A COMPARISON OF NOISE-SHAPING CLOCK GENERATORS FOR SWITCHED-CAPACITOR FILTERS *

Raymond W. Zeng[†] *and Paul J. Hurst*
Solid-State Circuits Research Laboratory
Department of Electrical and Computer Engineering
University of California, Davis, CA 95616
email: hurst@ece.ucdavis.edu
phone: 530-752-0583

Feb. 11, 2000

ABSTRACT

A programmable clock generator that uses noise shaping can generate a wide range of sampling frequencies with fine resolution for switched-capacitor filters (SCFs). In this paper, different noise-shaping transfer functions for such a clock generator are compared through simulation and measurement of two SCFs driven by the clock generator. A tone detector that uses a bandpass SCF driven by the programmable sampling-clock generator is also described.

* This work was supported in part by UC MICRO grant 96-077 and 97-171.

[†] R. Zeng is now with Intel Corp., Folsom, CA, 95630.

1. INTRODUCTION

A switched-capacitor filter (SCF) provides an accurate and stable frequency response that is determined by well-defined capacitor ratios and a crystal-derived sampling clock. The frequency response of a SCF scales with its sampling frequency; therefore, the filter response can be programmed by changing the sampling frequency f_S. If the sampling clock f_S is realized by integer division of a high-frequency system clock f_C, only widely spaced sampling frequencies can be generated. A programmable sampling-clock generator with noise shaping can generate a wide range of sampling frequencies with fine resolution, as described in [1]. The generated clock f_S has nonuniformly spaced sampling edges that align with the edges of a fast system clock f_C; such alignment helps to reduce coupling of switching noise from digital circuits into the SCF. In [1], the jitter in the sampling edges was concentrated at high frequencies by a sixth-order noise-shaping loop. When such a sampling-clock generator was used to clock a bandpass SCF, the center frequency of the bandpass filter (BPF) was programmable with a resolution of 0.08%. One drawback of this approach is that the input to the filter must be bandlimited to well below f_S to keep the effect of the clock jitter small, but this can often be achieved by using an imprecise continuous-time lowpass filter before the SCF with a programmable sampling frequency.

Measured results for two SCFs revealed that the signal-to-noise ratio (SNR) was limited by circuit noise sources (from the op amps and switched capacitors) rather than by sampling noise caused by the jitter in the sampling clock [1]. Based on these results, lower order noise-shaping clock generators that give lower signal-to-sampling-noise ratio (SSNR) and require less IC area are worth investigating. This paper considers such clock generators and presents simulated and measured performance for different clock generators driving two SCFs.

2. CLOCK GENERATOR

To generate a sampling clock with predominantly high-frequency jitter, the clock generator in Fig. 1 is used. The clock generator is composed of two major blocks: a quantizing, noise-shaping loop and a number-controlled oscillator (NCO). The input is a constant D that equals the fast-clock frequency divided by the desired sampling frequency ($D = f_C/f_S$). The input D is a $B.F$-bit number, that is, it has B integer bits and F fractional bits. The output of the noise-shaping loop is a B-bit integer $P[n]$, which is the instantaneous normalized sampling period at time n. To produce the sampling clock f_S, the NCO divides the fast clock f_C down by $P[n]$. Therefore, the n^{th} sampling instant, t_n, is given by

$$t_n = t_{n-1} + P[n]T_C = nT + \tau[n], \tag{1}$$

where $T = 1/f_s$ is the desired sampling period, $T_C = 1/f_C$ is the fast-clock period, and $\tau[n]$ accounts for the jitter due to the nonuniform sampling.

Initially assume that $d[n] = 0$ in Fig. 1. (Nonzero dither $d[n]$ will be considered in the next section.) The feedback loop has a multi-level quantizer Q in the forward path and a finite-impulse-response (FIR) feedback filter $G(z)$ [2]. The quantizer truncates the fractional bits in its input to produce a B-bit integer output $P[n]$. The quantization noise $N[n]$, which is the difference between the quantizer input and output, is fed back to the input through filter $G(z)$. The transfer function $H_T(z)$ from the quantization noise $N[n]$ to the jitter $\tau[n]$ in the sampling clock is [1]

$$H_T(z) = \frac{\tau(z)}{N(z)} = \frac{G(z) - 1}{1 - z^{-1}} T_C. \tag{2}$$

Assuming $N[n]$ is white, the spectrum of the jitter $\tau[n]$ is set by $H_T(z)$. To assure low jitter in a specified lowpass region, $H_T(z)$ typically has transmission zeros near or at dc [2].

3. COMPARING DIFFERENT G(z)

In [1], a sixth-order $G(z)$ gave a simulated SSNR at the filter output that exceeds 70 dB for low input frequencies ($f_{in} < 0.1 f_s$) for a second-order BPF and a fifth-order LPF. (The noise in the simulated SSNR includes only the noise in the samples due to jitter; it does not include circuit noise.) However, measured results under the same conditions revealed that the output SNR was typically limited to about 60 dB [1]. Therefore, the SNR was apparently limited by circuit noise sources (from the op amps and switched capacitors) rather than by sampling noise. Based on these results, lower order $G(z)$ that give lower SSNR and are simpler to implement are worth considering. The objectives when selecting a $G(z)$ are:

1- A high simulated SSNR at the SCF output for input signals in a specified bandwidth, so that the actual output SNR is limited by circuit noise rather than sampling noise.

