

Two common methods exist to compand voice for transmission through a PCM channel. In Europe, A law involves converting a 12-bit input signal to an 8-bit encoded output. In the US, μ law involves encoding 13 bits to 8 bits. You can use a translator to convert from A law to μ law (**Figure 1**). The converter is asynchronous and requires only an 8-bit A law input to provide an 8-bit μ law output.

In A law, the input level divides into eight regions in which a uniform 4-bit conversion takes place. Regardless of the region, the output encodes 16 possible values. Each region corresponds to a segment in **Figure 2**, and the lower values have a better resolution (this **figure** shows only segments 0 through 5). To encode the input takes 8 bits; 4 bits indicate the uniform converted value in the segment, and the other 4 bits divide to represent the segment value itself (S0 to S7, coded with 3 bits) and whether the signal is positive or negative (1 bit).

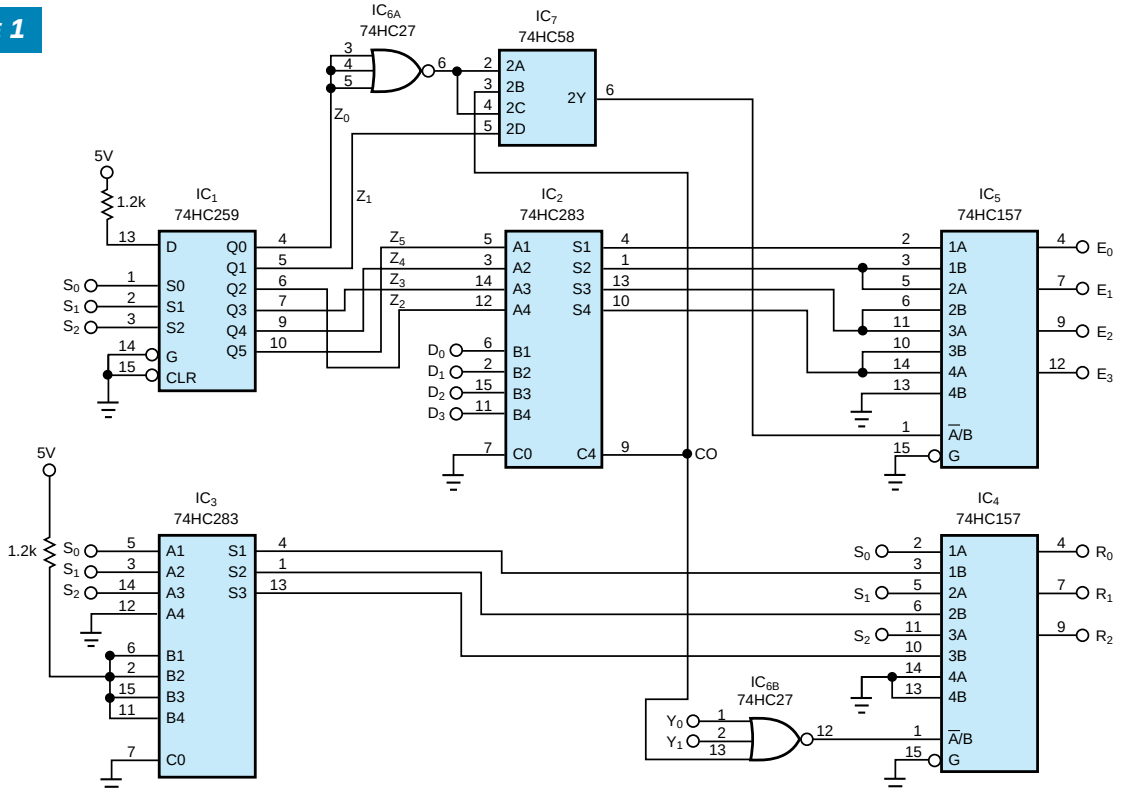
Alternatively, with μ law, also included in **Figure 2**, all but the first segments have a wider dynamic range and thus more spaced quantization levels (for 4 bits) compared with A law. Instead of 12 bits, 13 bits imply a wider dynamic range but a worse resolution for low input levels.

Figure 2 also illustrates the loss of resolution when converting the output A of A law to output A' of μ law. Depending on the law, either 8 bits (A law) or 4 bits (μ law and higher quantization levels) represent the value, therefore, the transitions occur faster around output A than around output A'. For the A to A' translation, the slope of A law is twice the slope of μ law.

Although information loss occurs during the conversion of value A, the same is not true for B. For B, the A law and μ law slope are the same, and the quantization level is the same. Thus, the difference between B and B' involves only a translation and a change of segment (B in S4, B' in S3). A simple comparison shows that the A value suffers a translation and a loss of information but remains in the same segment after conversion.

