

# STING: A Novel Signal Analyzer

Mark J. Boyd and Kevin Klenk

Baskin School of Engineering

University of California

Santa Cruz, CA 95064

Email: mjboyd@cats.ucsc.edu and kklenk@cse.ucsc.edu

## Abstract

The Signal Timer Implementing Novel analog (STING) circuit determines the amplitude and frequency of a signal utilizing only an operational amplifier (opamp) and a Xilinx 4003 Field Programmable Gate Array (FPGA).

The circuit uses the hysteresis of the FPGA I/O pins to measure the slew rate of the opamp output. This provides instantaneous signal amplitude and frequency using minimal hardware.

## Introduction

Many successful approaches provide a digital representation of an analog signal. Low pass filters, sample and hold circuits, analog to digital converters with parallel comparators, successive approximation schemes and all-capacitive charge redistribution techniques have all been implemented in various systems.

The significant drawbacks of each of these approaches involve either the use of costly parallel hardware or excessive clock cycles to determine signal characteristics. Furthermore, the filter approach attenuates the signal and has a limited frequency range, and the charge redistribution scheme is both slow and very susceptible to noise.

The challenge in building a low-cost, high speed, robust signal analyzer is to overcome the basic limitation inherent in each of these designs: the limited slew rate of the opamp.

## Design Overview

### (1) A Unity-Gain Follower with Hysteresis.

By using a unity-gain follower and separating the feedback loop by the hysteresis of two I/O pins, we implemented a simple slew rate timer. The slew rate of the opamp output was dependent upon the difference between the two opamp

inputs; a larger difference provided a faster slew and a smaller difference provided a slower slew. Timing these slews allowed us to determine the amplitude of the signal.

STING times the fall rate of the output by counting the number of cycles a high is generated before a low is detected. The circuit continuously oscillates between .75V and 2.25V (the hysteresis limits of the Xilinx I/O pins), and this counter determines the most recent voltage as a simple function of the 1.5V fall period.

Thresholding further improves the performance. Since the opamp output never slews more than 1.5V before providing a result, an added advantage is that the the slew rate limitation is effectively minimized.

The resulting circuit has the capability to provide instantaneous amplitude and frequency results within the time required for a single slew of the opamp output. Furthermore, the base circuit (without output implementation or square wave glitch suppression) requires only one opamp, two I/O pins, and 20 combinational logic blocks (CLBs).

## Equipment Specifics

### (1) Operational Amplifier Slew Rate.

The TLC27M4 opamp had a slew rate of .5 V/ $\mu$ s. It required a power supply with  $V_{dd}$  of 3 V to 16 V (although 5 V was most easily available).

### (2) Hysteresis of Xilinx 4003 FPGA Input and Output Pins.

The FPGA accepted input of 0 V to .75 V as a logical low value, and 2.25 V to 5 V as a logical high value. The FPGA provided output of 4.2 V as a logical high, and 0.0 V as a logical low. The I/O pins could source 4 mA and sink 12 mA.

### (3) FPGA Implementation.

The Xilinx 4003 FPGA provided 200 CLBs.[4] As a practical matter, using more than 60% of these blocks would prevent the use of a 20 MHz clock without the addition of registers along critical paths.[3] Adding registers at that point would only improve the timing slightly, since addition of blocks would continue to restrict the routing. Dividing the clock using a toggle flip-flop was an option, but would have limited the range of detectable frequencies.

The Xilinx demo board provided two 7-segment displays and one 8-segment LED bar. The displays showed frequencies from 0 to 99 KHz, with the decimal points indicating values over 100KHz. The bar LED showed instantaneous

signal amplitude in .45V increments. The demo board also provided the programming port, 20MHz clock, and reset signals.

## Implementation of the Circuit

### (1) Testing the Transfer Function.

While testing the transfer function of the input and output pads of the FPGA, we experimented with the hysteresis characteristics of the I/O pins.[2] Based on the hysteresis results, we connected the feedback loop directly through the FPGA. This idea was a modification of the traditional unity-gain voltage follower.[5]

### (2) The Feedback Loop and Runs of Highs.

The result was a circuit which produced runs of 0.00 volts and 4.20 volts on the FPGA output pad. The elapsed time of a run of highs was equivalent to the time required for the opamp to slew from 2.25V to 0.75V, given the signal voltage as the positive comparator input and 4.20V as the negative comparator input to the opamp.

We noticed that for a sine wave, the length of a run of highs decreased monotonically until it was shortest (at the valley of the signal), and then increased again monotonically. We also hypothesized that a low frequency square wave would produce multiple additional frequencies since a series of almost identical runs could easily vary by a count of one (50ns).