2- Low noise gain, which is the sum of the absolute value of the coefficients of $G(z)$ [3], to keep the peak-to-peak jitter in f_s small. The peak-to-peak jitter determines the minimum half-clock period during which the op amp(s) in a SCF must settle [1].

3- Highpass (or perhaps white) jitter.

Five different $G(z)$ that were investigated are listed in Table 1. $G_0(z)$ is the sixth-order transfer function that was used in [1]. The other four $G(z)$ are lower order and are simpler to implement since almost all of the coefficients of these $G(z)$ are powers of two. Each $G(z)$ gives at least one zero at $z = 1$ in $G(z)–1$ to cancel the pole at $z = 1$ in $H_T(z)$ in (2). For $G_1(z) = z^{-1}$, $H_T(z)$ is a constant which means the jitter is white if $N[n]$ is white. However, because D is a constant, $G_1(z)$-$G_4(z)$ generate periodic $P[n]$, causing tones to appear in the spectrum of $\tau[n]$ [4]. As a result, the sampling noise consists of tones rather than shaped noise [5,6]. To eliminate these tones, a dither signal d is added to the input of the quantizer Q as shown in Fig. 1 [6]. This dither is a 0.4-bit pseudo-random number that is uniformly distributed from 0 to 1. The total noise $N[n]$ in Fig. 1 is $N[n] = d[n] + q[n]$, where $d[n]$ is the dither and $q[n]$ is quantization noise.

The noise gain for each G is listed in Table 1. The peak-to-peak jitter is determined by the noise gain [1]. For G_0 with no dither signal added, the peak-to-peak jitter is $4T_c$ (4 is the noise gain of 3.5 rounded up to the next integer). Each of G_1-G_4 produces a peak-to-peak jitter equal to twice the noise gain times T_c because the added dither here doubles the peak-to-peak value of $N[n]$. Therefore, only G_1 gives a peak-to-peak jitter (= $2T_c$) that is less than the peak-to-peak jitter with G_0.

4. SIMULATED AND MEASURED RESULTS

The clock generator of Fig. 1 was built with $B = 6$ and $F = 6$, using a Xilinx XC4005 so that different $G(z)$ could be implemented easily. The bit lengths used for the signals in the clock generator are shown in parentheses in Fig. 1. As a measure of complexity, the percentage of configurable logic blocks (CLBs) used in the Xilinx chip to implement the different clock generators is given in Table 1. G_1-G_4 were built with a dither generator, and the complexity of the dither generator is included in the CLB counts. The clock generator using G_0 was also implemented (without dither) for comparison. Measurements confirmed that clock generators using these five different $G(z)$ produced a wide range of sampling frequencies with the same resolution. Therefore, they provided similar control over the frequency response of a SCF.

The five clock generators were used to clock two SCFs: a second-order BPF with $Q = 10$ and $f_o = 0.01 f_s$ and a fifth-order LPF with $f_{-3\mathrm{dB}} = 0.04 f_s$. These transfer functions are plotted in [1]. For the following simulations and measurements, the test conditions were: $f_C = 2$ MHz, $D = 20.25$ (which gives $f_s = 98.77$ kHz), $f_{in} = 1$ kHz and $V_{in} = 1.5\ \mathrm{V_{p\text{-}p}}$. (The SSNR is independent of the input amplitude, so the choice of amplitude is not important [1].)

Fig. 2a and 2b show the simulated spectra of the samples of the input sine wave using G_1 and G_4 in the noise-shaping loop, respectively. These samples are the input samples $x[n]$ in the model in Fig. 1b in [1]. The results for these two G are plotted because they are two extreme cases: a first-order and fourth-order $G(z)$. Note that the noise floors are shaped differently and in agreement with the corresponding H_T's in Table 1. In these simulations, the only noise is sampling noise.

4.1 *Lowpass SCF*

The simulated spectra of the LPF output, $|Y(\omega)|$, are shown in Fig. 3 for G_1 and G_4. Both spectra show the effect of the LPF's high-frequency attenuation on the noise floor. The passband noise floor for G_1 is much higher than for G_4, and the simulated SSNR for G_4 is 87.8 dB, which is much higher than the SSNR of 68.4 dB for G_1.

Fig. 4 shows the measured spectra of the LPF output for G_1 and G_4. These spectra were found by sampling the SCF output with a 14-bit ADC that is clocked by f_s and then taking a FFT of the samples. In both cases, the noise floor is higher at high and low frequencies than simulated, which means that circuit noise is much larger than sampling noise. Comparing Figs. 4a and 4b, the noise floor for G_4 in the SCF stopband is somewhat higher than for G_1, whereas the passband noise floors are about the same. As a result, the measured SNR, which includes the effect of sampling and circuit noise but excludes distortion, is larger for G_1 than for G_4, as listed in Table 2. In contrast, the simulated SSNR for G_1 is smaller than the SSNR for G_4. This discrepancy between the relative SSNRs and SNRs in Table 2 is most likely caused by circuit-noise sources, which are not included in the SSNR simulations. For comparison, the measured SNR with a uniform sampling clock with $f_s = 100$ kHz and $D = 20$ is 66.3dB for the LPF. This value is larger than the measured SNRs in Table 2. Therefore, it appears that the effect of circuit noise on the SNR is greater with a nonuniform sampling clock. Comparing the measured SNRs at the LPF output for G_1-G_4 in Table 2, G_1 gives the highest measured SNR.

