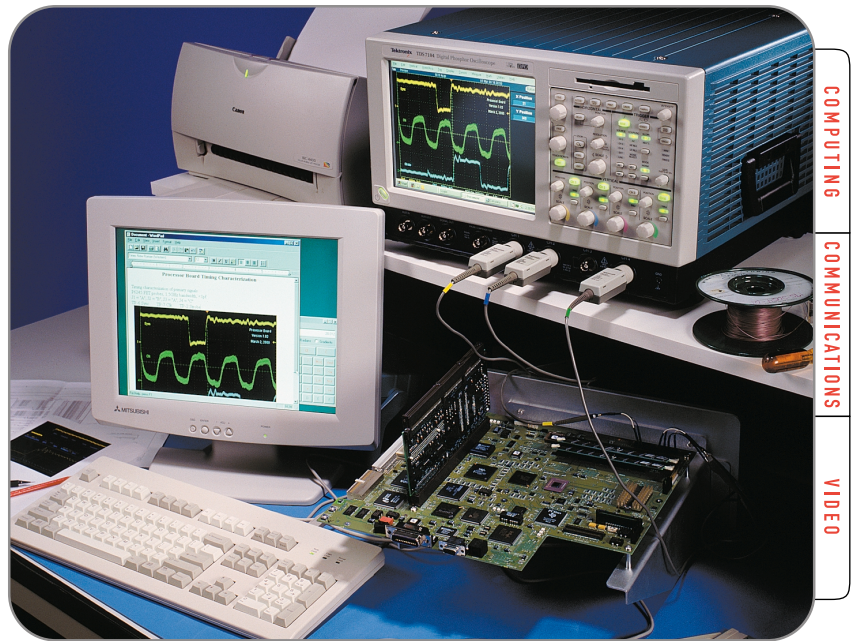


# Analyzing Clock Jitter Using Mathcad



## ▶ Easy data connectivity using the TDS 7000 Series.

Characterizing jitter, whether in the 100 picoseconds (ps) or 100 microseconds ( $\mu$ s) range, can be a time consuming effort requiring detailed analysis of waveform data. Automating this analysis can result in increased accuracy, improved efficiency and significantly improved test repeatability as compared to a manual process.

The TDS7000 Series digital phosphor oscilloscopes, with open Windows<sup>®</sup>, offers the unique advantage of enabling industry-standard analysis and data-base applications, such as Excel, Mathcad<sup>®</sup> and MATLAB<sup>®</sup>, to reside within the instrument itself<sup>1</sup>. The waveform data in the oscilloscope's acquisition memory can be exported, then imported to the analysis application, processed and displayed—all on the same platform.

This document will describe the process of using the TDS7000 Series digital phosphor oscilloscopes and Mathcad to capture signal data and then to conduct simple jitter analysis of an NRZ (non-return-to-zero) clock signal. This includes:

- ▶ How to determine the sample resolution for accurate results
- ▶ How to move waveform data from the TDS7000 to Mathcad
- ▶ How to build a simple jitter analysis worksheet
- ▶ How to use Mathcad to graph the results

<sup>1</sup> The TDS7000 supports a second Windows monitor output which will greatly enhance the oscilloscope's functionality as an analysis system. The second monitor is not required but enables you to actively use the scope while simultaneously running a separate application on the second monitor.

# Analyzing Clock Jitter Using Mathcad

► Application Note

## Overview of Jitter

### What is Jitter?

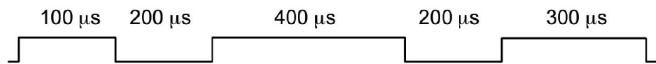
Jitter has two overlapping definitions:

1. The deviation of a signal transition from its ideal position in time, or...
2. The timing variation from transition to transition

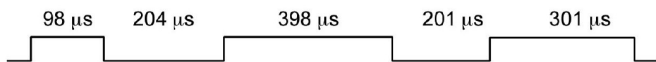
Jitter intervals often fall in the range from tens of ps to a few hundreds of ps. As clock frequencies reach the 1 GHz range, this seemingly small amount of jitter error can become a significant portion of the “timing budget,” that is, the time allotted for a series of logical operations. For example, at the standard SONET/SDH bit rate of 2.5 Gbits/sec, one unit interval (one data bit) is only 400 ps. The transmitter and receiver components consume most of this budget. Jitter can make too much of the remaining time unavailable (or at least uncertain) for necessary operations.

