Care and Feeding of the One Bit Digital to Analog Converter

Jim Thompson, University of Washington, 8 June 1995

Introduction

The one bit digital to analog converter (DAC) is a magical circuit that accomplishes D/A conversion without using any analog components. This is a neat trick. The matched resistors required by conventional current summing DACs become more and more difficult to make as the number of bits per word increases. Even with laser trimming, temperature changes can adversely affect the linearity of such converters. Avoiding the use of matched resistors results in a converter that is not only cheaper, but is inherently linear.

These advantages don't come free, though. First, the noise level in a one bit DAC is related to the speed at which it operates. Very low noise levels require very high clock speeds. Second, and perhaps more important to engineering students, the one bit DAC is at least an order of magnitude more difficult to understand than a current summing DAC.

That's where this paper comes in. I can't do anything about the clock speed problem, except maybe to show you that it's not so big a problem after all, but I hope that I can eliminate once and for all the mystery associated with this little circuit.

The Circuit

Technically, the circuit we call a 'one bit DAC' is really a 'delta sigma modulator with one bit DAC', but that's too many words for the front panel of a CD player. The delta sigma modulator is a circuit that translates a binary number into a pulse train whose duty cycle (the fraction of time that the signal is high) is proportional to the binary input. This pulse train is then converted into an analog signal by averaging it over time with a low-pass filter.

The delta sigma modulator gets its name from its block diagram (figure 1). The 'delta' refers to the first block, which calculates an error signal, or the difference between the input and the output. The 'sigma' is the second block, which accumulates or sums this error signal.
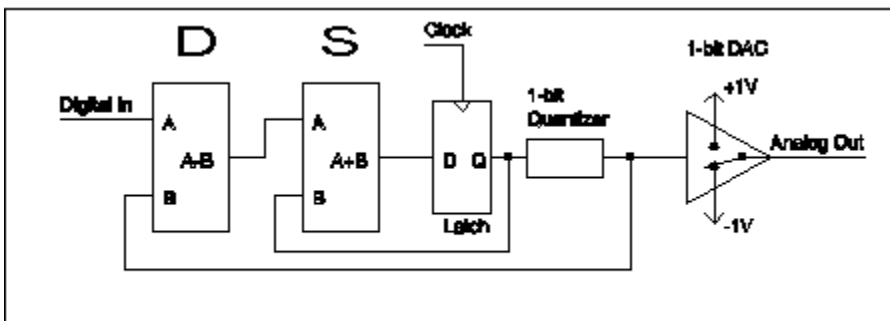

Figure 1. Delta-sigma modulator, block diagram

Explanation 1: The Analog Approach

Now, what do these blocks do? Look at an analog version of this circuit (figure 2). Note that this is a closed-loop system with negative feedback. As with any such system, the positive input can be thought of as the

1

'reference' input. As long as the loop is stable, it will tend to cause the negative input to track the input.
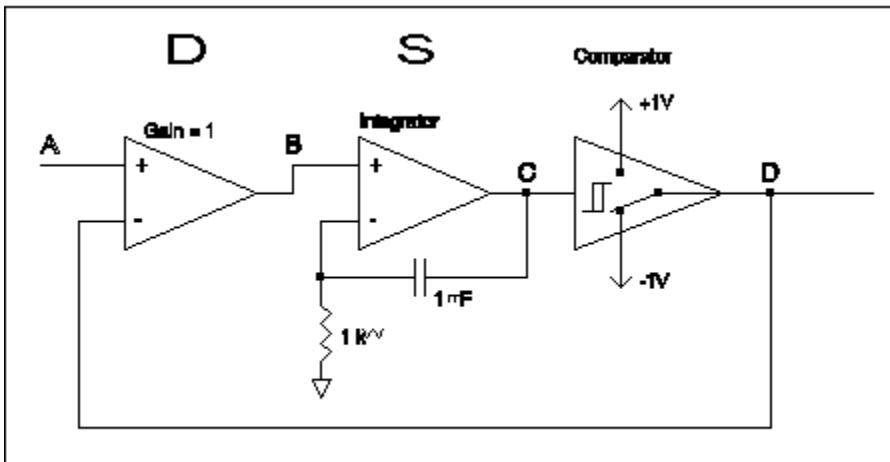