4.2 *Bandpass SCF*

The simulated spectra of the BPF output, $|Y(\omega)|$, are shown in Fig. 5 for G_1 and G_4. The noise floor for G_1 is much higher than for G_4 at low and high frequencies due to the extra zeros in $G_4(z)-1$. The simulated SSNR at the BPF output for G_4 is 98.7 dB, which is much higher than the SSNR of 82.9 dB for G_1.

Fig. 6 shows the measured spectra of the BPF output for G_1 and G_4. Comparing Figures 6a and 6b, the noise floor for G_4 is slightly higher than for G_1. As a result, the measured SNR is larger for G_1 than for G_4 (see Table 2). Again, this result disagrees with the results from SSNR simulations since the simulated SSNR for G_1 is much smaller than the SSNR for G_4. The simulated SSNRs and measured SNRs at the BPF output for all the G's are listed in Table 2. Comparison of the measured SNRs for both SCFs reveals that G_1 has the highest SNR and G_4 has the lowest SNR among G_1-G_4. Therefore, considering measured SNR, peak-to-peak jitter, and complexity, $G_1(z)$ is a good choice for these two SCFs.

5. TONE DETECTOR EXAMPLE

One possible application of this clock generator is in a dual-tone multi-frequency (DTMF) receiver that uses a programmable BPF to detect one of many signaling tones. For DTMF signalling, 8 different frequencies (697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1336 Hz, 1477 Hz, and 1633 Hz) with amplitudes between –2 dBm and –32 dBm must be detected [7]. A block diagram of a DTMF tone detector is shown in Fig. 7. An imprecise second-order RC LPF with $f_{-3dB} << f_s$ ($f_{-3dB} \approx 5$ kHz) is used as a bandlimiting filter. After this RC LPF, the signal passes through a high-Q programmable bandpass SCF (the same second-order BPF described in Section 4, with $Q = 10$ and $f_o = 0.01 f_s$). This BPF is programmed to pass a narrow band centered at the tone frequency of interest. Its center frequency is set by changing D, the input to the noise-shaping clock generator. Since $f_o = 0.01 f_s$ for this BPF, f_s will be in the range from 69.7 kHz to 163.3kHz. The BPF center frequency is programmed by the noise-shaping clock generator that uses G_1. The filter output is then digitized and processed digitally to verify that a signal with the correct frequency and sufficient amplitude is present. Different tones can be detected by changing D to change the sampling frequency f_s and the center frequency of the BPF. With $D = 6.6$ bits and $f_c = 2$ MHz, the center frequency of the BPF can be set to within 0.06% of any of the 8 DTMF tone frequencies. The values of D used to detect the DTMF tones are given in Table 3. With simple amplitude qualification (using a peak detector) and frequency qualification (using a counter to check for a constant and correct period) by the digital-signal processor (DSP) (a Xilinx chip) in Fig. 7 [7], this bread-boarded system detects tones with a frequency that is within ±2.3% of the desired DTMF tone frequency with amplitudes between 10 dBV and –40 dBV.

6. CONCLUSION

A noise-shaping clock generator can use many different $G(z)$ to generate a wide range of sampling frequencies for SCFs. Based on simulation and measurement with five different $G(z)$ in two SCFs, a programmable clock generator that uses a simple first-order $G(z) = z^{-1}$ and dither can achieve acceptable performance for many filtering applications. First-order noise shaping is sufficient because the resulting sampling noise is much smaller than the circuit noise.

REFERENCES

[1] P. J. Hurst and B. C. Rothenberg, “A programmable clock generator that uses noise shaping and its application in switched-capacitor filters,” *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 403-411, April 1995.

[2] S.K. Tewksbury and R.W. Hallock, "Oversampled, Linear-Predictive and Noise-Shaping Coders of Order N>1," *IEEE Trans. on Circuits and Systems*, vol. 25, no. 7, pp. 436-447, July 1978.

[3] L.R. Carley and J. Kenney, "A 16-Bit 4'th Order Noise-Shaping D/A Converter," *Custom IC Conf.*, pp. 21.7.1-4, 1988.

[4] J. C. Candy and O. J. Benjamin, "The structure of quantization noise from sigma-delta modulation," *IEEE Trans. on Comm.*, vol. 29, pp. 1316-1323, Sept. 1981.

[5] W. Chou and R. M. Gray, “Dithering and its effects on sigma-delta and multistage sigma-delta modulation,” *IEEE Trans. on Inform. Theory*, vol. 37, no. 3, pp. 500-513, May 1991.

[6] S. R. Norsworthy, “Optimal Nonrecursive noise shaping filters for oversampling data converters part1: theory,” *ISCAS*, pp. 1353-1356, 1993.

[7] B.J. White, G.M. Jacobs, and G.F. Landsburg, "A monolithic dual tone multifrequency receiver," *IEEE J. Solid-State Circuits*, vol. 14, no. 6, pp. 991-997, Dec. 1979.

FIGURE CAPTIONS

Fig. 1: A block diagram of the clock generator, showing the noise-shaping loop in detail.