### Jitter Characteristics

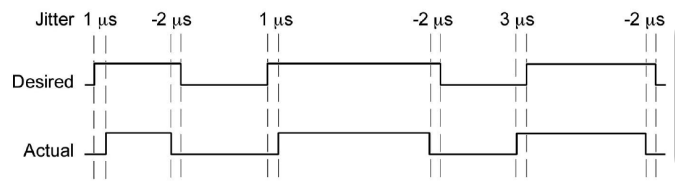
Let’s take an illustrated look at jitter. Consider the following waveform:



Every time the waveform goes positive above a specified threshold, the data is a logical one. When the waveform is below the threshold, the data is a logical zero. Now suppose that when this waveform is sent and received, its timing is affected as shown below:

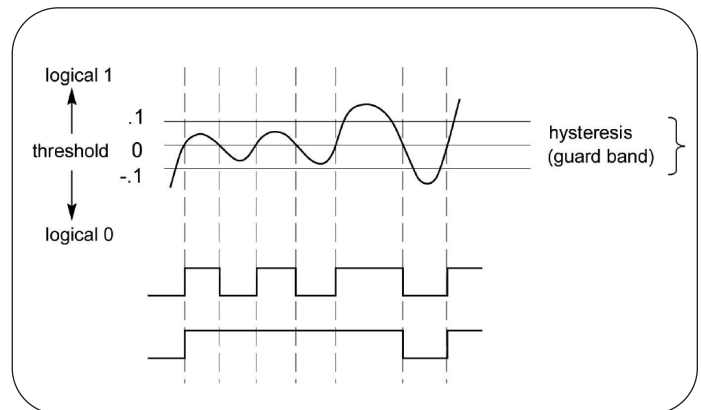


Here the signal transition deviates from its ideal position in time and it varies from transition to transition—the definition of jitter. Where the location of the edges has changed, because of factors such as noise or other error sources, the amount of change is the jitter figure, expressed in microseconds (μs), nanoseconds (ns), or picoseconds (ps) depending on the magnitude of the jitter.



### Hysteresis Effects and Guard Bands

When detecting and evaluating jitter, threshold crossings and hysteresis must be considered. It is common practice to use a “guard band” (essentially a margin that widens the ideal threshold). Figure 1 shows the threshold to be crossed and the guard band around it. To avoid registering noise as an actual change in logic level, a threshold crossing is not considered a valid transition unless the waveform first goes outside the guard band. Enforcing this rule ensures that only the true edges (bottom trace in Figure 1) are detected.



► **Figure 1:** The threshold and hysteresis in clock jitter analysis

### Jitter and Your Design

It's an acknowledged fact in the industry: jitter can affect the stability of any high-speed circuit design. Increasingly, this realization is reflected in standards, specifications and compliance guidelines.

A case in point is the USB 2.0 Specification promulgated by the USB Implementer's Forum. This document exists to ensure interoperability among USB-based products from diverse manufacturers, and, like other standards, is essential to the continued success and health of the platform. Consumers will demand USB 2.0 capability as the amount of readily accessible multimedia content grows in the coming years.

USB 2.0 has a data rate of up to 480 Mbits/sec., which is sufficient to deliver video and other bandwidth-intensive content. It is also fast enough to be susceptible to jitter. Therefore the Compliance Suite for USB 2.0 Certification includes jitter specifications. Any new platform that includes USB 2.0 (an increasing number of consumer and office products) must meet these guidelines in order to qualify for the certification.

The USB 2.0 Specifications are just one example that demonstrates how a few ps of timing inaccuracy can affect not only the technical functionality, but also the market viability of a new product.