### (3) Initial Design.

This led us to design our initial circuit. We chose an eight bit counter and registers since these would measure the longest run of highs ( $8\mu s$ ) without overflowing (since  $50ns \times 256$  is  $12.8\mu s$ ). We did not know how to solve the square wave or amplitude problem at this point, and left this for later analysis.

The hysteresis of the I/O pins would provide immunity from minor glitches near the middle and valley of the signal (for signals with a large  $\Delta$  amplitude). In this initial circuit, for smaller  $\Delta$ s and glitches occurring near the peak of a signal with a low average amplitude, the glitch would be detected as an additional frequency.

Implementing the circuit took about an hour, and the results displayed correctly, but required scaling. We used an AND4 to scale the two 8 bit counters to an approximate  $1\mu s$  timer.

### (4) Problems with Glitches, Square Waves and Unstable Output.

We then discovered that we could avoid the off-by-one problem of the square wave by averaging the previously stored average with the new value. This

Signal Shape	Minimum Amplitude (V)	Maximum Amplitude (V)	Min. Acquired Frequency (KHz)	Max. Acquired Frequency (KHz)
Sine	0.05	3.90	3	136
Sine	0.05	0.50	12	135
Sine	3.65	3.15	3	63
Square	0.05	0.50	1	140
Triangle	0.05	2.50	3	156

Table 1: STING Performance Test Results

corrected the problem while reducing the resolution of the signal detection by 25mV (since the slew rate is 25mV per 50ns).

The circuit now worked quite well determining frequencies, especially for signals close to ground with very small  $\Delta$  amplitudes. It also worked well for all shapes of waveforms, including square waves.

(5) **Instantaneous Amplitude.**

Test Signal	0	1	2	3	4	5	6	7
20 KHz	0.44	0.92	1.32	1.64	2.00	2.44	3.00	3.44
1 Hz	0.44	0.96	1.28	1.68	2.08	2.60	3.08	3.52
Timer Bit	2	3	4	5	6	7	8	9+

Table 2: LED Display Amplitude Thresholds

Laboratory measurements showed that the length of the high runs varied linearly with the voltage for signals with a frequency below 100KHz.

Within one period, a 100KHz signal varying from 1V to 4V produced a  $8\mu s$  run during and extending past the highest points of the signal and a  $2\mu s$  run during the lowest point. We were able to easily scale the signals to .45V increments (1/9 of 4V) and produce them instantaneously on the LED bar display.

(6) **Final Circuit Timing.**

The final circuit had a clock to setup maximum delay of 48.4ns, meaning that we were able to retain the original clock speed (20MHz) without further loss of resolution or clock division.

This kept the worst case minimum detectable  $\Delta V$  at 50mV. All of the behavior within the circuit remained synchronous, with no gated clocks or glitchy internal signals.[3]

## Conclusion

Figure 1 shows the final circuit design. The ADD8 component used to avoid the off-by-one error caused by slightly out of phase square waves is not essential, but included since it was implemented in the original design. The 2 decimal frequency display and 1 microsecond counter shown in the schematic can easily be replaced by a lookup table or decoder to provide instantaneous frequency, but again, we have chosen to provide the details of the implemented design.

The schematic shows clearly the essential benefits of the design. STING requires minimal logic and only a single opamp to provide an instantaneous and accurate digital representation of the given signal.

## Future Work

The STING design outlined in this paper was implemented using easily acquired and affordable parts. Using more sophisticated opamps and deep-submicron technologies can easily improve the performance by two orders of magnitude, allowing accurate detection of frequencies as high as 20MHz. Additionally, the schematic logic can easily be implemented in software or on a microcontroller or ASIC instead of using an FPGA.

## References

- [1] J. L. Hennessy, D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Inc., 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, 1996. [henn96]
- [2] Wakerly, John F. *Digital Design Principles and Practices, 2nd Edition*. Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, USA, 1994. [wake94]
- [3] Chan, Pak K. and Mourad, Samiha. *Digital Design Using Field Programmable Gate Arrays*. Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, USA, 1994. [chan94]
- [4] Xilinx, Inc. *The Programmable Logic Data Book, 1994*. Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124, USA, 1995. [xili95]
- [5] Mims, Forrest M. *The Engineer's Mini-Notebook: Op-Amp Circuits*. Radio Shack, Fort Worth, TX 76102, USA, 1995. [sims85]

If accepted, this paper will be presented by Mark J. Boyd. All appropriate organizational approvals for the publication of this paper have been obtained. If accepted the authors will prepare the final manuscript in time for inclusion in the Conference Proceedings and will present the paper at the Conference.

Figure 1: Overview of Analyzer Circuit Design.