Fig. 2: Simulated spectrum of the SCF input samples $x[n]$ using a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.

Fig. 3: Simulated spectrum of the SC LPF output with a) G_1 in the noise-shaping loop and b) G_4 in the noise-shaping loop.

Fig. 4: Measured spectrum at the output of the SC LPF with a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.

Fig. 5: Simulated spectrum of the SC BPF output with a) G_1 in the noise-shaping loop and b) G_4 in the noise-shaping loop.

Fig. 6: Measured spectrum at the output of the SC BPF with a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.

Fig. 7: DTMF tone detector using a SC BPF and a noise-shaping clock generator. (DSP is a Xilinx XC4005)

Table 1: Comparison of different $G(z)$

Table 2: SNR at the filter outputs for different $G(z)$ with $D = 20.25$.

Table 3: Values for the DTMF tone detector in Fig. 7 with $f_C = 2$ MHz.

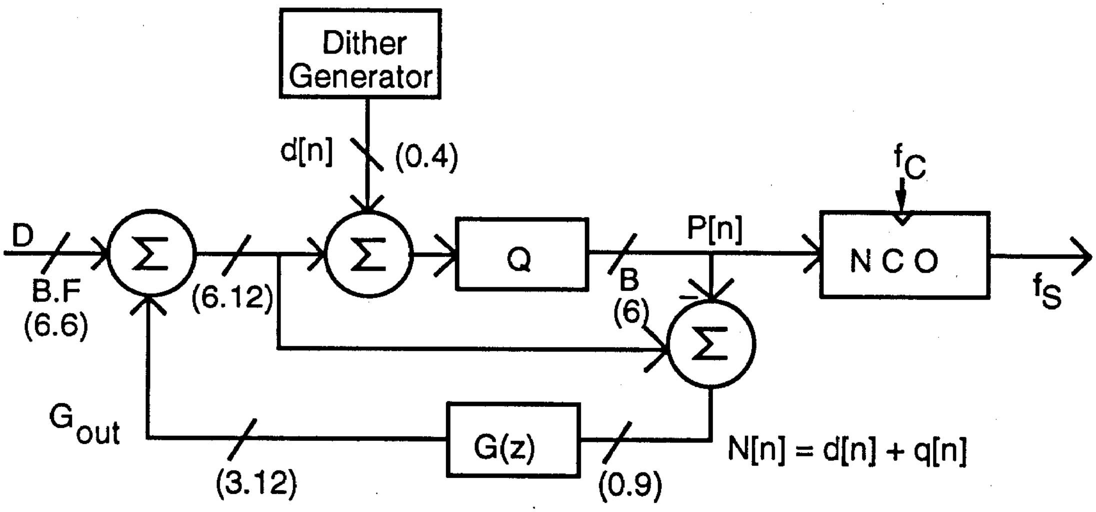


Fig. 1: A block diagram of the clock generator, showing the noise-shaping loop in detail.

