The tones used you hear when dialing a number on a telephone are called **dual-tone multifrequency** (DTMF) tones. Each "tone" consists of two superimposed frequencies. The DTMF system is based on a 4 x 4 matrix with four row frequencies (697, 770, 852, 941 Hz) and four column frequencies (1209, 1336, 1477, and 1633 Hz). Most telephones only implement 3 of the 4 columns, and the resulting 12 positions in the matrix are used just like the telephone dial pad implies they are. That is, the digit 2 is the first row, second column, and consists of tones at 770 Hz and 1209 Hz mixed together.

There are a number of ways of generating these tones, and one common method is to use a recursive digital resonator – this might be done, for example, in a cell phone. The resonator is implemented as a two-pole filter described by the following difference equation:

$$X_n = k * X_{n-1} – X_{n-2}$$

where *k* is a constant defined as:

$$k = 2 \cos(2\pi * \text{toneFrequency} / \text{samplingRate})$$

The eight values of *k* required for DTMF dialing can be precomputed and stored in ROM. For example, the constant required to produce a Column 2 tone (770 Hz) at a sample rate of 8 kHz is:

$$k = 2 \cos(2\pi * 770 / 8000) = 2 \cos(0.60) = 1.65$$

One more value must be calculated: the initial impulse required to make the oscillator begin running. Clearly, if X_{n-1} and X_{n-2} are both zero, every succeeding X_n will be zero. To start the oscillator, set X_{n-1} to zero and set X_{n-2} to:

$$X_{n-2} = -A * \sin(2\pi * \text{toneFrequency} / \text{samplingRate})$$

In our example, assuming a sine wave with unit amplitude is desired, this reduces to:

$$X_{n-2} = -1 * \sin(2\pi * 770 / 8000) = -\sin(0.60) = -0.57$$

Reducing this to code is simple: first, two intermediate variables (X1, X2) are initialized. X1 is initialized to zero, while X2 is loaded with the initial excitation value (-0.57) to start the oscillation. To generate one sample of the sinusoid, perform the following operation:

```
X0 = k * X1 – X2
X2 = X1
X1 = X0
```

Each new sample on the sine wave is calculated using one multiplication and one subtraction.

Now, if you actually intended to dial a number on the phone pad, you would run two of these iterations at the same time – one for the column tone, and one for the row tone. For example, suppose you wanted to dial an "8". Then the generator would have to create the Column 2 tone (770 Hz, as above) as well as the Row 3 tone (1477 Hz). If you do the calculations outlined above, you'll find you need to use k=0.8 and an initial value of -0.92 to generate the Row 3 tone.

In pseudo-code we could write this pair of resonators as:

```
3 kx = 1.65
4 X2 = -0.57
5 X1 = 0.0
6 ky = 0.80
7 Y2 = -0.92
8 Y1 = 0.0

  while (button 8 is pressed) {
        X0 = kx * X1 – X2
        Y0 = ky * Y1 – Y2
        X2 = X1
        X1 = X0
        Y2 = Y1
        Y1 = Y0
        Store X0        # simulates tone output
        Store Y0        # pick any (fixed) address you like
  }
```

If the "8" button was pressed for something like 0.1 seconds, then this loop would run 800 times because we're sampling at 8 kHz. I'm not sure how quickly your simulators are executing, so to be careful let's run just 100 iterations. You should be able to assemble this code so that it only refers to registers once you're inside the loop (except for the stores, of course). As long as you don't break the algorithm you're free to change the order of the instructions inside the loop if that makes the assembly easier.

Acknowledgement: The resonator code and explanation of DTMF signaling comes from the Maxim IC Web site, at www.maxim-ic.com.