Figure 2. Analog delta-sigma modulator

In this circuit, the output is connected directly to the negative input, so the overall gain is one. Since there is a very high loop gain, due to the presence of a comparator inside the loop, the overall gain should be exactly one. Note the hysteresis in the comparator. This approximates the time delay inherent in a digital accumulator circuit, and allows our analysis to proceed in finite time steps. The output switches to +1 V when the input is greater than +0.1 V, and to -1 V when the input is less than -0.1 V. Also to make calculations simple, the integrator component values were chosen to give +1 V/mS output ramp for each volt of input.
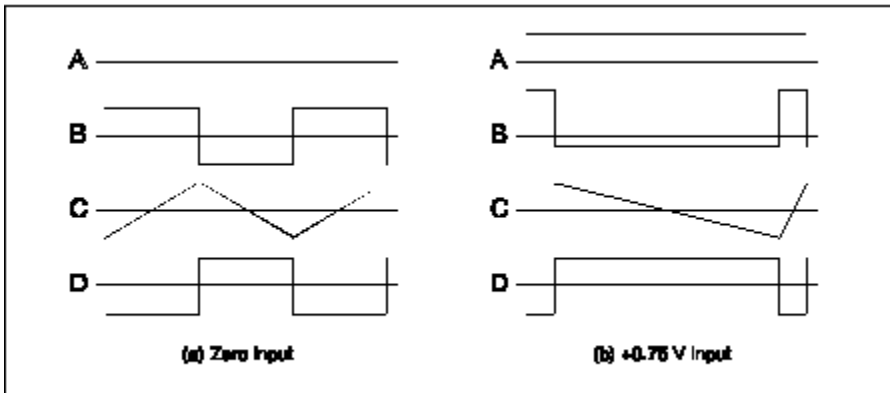

Figure 3. Analog waveforms

Watch what this circuit does when we give it a zero input at node A (figure 3a). Assume the integrator output at node C is initially -0.1 V, and the output at node D is -1 V. The initial value at node B, then, is +1 V. This causes the integrator to ramp positive at a rate of +1 V/mS. It continues to do so for 200 uS, until node D reaches the comparator threshold of +0.1 V. At this time, the comparator switches states, and node D goes to +1 V. Node B goes to -1 V, and the integrator begins ramping negative, now at -1 V/mS. It does so for 200 uS, at which time node C reaches the comparator's negative threshold of -0.1 V, causing node D to go back to -1 V. Note that this was the initial state, and the sequence will repeat, ramping alternately positive for 200 uS, and negative for 200 uS. The corresponding output at node D is negative for 200 uS and positive for 200 uS. The duty cycle of the output is 0.5. Since the output switches from +1 V to -1 V, the average voltage with a 0.5 duty cycle is, guess what, 0 V.

Now let's try it with a different input (figure 3b). Put +.75 V at node A, using the same initial conditions, node C = -0.1 V, node D = -1 V. Now, due to the input voltage, the output of the differential amp at node B is +1.75 V. The integrator ramps at +1.75 V/mS for (0.2 V / 1.75 V/mS) = 114.3 uS, when node C reaches +0.1 V. Now node D goes to +1 V, and node B to -0.25 V. Now the integrator ramps negative at only -0.25 V/mS, and it takes 800 uS for node C to return to -0.1 V. Once again, the initial state has been reached and the circuit will

oscillate. Notice, though, that the output is positive for 800 uS, and negative for 114.3 uS, for a total period of 914.3 uS. The duty cycle is 800/914.3 = 0.875. With the output switching between +1 and -1 V, the average voltage is (1 V * 800 uS + -1 V * 114.3 uS) / 914.3 uS = +0.75 V. Our closed loop still has an overall average gain of 1.

Similarly, for an input of -0.75 V, the duty cycle will be 0.125. Prove it to yourself. In fact, there is a linear relationship between the input voltage and the output duty cycle and the average output voltage. The input voltage always causes the integrator to ramp faster in the direction of the input voltage than in the opposite direction.

This is all very nice, but if we were to actually build this circuit, and we tacked a low-pass filter on the end of it to average the output, what we would have built is an analog to analog converter. Not very useful.

Explanation 2: The Digital Version

I started with the analog circuit, because it's easy to look at continuous-time waveforms and understand what's happening. When looking at a sequence of states in a digital circuit, it's a little harder to see, since all of the signals are binary numbers. Keep in mind, though, that this is the SAME CIRCUIT, and the same things are happening.

Note that this circuit is modified from Figure 1, so that it does not require a subtracter circuit. We use 10-bit adders to ensure that they do not overflow, and the subtraction is performed by adding the 2's complement negative of the quantizer output. The quantizer in this example is very simple: We take the most-significant bit of the signal at node C. To make the overall gain of the loop unity, the range of this signal must be the negative of the range of the input signal. The feedback signal therefore must switch between zero and -256. Fortunately, it is easy to multiply a single bit number by a constant. The feedback signal is simply wired to the [[Delta]] adder such that when it is high, all of the bits that are high in the number -256 (1100000000 binary) are set high; when the feedback signal is low, all inputs are low.