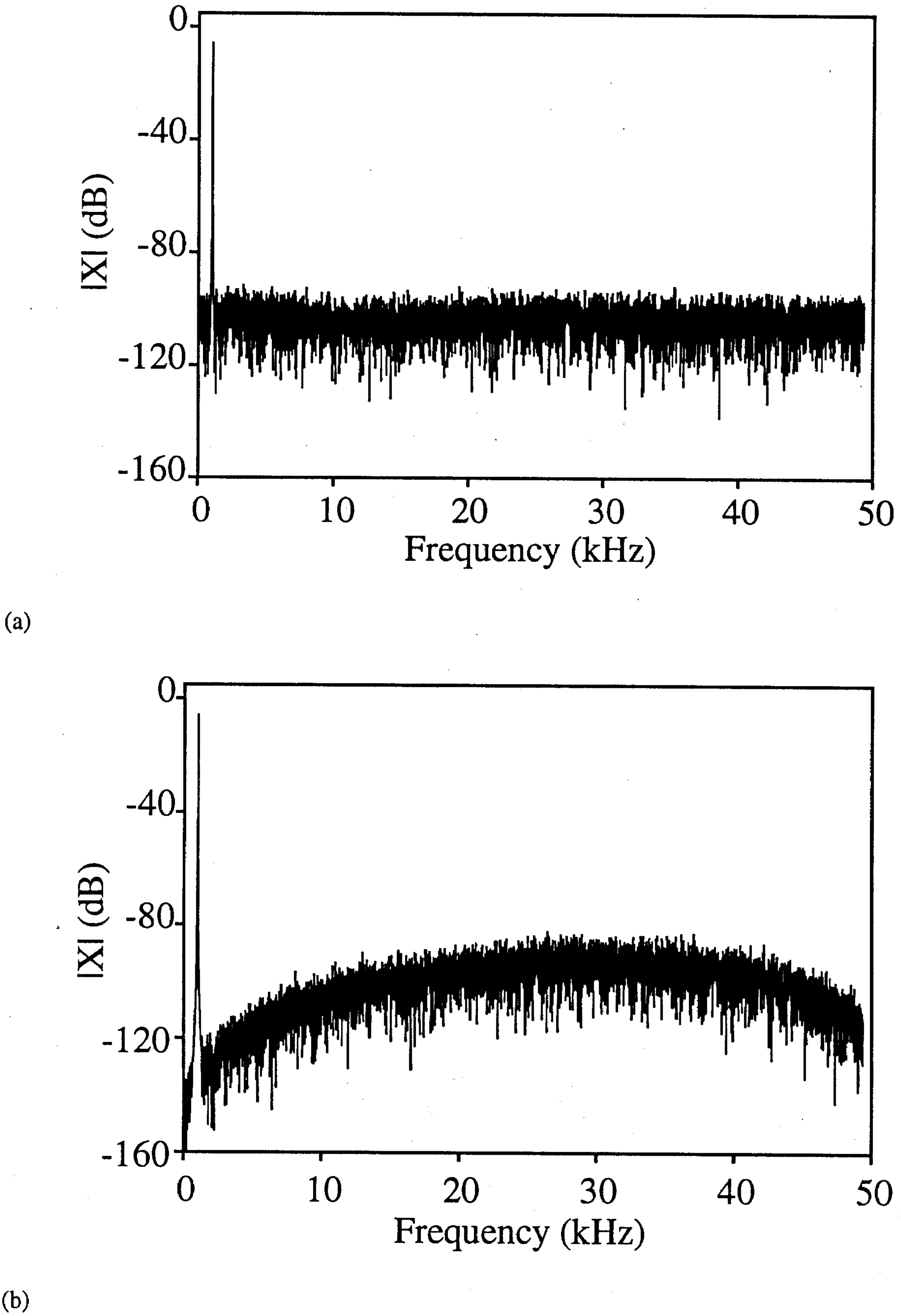


Fig. 2: Simulated spectrum of the SCF input samples $x[n]$ using a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.

