Five Ways to Shave Test Time

By Doug Rathburn
Keithley Instruments, Inc.

If your rack and stack instruments seem slow,
it may be the default settings you are using.

In manufacturing, any bottleneck is objectionable, but it really stands out in a test station at the end of a production line. This puts a lot of heat on the engineer responsible for test station throughput. Those who find themselves in that position may need to look for ways of shaving test time.

The author, an application engineer, has found that the problem often is a matter of the test instrument being used with default settings from the factory. There are five widely used settings that can be adjusted to speed up measurements, but they must be balanced with accuracy requirements. The five instrument settings involve:

1. signal integration period
2. auto-zero function
3. triggering functions
4. digital filtering
5. auto-ranging

Since most production test systems are automated under PC control, data traffic on an external (usually, IEEE-488) bus should also be considered. The way the system is programmed to use the external data bus has significant effects on test cycle time. (While the tips in this article are aimed at rack-and-stack instruments connected on a GPIB bus, most of them also apply to stand alone bench-top instruments in a wide variety of applications.)

Default Instrument Settings

Knowing that usability with a front panel is important, most instrument manufacturers use default settings that are user-friendly. Generally, this means that any instrumentation or data acquisition hardware configured with a front panel will run relatively slow as shipped from the factory. While an instrument's speed and accuracy specs may be publicized and well known, the manufacturer has to reckon with what the user sees on the panel. For example, if you purchase an instrument and out of the box it is reading 2000 samples per second, your eyes would not be able to distinguish the data on the panel display. This would disturb many users; some might even think the instrument is defective.
To a test engineer craving high throughput, user-friendly, default settings that

allow front panel readings can be frustrating. Fortunately, the five settings listed earlier can be used to manipulate the sample rate.

Signal Integration Period: A major component in total test time is how long it takes the analog-to-digital converter (ADC) to acquire the data. With respect to integrating ADCs, which are common in most rack-and-stack hardware, the acquisition time typically is expressed in terms of the number of power line cycles (NPLCs). The reason for this type of measurement is because line cycle noise is periodic, so integrating several samples allows it to be subtracted from the digitized data. Most instruments are shipped from the factory with NPLC set to 1.0, i.e., the test signal is sampled over the duration of one input power line (50 or 60 Hz) cycle. Since the duration of one line cycle for 60 Hz is roughly 16.67ms, the default test time can never be shorter than this.

Versatile instruments allow you to configure the NPLC setting to less than one, but this may have detrimental effects on the integrity of your test data. For virtually complete noise rejection, the measurement must be integrated over an entire line cycle period, or integer multiples thereof. If the NPLC setting were 0.1, the measurement would be ten times faster than the example above, but the instrument would extrapolate the noise out to 1.0 NPLC and includes it as part of the reading, which reduces accuracy.

This estimation of line cycle noise at sub-line cycle intervals has the effect of reducing instrument sensitivity. (See sidebar.) Most instruments come with data or calculations to determine how much resolution/sensitivity is sacrificed at different NPLC settings. If speed is your goal, set NPLC as low as possible, commensurate with minimum resolution and accuracy requirements. (See **Table 1** for speed and resolution comparisons.)

Table 1. Measurement times and resolutions for selected NPLC settings

NPLC	Time @ 60Hz	Time @ 50Hz	Typical resolution
10	166.67ms	200ms	6-1/2 digits
1	16.67ms	20ms	5-1/2 digits
0.1	1.67ms	2ms	4-1/2 digits
0.01	0.167ms	0.2ms	3-1/2 digits

Changing the NPLC setting will affect measurement resolution, but not necessarily measurement accuracy. (See sidebar.) As a rule of thumb, you lose one digit of resolution for every order-of-magnitude you reduce the NPLC setting. This reduction implies that if you have a 5-1/2 digit meter measuring at 1.0 NPLC, the resulting resolution at 0.1 NPLC would be only 4-1/2 digits as shown in **Table 1**. Generally, there is an upper bound to the resolution of the instrument at 10 NPLC, which represents the best resolution achievable with the ADC.

The difference between accuracy and resolution with respect to integration rate can be illustrated using the example of a camera. If you take a picture of a tree with the optimum combination of lens aperture and shutter speed for given lighting conditions, you will get a picture of the tree with excellent detail. If the shutter speed is increased and the lens aperture stays the same, the amount of light striking the film is reduced and the resulting image is darker. You can still see that the image is a tree, but detail is reduced. With instrumentation, getting a useful measurement depends not only on the inherent accuracy of the tool, but also on the data acquisition period, which determines the amount of detail in the measurement. As with a camera, if you shorten an instrument's integration time (data acquisition period) you will probably be able to still see the signal, but the amount of detail (resolution) may suffer.