For digital representations of bipolar analog signals, it is standard practice for the most negative analog input to be represented by binary 0, and the most positive input to be represented by 2^N-1, for an N-bit binary word. An input of zero is then 2^{N-1}. For N=8, this is known as an 'excess-128' code, and is equivalent to 2's complement, with the most-significant bit (sign bit) inverted. Note that this convention applies only at the input, node A. At all other nodes, the numbers will be treated as 2's complement.
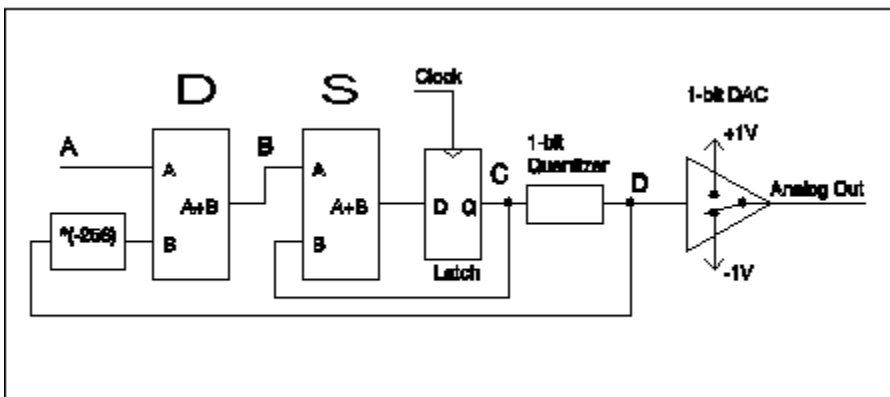

Figure 4. Modified digital circuit

Using excess-128 code for the input, we will now walk through an 8-bit version of the digital circuit with different inputs. The input range is between -1 and +1 volt, so the excess-128 code is (128 * Vin) + 128

OK, here goes. Assume an initial value at node C of 512. Put the code 128 (corresponding to an analog input of

0 V) on node A. (See table 1 for summary.) Since node D is high, node B now goes to 128 - 256 = -128. The [[Sigma]] adder adds this value to its previous state. On the rising edge of the next clock, node C goes to 512 - 128 = 384 and the output at node D goes low. Now node B goes to 128 + 0 = 128, and on the next clock, node C goes to 384 + 128 = 512. This is the same as the initial state, so the circuit oscillates, with node D high one cycle and low the next. The duty cycle is .5, and the output of the DAC is zero.

Table 1. Digital DAC operation, input = 0 V

t	A	B	C	D
0	128	-128	512	1
1	128	128	384	0
2	128	-128	512	1

For other input codes, the pattern doesn't repeat so quickly. Table 2 shows the values of each node for an input of 0.75 V, or excess-8 code 224. Values are shown after adder outputs settle from each state change, so for each state, C = Ct-1 + Bt-1, D = (0 if

C < 512; 1 if C ^{3} 512), B = A - 256 * D.

Table 2. Digital DAC operation, input = 0.75 V

t	A	B	C	D
0	224	-32	512	1
1	224	224	480	0
2	224	-32	704	1
3	224	-32	672	1
4	224	-32	640	1
5	224	-32	608	1
6	224	-32	576	1
7	224	-32	544	1
8	224	-32	512	1

State 8 is identical to state 0, so the pattern repeats every 8 clock cycles, with seven states high and one state low at D. The average output of the 1-bit DAC is (7 - 1)/8 = 0.75 V.

Notice that the same thing is happening that happened in the analog circuit. When the input is greater than zero, the [[Sigma]] adder counts up faster than it counts down. This directly corresponds to the analog integrator, ramping up faster than it ramped down for positive input signals.

One more example, just because it's so hard to believe. This time, the input is -0.5 V, or excess-8 code 64 (table 3.)

Table 3. Digital DAC operation, input = -0.5 V

t	A	B	C	D
0	64	-192	512	1
1	64	64	320	0
2	64	64	384	0
3	64	64	448	0
4	64	-192	512	1

This time, the pattern repeats in only 4 states, with one state high and three states low. The average output is (1 - 3)/4 = -0.5 V.

One more thing that needs to be mentioned: The number of states that the circuit goes through before it repeats

is not a simple function of the input voltage, so the frequency of pulses, or the width of these pulses, doesn't mean a thing. This is not a pulse width modulator. Rather, it is a pulse density modulator. Only the average value of the output, or the percentage of time that the output is high, is significant.

To prove this: just one more example, this time, an input of 3/8 volt. The excess-128 code is 176. (table 4.)