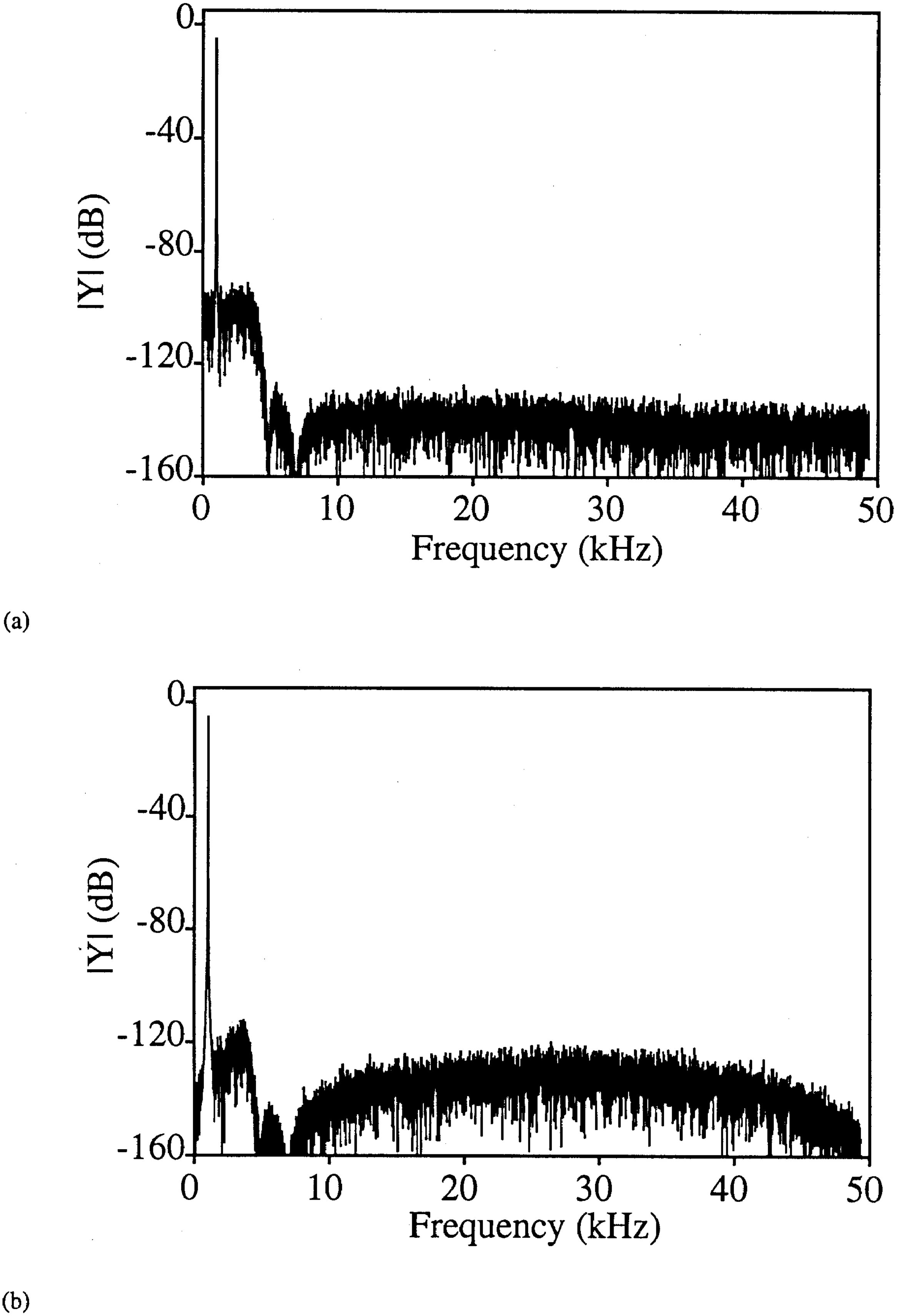
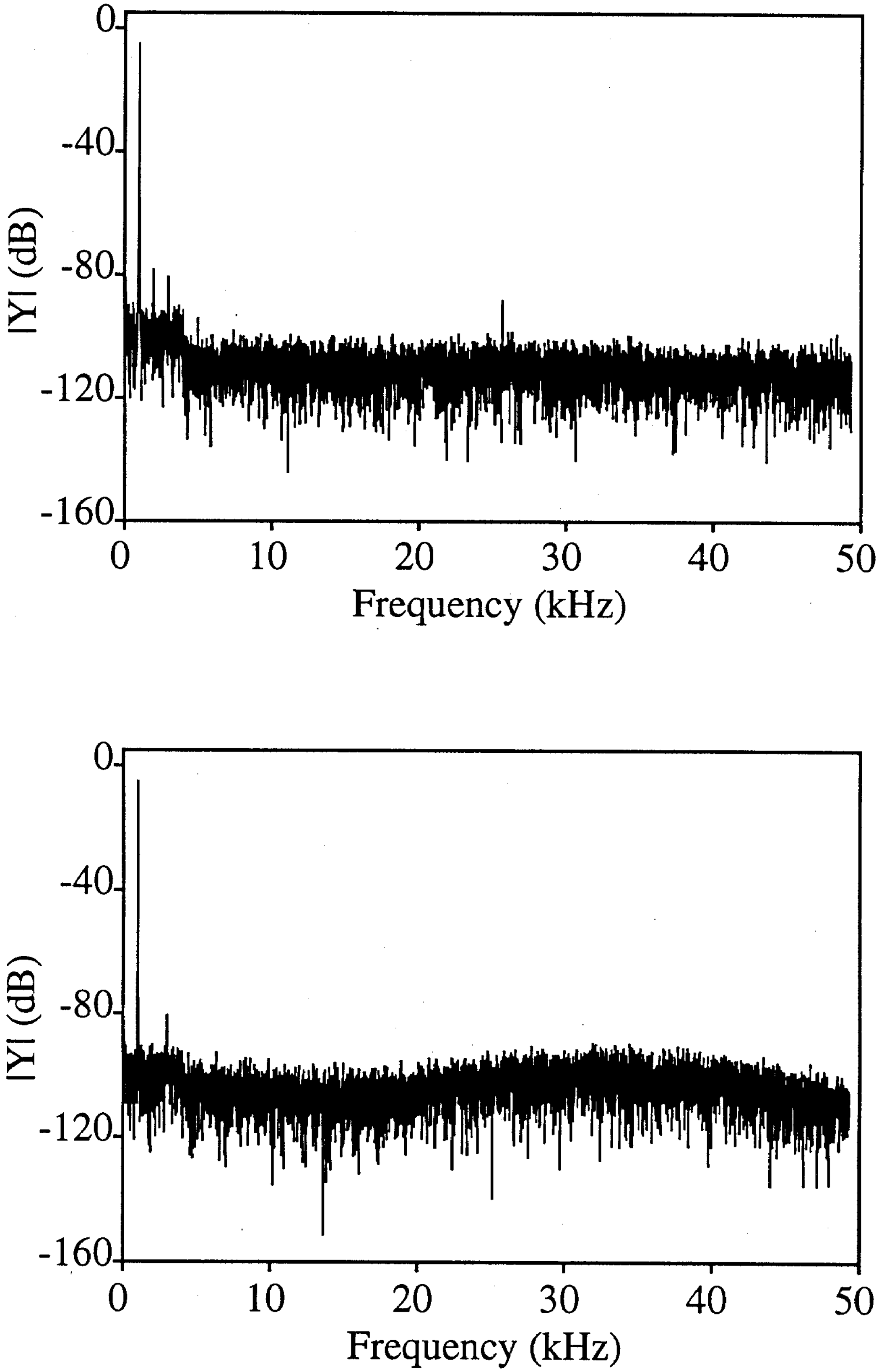