Auto-zero Function: Changes in ambient temperature can affect ADC performance. The temperature drifts alter voltage offsets within the instrument. A high-quality ADC will correct for these voltage offsets periodically throughout the measurement process. A typical correction sequence involves three steps: measuring the input signal, measuring the ADC reference voltage and taking a zero reading with the ADC inputs shorted. Therefore, for every reading, the instrument actually takes three measurements.

Each of these measurements is made with the NPLC setting of the instrument. For a default factory setting of 1.0 NPLC, a single reading will take at least 50ms (3 x 16.67 ms) for a 60Hz line input instrument. This correction process, or auto-zero function, is incorporated into most instruments, and the typical manufacturer's default is to perform auto-zero on every reading. **Figure 1** illustrates how auto-zero affects measurement speed.

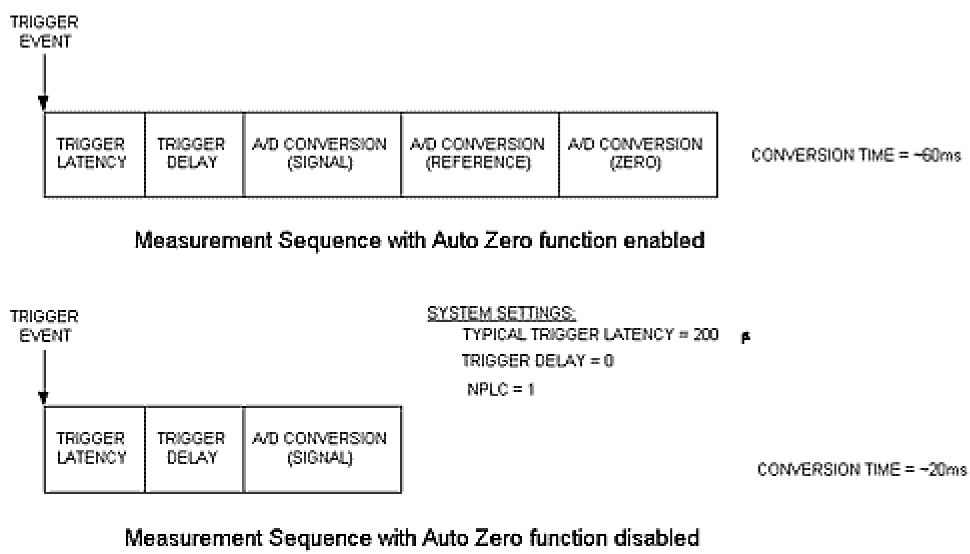

Figure 1. Effect of auto-zero on measurement time

Most instruments allow the auto-zero function to be disabled, either from the front panel or over the external data bus. Doing this increases measurement throughput by a factor of three for a given NPLC, but with a sacrifice in accuracy. By not performing an auto-zero, the baseline reference voltage will drift away from its zero value with time and changes in temperature. As this happens, the resulting readings also drift, i.e., become inaccurate.

Since temperature changes usually are slow compared to test time, it may be

possible to selectively disable the auto-zero feature of your instrument.
Production testing typically involves batch processing, which can be completed in just a few seconds. Generally, this not enough time for ambient temperature changes to affect your readings. By having the test algorithm call for auto-zero only once at the beginning of each batch, or at some other extended interval, test time is reduced without significantly affecting accuracy.

Digital Filters: A common method of dealing with random noise is to use a filtering scheme. Currently, digital techniques are used for most filters designed to stabilize noisy measurements. Analog filters exist, but are not common in instruments designed for high-speed testing environments. Typically, averaging filters are used, which have algorithms that compute either a repeating or moving average. This removes the random noise artifacts because their excursions above and below the signal level are about equal over a sufficient period of time.

A repeating filter involves filling a memory stack with readings and taking an average to yield one reading. Once the reading is computed, the stack is flushed and the process repeats for the next measurement. This type of filter is the slowest, since the stack has to be completely filled for each reading.

The moving average filter uses a first-in, first-out stack. For the first reading, the stack is filled and the samples are then averaged. For subsequent readings, the oldest sample in the stack is discarded and replaced with a new one. The stack is re-averaged, yielding a new reading. This method is the faster of the two, but since not as many samples are taken, it is slightly less stable than the repeat averaging filter.

Obviously, filtering requires much more time than a single reading, and it may also cause strange patterns in test results. If your measurements are well above the noise floor of the instrument and other random noise sources, then disabling the filter function will improve throughput. If filtering must be used, first try the moving average type for the reasons given above.