### The Algorithm

The algorithm we'll use is as follows:

- ▶ Measure the timing of the edges
- ▶ Derive the clocks using the edges and the symbol rate
- ▶ Determine the average measured symbol rate
- ▶ Calculate the error in the average measured symbol rate
- ▶ Reconstruct the timing of the edges using the derived clock and the average measured symbol rate
- ▶ Calculate the jitter from the measured timing of the edges and the reconstructed timing
- ▶ Plot the jitter

### Terminology

The clock jitter example will make use of the following terms:

Table 1: Terminology used in the clock jitter problem

Term	Meaning
Symbol Rate	The frequency at which the communication system is sending data. (In RS232 this would be the baud rate.)
Sample Rate	The frequency at which the oscilloscope is sampling data (measured in samples/second).
Sample Interval	The time difference between samples (measured in seconds/sample). The oscilloscope user interface calls this resolution. Mathematically the <i>Sample Interval</i> = $1 / \text{Sample Rate}$ .
Threshold	The voltage used to determine if a voltage value is a logical zero (0) or a logical one (1).
Edge	A place where the waveform crosses the threshold. We are interested in the time when this occurs. Since an edge usually occurs between samples, we must use a technique called linear interpolation to find it accurately.
Hysteresis	Defines a guard band around the threshold that is used to make the algorithm less sensitive to small amounts of noise. For each edge there must be at least one point outside the guard band.

# Analyzing Clock Jitter Using Mathcad

► Application Note

## Determining the Best Sample Interval

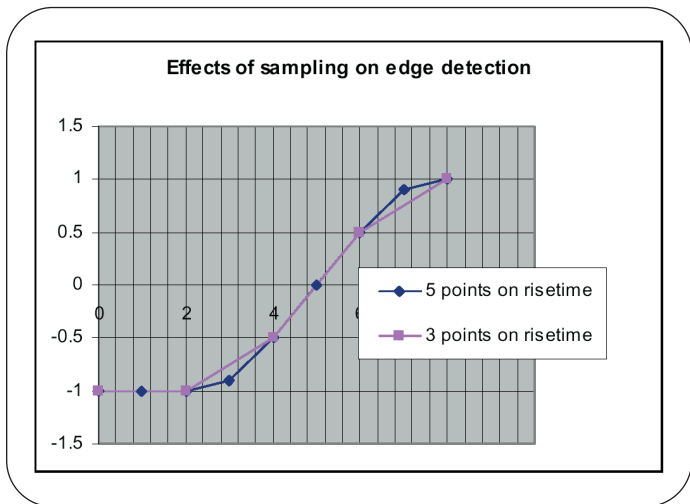
Before acquiring the waveform data, it is necessary to decide on the best sample interval to use. In trying to measure jitter, we must balance two things:

- Capturing as many edges as possible (or desired)
- Locating each edge as accurately as possible

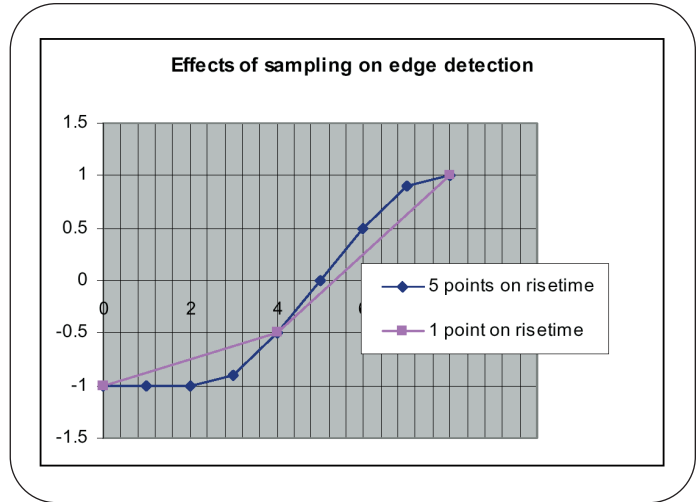
To achieve this, we want to sample the data with enough resolution to locate the edges, but not so much that we begin to limit the number of edges that we can reasonably capture. There are two tradeoffs:

- Record length versus the time it takes to process the data
- The number of edges we can capture in a given record length versus the accuracy of finding the edges

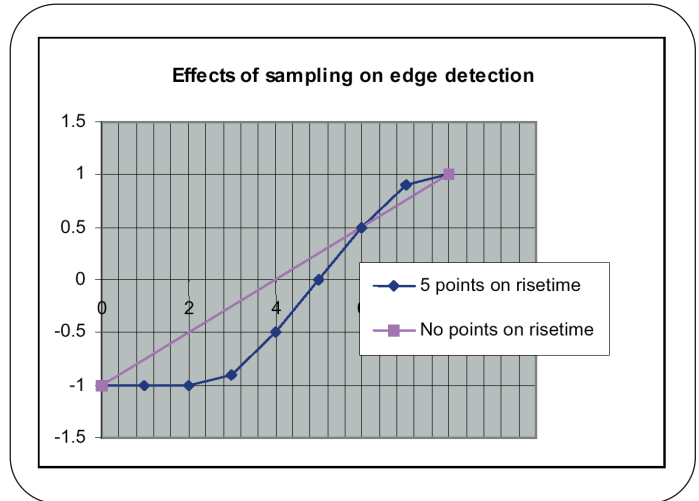
Let's explore this last tradeoff. The following graph shows an edge sampled with both three and five points on the rise time. Both pass through 0 at the horizontal value of 5. So we'll say that the edge occurs at 5.



The next graph shows the situation with five points versus a single point on the rise time. In this case, the line with a single point on the rise time passes through 0 at the horizontal value of 5.3. The data is undersampled and this undersampling has led to an error of 0.3. The result will look like jitter, but it's just an error in our measurement.



The final graph shows the situation with no points on the rise time. In this case, the line with no points on the rise time passes through 0 at a horizontal value of 4. The data is severely undersampled and this problem has led to an error of -1. Again, the result will look like jitter, but it's just an error introduced by undersampling our data in the measurement.



From the above, we can see that a sample interval (horizontal resolution) that gives us between three and five samples on the waveform rise time will allow us to accurately find the edges. Higher resolution will just limit the number of edges in whatever record length we choose.

This first step is critical in assuring that the waveform acquisition and the subsequent analysis yield accurate results, since the error induced by undersampling will be indistinguishable from the jitter present in the signal.

Having determined the sample interval (the sample rate setting to be used for the acquisition), the waveform is captured using normal oscilloscope methodology.

### Moving Waveform Data to Mathcad

The next step is to export the stored data into a file format that is Mathcad-compatible.

The TDS7000 Series oscilloscope offers a simple point-and-click interface for transferring data from the instrument to the Mathcad application.

Remember, with a TDS7000 Series oscilloscope, both the acquisition and storage tools and the Mathcad application reside within the same integrated benchtop instrument.

### Exporting the Waveform

When you export data for use in Mathcad, two files are created:

- ▶ The data file has the sampled waveform values, stored with one value per line. If, for example, you exported a waveform with the name *mydata*, the exported file would be named *mydata.dat* in Mathcad.
- ▶ The other file, the header file, is optional and contains waveform scale factors such as Record Length and Sample Interval. Here, a waveform exported with the name *mydata*, receives the name *mydata\_hdr.dat* in Mathcad. This file is generated if you select the "Include waveform scale factors" option when you export data

To export data stored in a TDS7000 oscilloscope, select menu bar mode, then select File> Export Setup. The Export Setup dialog box appears. To create the waveform data (.dat) file, select *Mathcad* as the Data Destination, *Channel 1* as the Source, *All* as the Waveform curve data range, and check the box under Waveform Detail to include waveform scale factors. With the above selections completed, all the user needs to do is simply click the Export button. Then assign a file name if you choose not to use the default file name the system provides. For the purposes of this example, the filename *jitter5k* will be used. And although Mathcad doesn't strictly require a header file (\_hdr), we will create one that includes four fields: waveform record length, sample interval, trigger location, and trigger time offset.