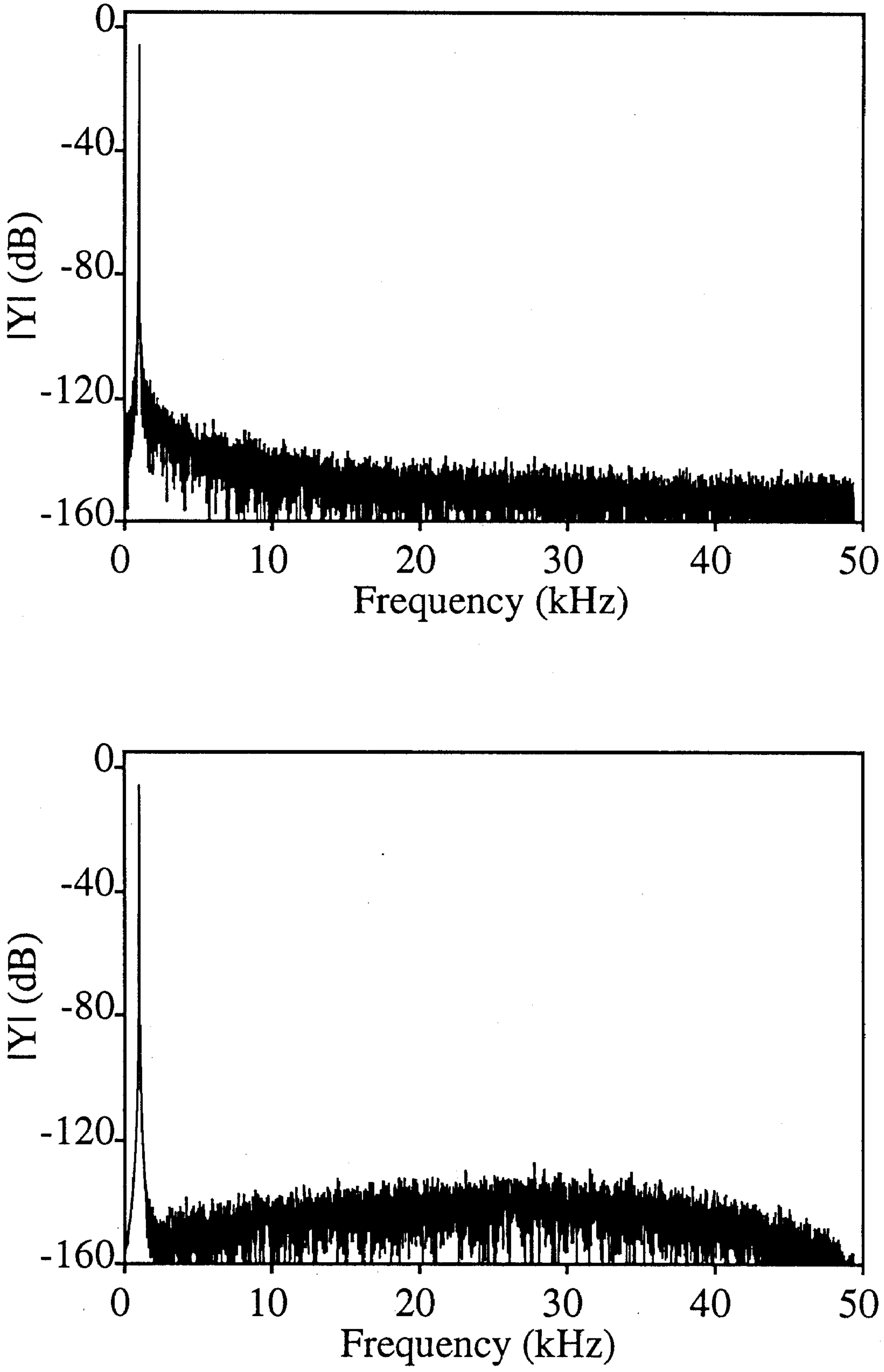
Fig. 3: Simulated spectrum of the SC LPF output with a) G_1 in the noise-shaping loop and b) G_4 in the noise-shaping loop.



(a)

(b)

Fig. 4: Measured spectrum at the output of the SC LPF with a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.



(a)

(b)

Fig. 5: Simulated spectrum of the SC BPF output with a) G_1 in the noise-shaping loop and b) G_4 in the noise-shaping loop.