The design of the encoder must take into account the A law signal's segment and offset value, as does the following algorithm for which the A law input signal is PSD, and the μ law output signal is QRE, for which P,Q=polarity (1 bit), S,R=segment (3 bits) and D,E=value (4 bits):
If S=0, then Q=P, R=S, and E=D.
If S=1, then Q=P, R=S, and E=D/2.

FIGURE 1

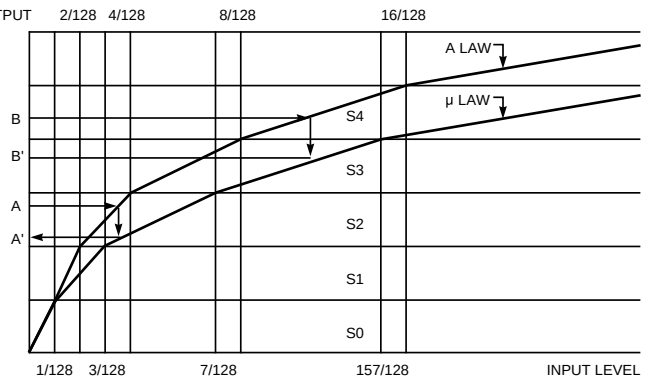


This A law-to-μ law translator inputs values of S and D and outputs E and R according to a specific algorithm.

If S=2 and D<8, then Q=P, R=S–1, and E=D+8.
If S=2 and D>7, then Q=P, R=S, and E=(D–8)/2.
If S=3 and D<12, then Q=P, R=S–1, and E=D+4.
If S=3 and D>11, then Q=P, R=S, and E=(D–12)/2.
If S=4 and D<14, then Q=P, R=S–1, and E=D+2.
If S=4 and D>13, then Q=P, R=S, and E=(D–14)/2.
If S=5 and D<15, then Q=P, R=S–1, and E=D+1.
If S=5 and D>14, then Q=P, R=S, and E=(D–15)/2.
If S=6, then Q=P, R=S–1, and E=D.
If S=7, then Q=P, R=S–1, and E=D.

According to this algorithm, the conversion requires both addition and subtraction, depending on S and D. You can express each subtraction as an addition to implement both in the same circuit. Thus, you can express the algorithm as follows, where CO=Carry out:
If S=2 and D<8, then Q=P, R=S–1, Z=8, and E=D+Z (CO=0).
If S=2 and D>7, then Q=P, R=S, Z=8, and E=(D–8)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
If S=3 and D<12, then Q=P, R=S–1, Z=4, and E=D+Z (CO=0).
If S=3 and D>11, then Q=P, R=S, Z=4, and E=(D–12=D–16+Z)=(D+Z)/2 (CO=1).
If S=4 and D<14, then Q=P, R=S–1, Z=2, and E=D+Z (CO=0).
If S=4 and D>13, then Q=P, R=S, Z=2, and E=(D–14)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
If S=5 and D<15, then Q=P, R=S–1, Z=1, and E=D+Z (CO=0).
If S=5 and D>14, then Q=P, R=S, Z=1, and E=(D–15)/2=(D–16+Z)/2=(D+Z)/2 (CO=1).
The value of Z depends on S: $Z=2^{5-S}$. Once you define Z, the algorithm performs the same D+Z operation for each S. The carry-out (CO) signal determines whether R is equal to S or S–1. Therefore, this implementation simultaneously solves two problems. Furthermore, the same technique applies for S=6 and S=7, when Z=0.

FIGURE 2



Converting output A of A law to A' of μ law incurs a loss of information. However, no information loss occurs when converting from B to B', because the slopes of the two curves are the same at that point.

In **Figure 1**, a 3×8 decoder, IC_1, converts S to Z, which IC_2 adds to D. If the CO is a 1, E is (D+Z)/2; otherwise, R is S–1. To choose between both options, the circuit uses the CO signal to control data selectors IC_4 and IC_5. These devices select between two possible outputs: S or S–1 and D+Z or (D+Z)/2, respectively. A second adder, IC_3, implements S–1 by summing the S inputs with 15. The circuit derives (D+Z)/2 by shifting D+Z into the inputs of data selector IC_5. Additional logic ensures that no conversion occurs when S=0 and that E=D/2 when S=1.

The 8-bit input is P0/S2/S1/S0/D3/D2/D1/D0, and the 8-bit output is P0/R2/R1/R0/E3/E2/E1/E0. The schematic doesn't show P0 because this parameter's value doesn't change. The circuit was tested with a Motorola (www.mot.com) MC145554 μ law PCM codec-filter and an 8TR641 (AT&T, www.att.com) E1 multiplexer. (DI #2192)

To Vote For This Design, Circle No. 412