If you are testing multiple devices with filtering enabled, be aware that results from many devices could be averaged together and bad devices could be hidden. Also, if different tests are being performed (e.g. 10V and 5V tests) the results can be averaged together inadvertently (the result would be 7.5V for 10V and 5V tests). The test program algorithm should be written so it clears the filter memory stacks at appropriate times to avoid these problems.

Auto-ranging: Most digital instruments automatically choose a measurement range that provides the best resolution for the input signal. **Figure 2** shows a typical instrument algorithm that performs auto-ranging. Notice that it takes a significant amount of time to sample and settle to the correct range, which can dramatically decrease throughput. If tests are repetitive, it is best to fix the measurement range and eliminate this process all together.

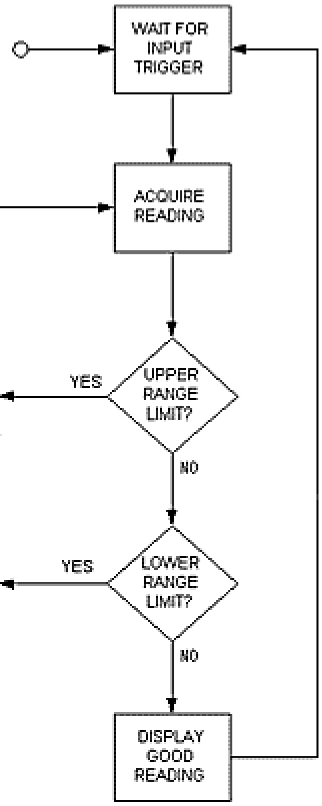

Figure 2. Example of an instrument auto-range algorithm

If you expect the test signals to fall within a certain span, the auto-range feature may be unnecessary. In QA testing, if the signal is outside a specified span or range, you know that the device under test (DUT) is bad. If your measurements are within the specified range and span, then auto-ranging and its decision time can be eliminated, thereby reducing a large part of measurement overhead.

As an example, if your signal is 2mV, but you were expecting 1.8V ±0.2V, and you used the 2V range on your DMM, it would read zero. Even though the DMM was not configured properly to measure 2mV, you know the DUT is bad since the result was not within the desired range. Auto-range is only necessary in a repetitive environment if you have to characterize (i.e. measure precisely) both good and bad devices. But since you do not know before hand if a DUT is good or bad, it is best to let the instrument choose the best range for the test. However, throughput will suffer.

External Data Bus Traffic: For most instruments, data communication over an external IEEE-488.2 bus is a programming consideration when this feature is used. The way a test program is written to control instruments with a GPIB interface could be cumbersome and reduce throughput. Many of the newer interfaces developed as possible replacements for GPIB are much faster, but good programming techniques can improve throughput, whatever the bus type.

In any event, GPIB is the de facto standard interface for rack-and-stack instrumentation today and requires careful consideration of the way test algorithms are constructed.

Most often, the IEEE-488.2 bus is used to download test data. However, some programs may be written to control large portions of the test sequence by sending SCPI commands to the instruments over the external bus. This introduces system overhead, not only due to bus traffic, but also due to PC processing.
A better alternative is to use instruments that store most of a test sequence in local memory. This allows each instrument to be configured once before testing commences and repeatedly triggered to run its test sequence. The instrument executes the test parameters with only one initialization command over the bus or from another trigger source. This cuts down on bus traffic, thereby reducing test time. (See **Figure 3**)