### Opening a Mathcad Worksheet and Entering the Data

The foregoing operations have produced a pair of files containing data that Mathcad can use. To import the waveform data from your oscilloscope into Mathcad:

- ▶ Start Mathcad, open a new worksheet and insert the file *jitter5k.dat*, preceded by the assignment operator (:=). Assign the file containing the waveform to array variable *w*.
- ▶ Similarly, insert the *jitter5k\_hdr.dat* file, which contains the sample interval as the second element in the array. Assign the header file containing the sample interval to array variable *h*.

From this point forward, Mathcad will calculate formulas and equations in real time based on the information in those variables. It's also necessary to enter several input data variables into the worksheet:

- ▶ The **symbol rate** of the clock used to generate the data
- ▶ The **threshold** used to determine if the waveform is a logic high (1) or low (0)
- ▶ The **sample interval** used to collect the data (and imported as the second element in the header array file)
- ▶ The index **k** of sampled elements in the waveform array file
- ▶ The **hysteresis** that establishes a guard band around the threshold in order to reject noise in the acquired waveform

Now Mathcad has all the raw material it needs to begin the jitter analysis. At this point, we have completed the data entry portion of the worksheet (Section "A" in the reference worksheet illustration Figure 3 on page 10).

## Analyzing Clock Jitter Using Mathcad

► Application Note

### A Procedure for Jitter Analysis

The balance of this application note covers the steps Mathcad uses to carry out the jitter analysis.

The procedure is as follows:

- Graph the waveform data
- Measure the timing of the edges
- Determine the average measured symbol rate
- Calculate the error in the average measured symbol rate
- Derive the clocks using the edges and the symbol rate
- Reconstruct the timing of the edges using the derived clock and the average measured symbol rate
- Calculate the jitter and RMS jitter values from the measured timing of the edges and the reconstructed timing
- Plot the jitter

### Graph the Waveform Data

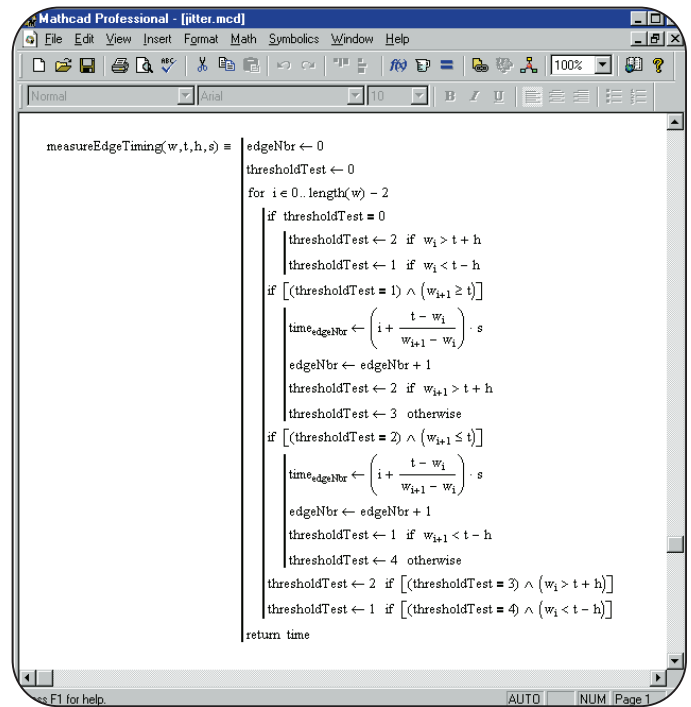
Using the entered data and variables, Mathcad can graph the input waveform very easily.

To create a waveform graph, simply click a blank space where you want the graph to appear (beneath the variable definitions), and tell Mathcad to produce an empty graph with placeholders on the x-axis and y-axis.

Use the record length as the X-axis, and the values in array “w” (the contents of the jitter5k.dat file) as the Y-axis. Mathcad immediately substitutes the data for each axis and plots it on the worksheet. Label the axes appropriately. Now the worksheet contains the graph shown in section “B” on the reference worksheet illustration (Figure 3 on page 10). Here the data is deliberately truncated to show greater detail on the waveform. The full graph consists of a 5000-point record.