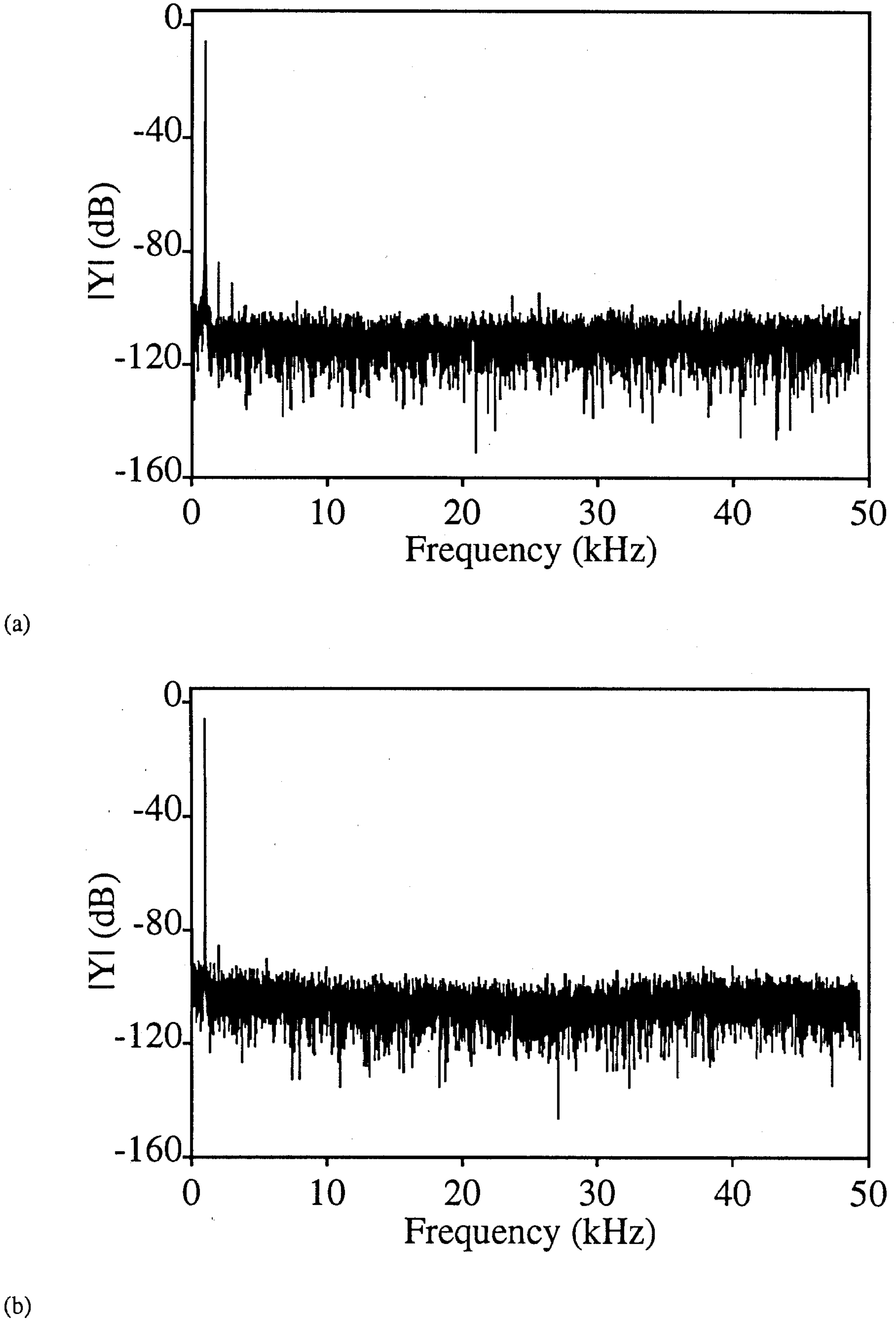


Fig. 6: Measured spectrum at the output of the SC BPF with a) G_1 in the noise-shaping clock generator and b) G_4 in the noise-shaping clock generator.

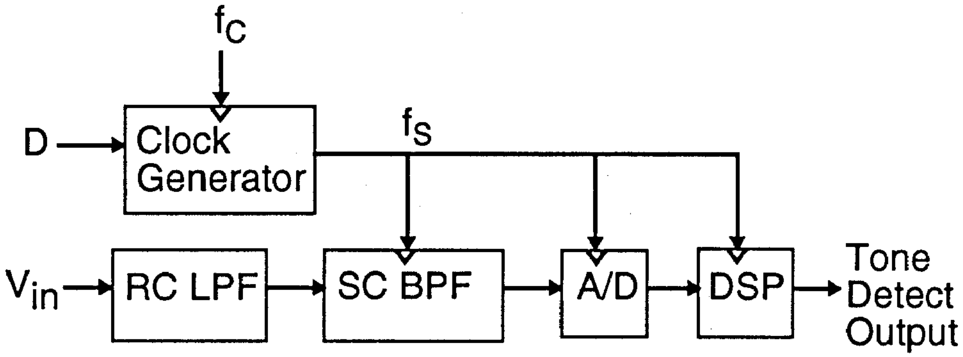


Fig. 7: DTMF tone detector using a SC BPF and a noise-shaping clock generator. (DSP is a Xilinx XC4005)

$G(z)$	$H_T(z) \cdot f_C$	Noise Gain	% of CLB
$G_0 = 2z^{-1} - 0.5z^{-2} - 0.625z^{-3}$ $- 0.125z^{-4} + 0.125z^{-5} + 0.125z^{-6}$	$(1 - z^{-1})^2 (1 + 0.5z^{-1}) \bullet$ $(1 + 0.5z^{-1} + 0.25z^{-2})$	3.5	98
$G_1 = z^{-1}$	-1	1	52
$G_2 = 2z^{-1} - z^{-2}$	$z^{-1} - 1$	3	62
$G_3 = 1.5z^{-1} - 0.5z^{-3}$	$(z^{-1} - 1)(1 + 0.5z^{-1})$	2	79
$G_4 = 2z^{-1} - 2z^{-3} + z^{-4}$	$(1 - z^{-1})^2 (1 + z^{-1})$	5	83

Table 1: Comparison of different $G(z)$

G(z)	LPF		BPF	
	Simulated SSNR(dB)	Measured SNR(dB)	Simulated SSNR(dB)	Measured SNR(dB)
G_0	84.2	61.9	86.6	63.0
G_1	68.4	61.7	82.9	63.8
G_2	84.2	60.4	99.9	61.7
G_3	80.1	60.5	88.1	62.6
G_4	87.8	58.1	98.7	59.2

Table 2: SNR at the filter outputs for different $G(z)$ with D=20.25.

DTMF frequency (Hz)	BPF f_0=0.01f_s (Hz)	BPF f_s (kHz)	D	
			octal	decimal
697	697.17	69.717	34.54	28.6875
770	770.16	77.016	31.76	25.96875
852	852.20	85.220	27.36	23.46875
941	941.18	94.118	25.20	21.25
1209	1208.7	120.87	20.43	16.546875
1336	1336.1	133.61	16.76	14.96875
1477	1476.4	147.64	15.43	13.546875
1633	1632.7	163.27	14.20	12.25

Table 3: Values for the DTMF tone detector in Fig. 7 with f_C = 2 MHz.