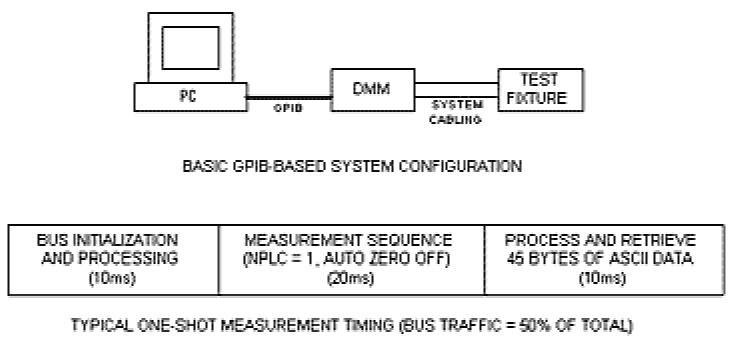

Figure 3*.* Typical system configuration and bus timing.
Sometimes you need to send actual measurement data to the PC for process control purposes or to be placed in a database for product tracking. If the external bus is used for triggering, this creates a bottleneck for test data transfers, which are already limited by how fast your PC and the GPIB interface board can send and receive data. With most production-grade instrumentation, the internal software trigger function requires only one command and does not represent a significant time bottleneck. Externally triggering an instrument with a hard-wired digital signal is even faster. Not only does it eliminate time needed to transmit and parse bus commands, but it also eliminates the instrument's software trigger latencies.
In some production tests it is not really necessary to send measurement data to the PC, only an indication of whether the DUT passed or failed. Many of today's instruments have binning (or end-of-test) capabilities built in, which facilitate this type of control. Such instruments can be programmed with pass/fail limits; if a measurement is within programmed limits, the instrument sends a digital signal pattern to the test system through a separate digital I/O port. Thus, the device handler places the DUT into the correct bin based on this signal, which does not use the IEEE-488 bus.
When the external bus is used, there are some instruments that by default transmit all the data associated with each measurement. For example, the default on a Keithley Model 2400 SourceMeter is to send ASCII data strings representing voltage, current, resistance, timestamp, and the measurement

status with each reading. Retrieving and parsing so much data can be time consuming. Instruments of this type should be programmed to send only the data segment needed for specific test purposes. For example, sending only the voltage data segment from a Keithley Model 2400 reduces the GPIB transmit time by a factor of five.

Another useful tool for reducing GPIB communication time is to send data to the PC in binary form and decode the binary string inside the PC. If a typical reading requires 15 bytes for an ASCII number, this could be reduced to only 4 bytes when transmitted in binary form. With the processing power of today's PCs, it just makes sense to do as many calculations inside the PC as possible, if that is where data is being stored.

A typical command or reading sent over the bus requires tens of milliseconds. If the actual measurement time is 10ms and it takes 10ms for the data to get to the PC, then the GPIB accounts for 50% of your total test time. While it may not be possible to eliminate GPIB time, you can control when data is actually transmitted, which may yield impressive gains in throughput. For example, when using a device handler or automated test fixture, it may be possible to send measurement data while the machine is indexing to the next part. Even with fast handlers, the index time can take 100ms, which often is sufficient time to send data to the PC, and perhaps even return some command sequences. Sending data while the system is unable to perform electrical tests eliminates the GPIB bottleneck.

A typical DC measurement time using factory defaults can be on the order of 500ms. When fully optimized for speed, many instruments can complete a measurement in one millisecond, or less. Such dramatic speed improvements immediately affect the bottom line by increasing production line throughput. Unlocking this kind of speed improvement requires a thorough understanding of instrument capabilities and how to use them to your advantage. And don't forget there usually is a tradeoff between speed and accuracy. The familiar admonition, "read the manual," is one that should be taken to heart when setting up a rack-and-stack instrument system. Finally, do not hesitate to call the instrument manufacturers; their application engineers may provide additional insights into instrument operation.

Resolution: This is the smallest portion of the signal that can be observed. In a digital measurement, basic resolution is determined by the number of bits the analog signal is chopped into by the A/D converter. For example, a 12-bit A/D converter has a resolution of one part in 212 (1/4096), or 0.0244%. Data acquisition equipment is usually specified in bits. Instruments are normally specified in digits or counts. For instance, 4 1/2 digit resolution is one part in 20000 counts because the readout can display numbers from 00000 to 19999. The term '1/2 digit' means that the most significant digit has less than a full range of 0 to 9. As a general rule, this implies that the most significant digit can have the values 0 or 1. Note that at 4-1/2 digit resolution, if the input range is bipolar, resolution is one in 40000 (2 x 20000). If the range is unipolar, the actual resolution is one in 20000.

Sensitivity: The smallest change in a measurement that can be detected. The sensitivity of an instrument is normally characterized by the top end of its lowest range divided by the instrument's resolution. It is specified in units of the pertinent measurement range, such as volts, ohms or degrees. For example, the sensitivity of a 16-bit A/D converter calibrated for 2V full scale is 2/65536, or 30.5 microvolts.

Accuracy: Accuracy is stated in terms of the closeness of agreement between a measured value and a standard value. Primary standards maintained by NIST (previously known as NBS) are usually acceptable for an absolute accuracy calibration. Relative accuracy is stated as the closeness of agreement between a measured value and a locally established reference value. This reference may or may not be traceable to an NIST standard. Frequently, accuracy is expressed as an uncertainty value in parts per million (ppm) or as a percentage of full-scale calibration value, such as 0.02% on the 10V range.

About the Author

Douglas Rathburn is a senior applications engineer at Keithley Instruments in Cleveland, OH where he is responsible for customer support, training, writing application software and technical documentation. He received BSEE and BA-Economics degrees from Case Western Reserve University in Cleveland.