### Measure the Timing of the Edges

Next we measure the timing of the edges in the raw input waveform. To do this, first we'll create a *global function*, then insert a call to that function. Since Mathcad evaluates global definitions before local definitions, we can put this function definition at the end of the worksheet. Figure 2 is a screen image of the function.



► Figure 2: The `measureEdgeTiming` function in Mathcad

The input parameters to the `measureEdgeTiming` function are:

- $w$  = the raw input waveform array
- $t$  = the threshold, whose value is used to determine whether voltage represents a logical 0 or 1
- $h$  = the hysteresis used to define the guard band
- $s$  = the sample interval used to collect the samples in the waveform array

Our initial task is to find the first edge outside the guard band.

The FOR loop checks input data samples in the waveform  $w$  to see if they are above, below, or in the guard band defined by the hysteresis  $h$  and threshold  $t$ . Once we're outside the guard band, we can begin searching for a threshold crossing that is a valid edge.

The IF clause that begins with "if [(thresholdTest = 1)]" tests for a sample that is greater than or equal to the threshold, denoting a positive edge. If the condition is met, we interpolate between two adjacent samples to locate the edge. Next a test is performed to see if the second sample is above the guard band.

The IF clause that begins "if [(thresholdTest = 2)]" tests for a sample that is less than or equal to the threshold. This is a negative edge, denoting a negative edge. If the condition is met, we interpolate between the two adjacent samples to locate the edge. Next a test is performed to see if the second sample is below the guard band.

The next statement checks to see if we are above the guard band again before searching for a negative edge. This condition occurs when we find a positive crossing and the value of the second sample is less than the value of threshold + hysteresis.

The next statement checks to see if we are below the guard band again before searching for a positive edge. This condition occurs when we find a negative crossing and the value of the second sample is greater than the value of threshold - hysteresis.

Once we have found all the edges in the waveform, we exit the FOR loop and the function returns the time array. When called, the `measureEdgeTiming` function assigns the returned result to the `measuredTime` array.

## Derive the Clocks Between Edges

There is always  $n$  number of clocks between each pair of edges. By subtracting the measured edge times between adjacent edges and then multiplying that value by the supplied symbol rate, we get a non-integer value of the number of clocks between edges. When we round the number to the nearest integer, we get a value that we then add to the clock array. The number stored in this array is the total number of clocks since the first edge.

## Calculate the Average Measured Symbol Rate

Now we can find the best fit of the measured edges (in the `measuredTime` array) and the derived clocks (in the `clock` array) to a straight line. To do this, we'll run a linear regression with Mathcad's slope and intercept functions. The formula for a straight line is:

$$y = a + bx$$

where:

- $a$  = the intercept (the point where a line will intersect the y-axis)
- $b$  = the slope (the rate of change along the line)

We want a formula that will give us the reconstructed time of an edge if we know the clock number. So we use:

- $y$  = the time of an edge (to be stored in `reconstructedTime`)
- $x$  = the clock number of an edge (in the `clock` array)

Mathcad's intercept function returns the value  $a$ . Mathcad's slope function returns  $b$ . The average measured symbol rate is calculated as  $1/b$ .

## Calculate the Symbol Clock Error Rate

With the average measured symbol rate, we can now calculate the symbol clock error rate. We arrive at this result by subtracting the supplied symbol rate from the average measured symbol rate and dividing the difference by the symbol rate. To calculate this field, use the following formula:

$$\text{measuredSymbolRateError} = \frac{(\text{measuredAverageSymbolRate} - \text{symbolRate})}{\text{symbolRate}}$$

### Reconstruct the Timing

The time of the first edge ( $\text{reconstructedTime}_0$ ) is the intercept of the best-fit line to the y-axis. To reconstruct the timing of all the edges (in  $\text{reconstructedTime}_k$ ), we apply the formula  $y = a + bx$  using the intercept  $a$  and slope  $b$  that we previously calculated.