This time, the sequence repeats in 16 states, but it generates five pulses during this sequence. Furthermore, four of the pulses are two clock periods wide, and the fifth is three clock periods. This is nonsense. But the ratio, 11 high out of the 16 states, gives us a DAC output of (11 - 5) / 16 = 3/8 V. I told you it was magic.

Table 4. Digital DAC operation, input = 0.375 V

```
t       A         B         C         D
0       176       -80       512       1
1       176       176       432       0
2       176       -80       608       1
3       176       -80       528       1
4       176       176       448       0
5       176       -80       624       1
6       176       -80       544       1
7       176       176       464       0
8       176       -80       640       1
9       176       -80       560       1
10      176       176       480       0
11      176       -80       656       1
12      176       -80       576       1
13      176       176       496       0
14      176       -80       672       1
15      176       -80       592       1
16      176       -80       512       1
```

Hey, Wait a Minute. What Gives Here?

By now you're sure to have guessed that it takes a number of cycles of the clock in order to be able to average the output with any kind of precision. That's true. In fact, with an 8-bit converter like this, the clock would have to run 256 times as fast as the input data sample rate in order to resolve each sample to full 8-bit precision. This isn't bad enough, though. For a 16-bit converter, to get full resolution out of each sample, the clock would have to run at 65,536 times the incoming sample rate. For a CD data stream, this would require a clock of 2.9 GHz. This is, believe it or not, impractical. Fortunately, it is also unnecessary.

If you wish to see a full analysis of what magic happens to resolution and noise level, see the papers in the references. What follows is a non-rigorous explanation.

Look again at Table 4, the conversion of 3/8 V. It took a full 16 clock periods to resolve this number to the nearest 1/8 volt. But if you look only at samples 0 through 3, the voltage averaged over this time is (3 - 1) / 4 = 1/2 V. This is accurate to the nearest 1/4 volt. In the worst case, as in samples 1 through 4, the average is 0 V, which is still within 1/2 V of the correct value.

The point is, the more samples we average, the better resolution we get, but even with only a few samples, we still get some degree of accuracy. Another way of saying this is, signals with long periods can be resolved better than signals with short periods. Or, one step further, low-frequency signals are resolved better than high-frequency signals. Just one tiny step further, since signal to noise ratio is approximately 6 dB per bit, and we have this blockbusting statement: high-frequency sounds are noisier than low-frequency sounds.

So What?

So what, indeed. Enter human perception. The dynamic range of a CD is 96 dB. That means that the noise level is 96 dB below the maximum amplitude. But the human ear only needs this good a noise floor in a narrow range of frequencies around 5 kHz. At 10 kHz, our sensitivity is about 12 dB poorer, so we can get away with two fewer bits of resolution. At 20 kHz, we're down at least 30 dB, or 5 bits. What this means is that we really only need maximum resolution up to a sampling rate of 10 kHz (to resolve those 5 kHz signals, keeping the Nyquist limit in mind). Furthermore, the 96 dB range represents a loudness range from just-audible sounds in an anechoic chamber to the sound of a heavy truck at full throttle. In most listening environments, this is slightly more range than necessary.

Assuming that 14 bits of resolution (84 dB) really is necessary at 5 kHz, with the required bit rate decreasing at higher frequencies, acceptable performance should be possible using $2^{14} * 10$ kHz, or 164 MHz clock speed. Of course, this is still a bit high, but it beats the heck out of 2.9 GHz, doesn't it. (In practice, other techniques are used to get acceptable noise performance, such as nesting delta-sigma modulators inside each other, or using delta-sigma modulation in conjunction with DACs of more than one bit. See Finck, 1989.)

I Lied

One more thing. I said a while back that the analog version of this circuit was of limited use. In fact, this is the circuit that is used for delta-sigma analog to digital conversion. By taking an analog signal and converting it to an oversampled 1-bit digital signal, we're half way there. What remains is to filter the digital signal using a low-pass finite impulse response (FIR) filter. The same noise and resolution arguments apply, and the same advantage in linearity is achieved as in delta-sigma digital to analog conversion.

Conclusion

There you have it. Not so frightening, after all, is it?

References

1. "High Performance Stereo Bit-Stream DAC with Digital Filter" R. Finck, IEEE Transactions on Consumer Electronics, Vol. 35, No. 4, Nov 89.

2. "Delta-Sigma Techniques" Crystal Semiconductor Corp, Austin Tx, AN10REV1, Jan 89.

3. "The Science of Sound" Thomas D. Rossing, Addison-Wesley, 1990, pp 86, 92.

4. "Delta-Sigma Modulated First Order Digital to Analog Converter" Troy Gilliland, Montaigne Van Allen, University of Washington, Jun 94.