### Calculate the Jitter

Next we solve for the jitter, calculated by finding the difference in timing between the reconstructed edge timing and the measured edge timing. We also display a related field, the measured symbol rate error calculated earlier, since excessive jitter can lead to an unacceptable error rate. To calculate these fields, use the following formulae:

$$\begin{aligned} \text{jitter}_k &:= \text{reconstructedTime}_k - \text{measuredTime}_k \\ \text{measuredSymbolRateError} &= 3.719 \times 10^{-5} \end{aligned}$$

### Calculate the RMS Jitter

To calculate the RMS (root mean square) jitter, use the following formula:

$$\begin{aligned} \text{RMSjitter} &:= \sqrt{\frac{\sum \text{jitter}^2}{\text{length}(\text{jitter})}} \\ \text{RMSjitter} &= 1.108 \times 10^{-6} \end{aligned}$$

All of the formulae and array definitions discussed above are shown in section “C” of the reference worksheet Figure 3 on page 9.

### Graph the Jitter Solution

Now we can graph our results, using the same procedure outlined earlier for the waveform graph. Use  $\text{reconstructedTime}_k$  as the range variable to be graphed on the x-axis and use  $\text{jitter}_k$  as the range variable to be graphed on the y-axis. The result is shown in section “D” of the reference worksheet illustration Figure 3 on page 10.

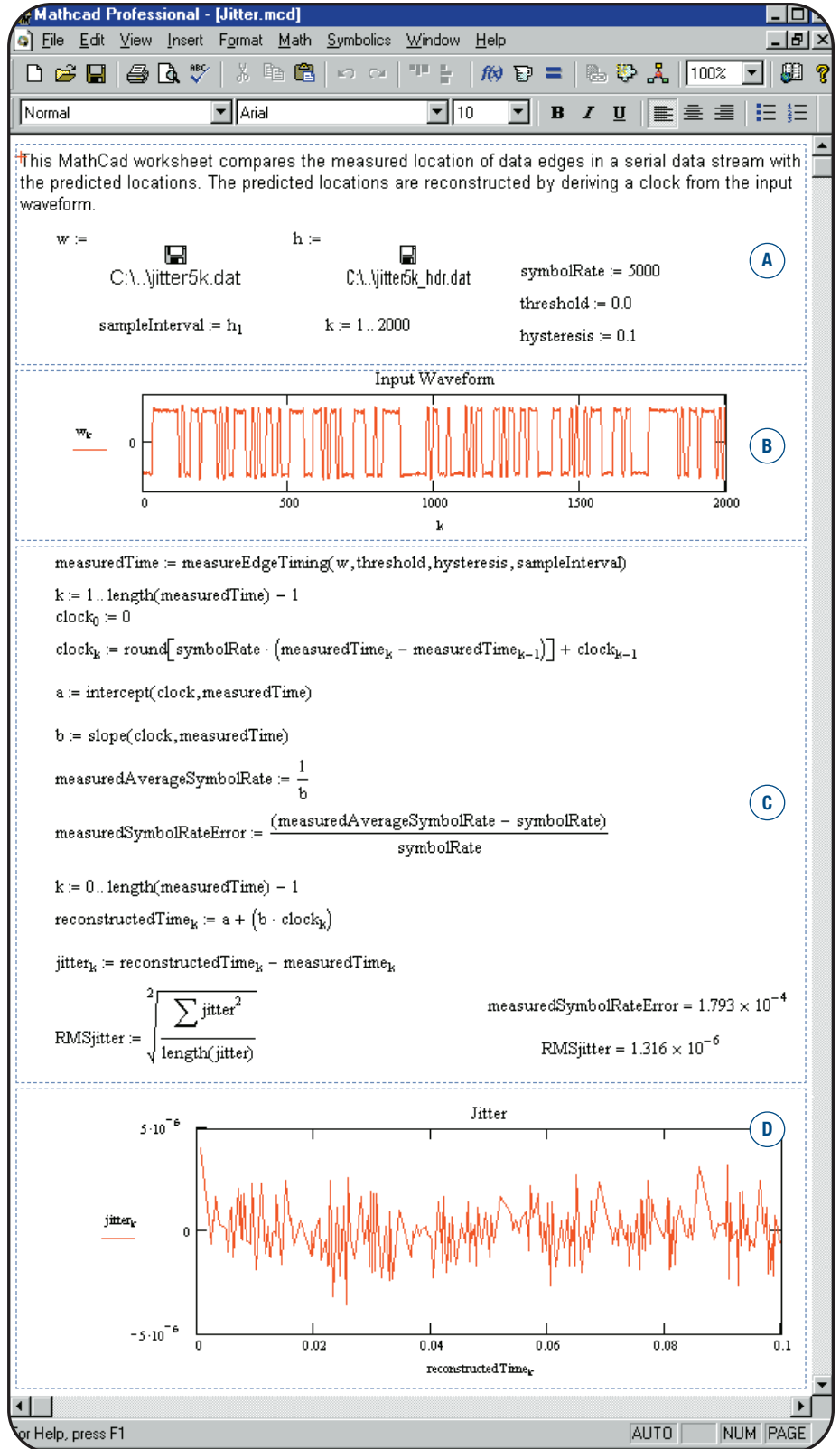
### Conclusion:

In this application note we have seen how the TDS7000 Series DPO and Mathcad work together to analyze waveform jitter. With its ability to run Mathcad on the same self-contained platform as the waveform acquisition tools, the TDS7000 is the tool of choice for critical jitter acquisition and characterization.



## Analyzing Clock Jitter Using Mathcad

► Application Note

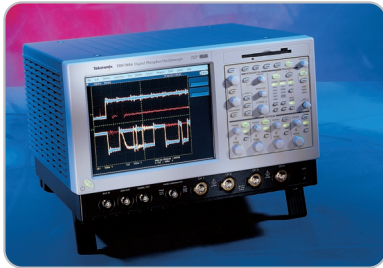


<sup>1</sup> Figure 3 is an example of a Mathcad Reference Worksheet that contains the data discussed throughout this document.

► **Figure 3:** Mathcad Reference Worksheet

## Analyzing Clock Jitter Using Mathcad

► Application Note



► TDS7054



► TDS7104



► TDS7404

[www.tektronix.com](http://www.tektronix.com)

ASEAN Countries (65) 356-3900

Australia & New Zealand 61 (2) 9888-0100

Austria, Central Eastern Europe,

Greece, Turkey, Malta & Cyprus +43 2236 8092 0

Belgium +32 (2) 715 89 70

Brazil & South America 55 (11) 3741-8360

Canada 1 (800) 661-5625

Denmark +45 (44) 850 700

Finland +358 (9) 4783 400

France & North Africa +33 1 69 86 81 81

Germany +49 (221) 94 77 400

Hong Kong (852) 2585-6688

India (91) 80-2275577

Italy +39 (2) 25086 501

Japan (Sony/Tektronix Corporation) 81 (3) 3448-3111

Mexico, Central America & Caribbean 52 (5) 666-6333

The Netherlands +31 23 56 95555

Norway +47 22 07 07 00

People's Republic of China 86 (10) 6235 1230

Republic of Korea 82 (2) 528-5299

South Africa (27 11) 651-5222

Spain & Portugal +34 91 372 6000

Sweden +46 8 477 65 00

Switzerland +41 (41) 729 36 40

Taiwan 886 (2) 2722-9622

United Kingdom & Eire +44 (0)1344 392000

USA 1 (800) 426-2200

For other areas, contact: Tektronix, Inc. at 1 (503) 627-6877



Copyright © 2001, Tektronix, Inc. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX and TEK are registered trademarks of Tektronix, Inc. All other trade names referenced are the service marks, trademarks or registered trademarks of their respective companies.

01/01 HMMH/PG 55W-14548-0

## For more information

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology.

Please visit "Resources For You" on our Web site at [www.tektronix.com](http://www.tektronix.com)