Application Note

Debugging a Mixed Signal Design with a Tektronix Mixed Signal Oscilloscope

Introduction

Today s embedded design engineer is faced with the challenge of ever-increasing system complexity. A typical embedded design may incorporate various analog signals, high- and low-speed serial digital communication, and microprocessor buses, just to name a few. Serial protocols such as I^2C and SPI are frequently used for chip-to-chip communication, but cannot replace parallel buses for all applications.

Tektronix®

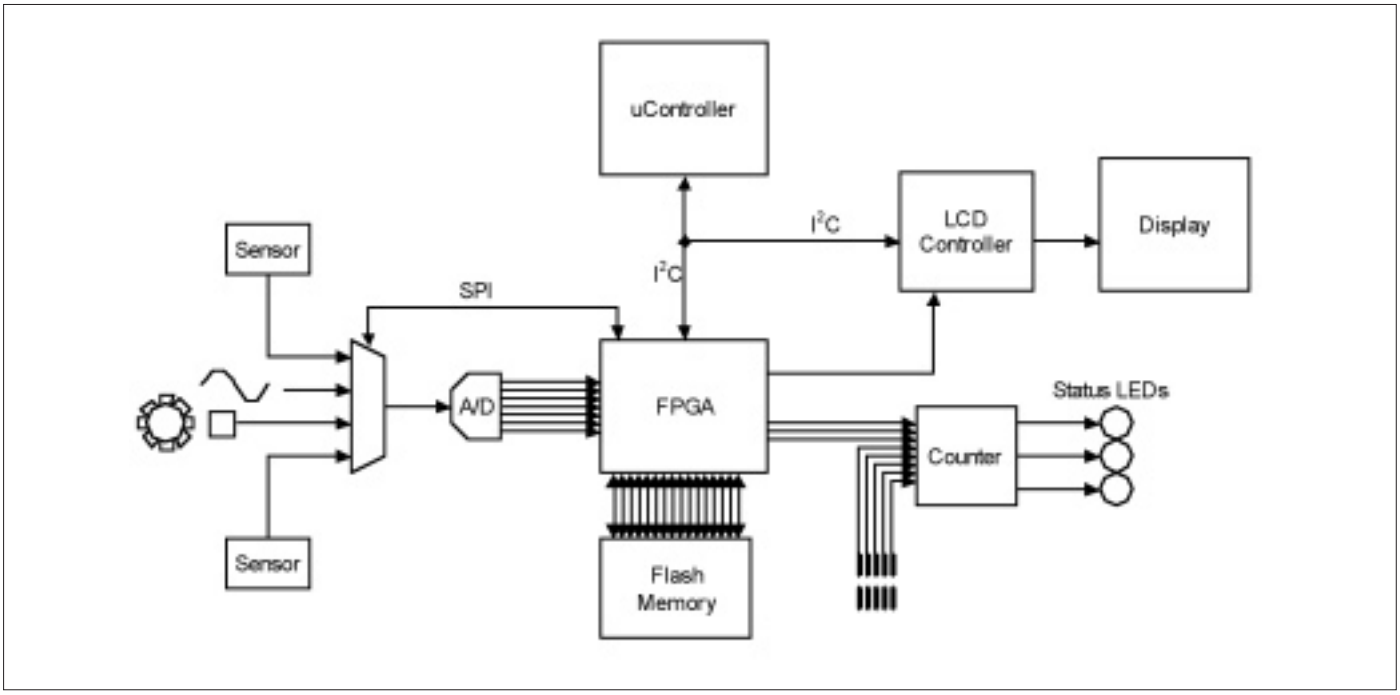

Figure 1. Simplified Acquisition / Instrumentation System.

Microprocessors, FPGAs, Analog-to-Digital Converters (ADC), and Digital-to-Analog Converters (DAC) are all examples of ICs that present unique measurement challenges in today s embedded designs. The engineer may need to decode a SPI bus between two ICs while observing the input and output of an ADC at the same time, on the same system board. An example of such a mixed-signal system is shown in Figure 1.

Debugging the hardware shown in Figure 1 is a difficult and somewhat daunting task to the engineer who is armed with his favorite 4-channel oscilloscope. Many engineers feel comfortable with their oscilloscope, and in the interest of saving time, may choose to acquire 3 or 4 oscilloscopes in order to probe multiple signals at once. Logic analyzers provide the ability to probe many digital signals, but the complexity of the debug task may not merit the setup and learning curve required to use the logic analyzer. Fortunately for engineers faced with this task, the Tektronix MSO4000 and MSO2000 Series Mixed Signal Oscilloscopes (MSO) can address their needs. The Tektronix MSO Series combines the basic functionality of a 16-channel logic analyzer with the trusted performance of a 4-channel Tektronix oscilloscope.

This application note demonstrates the industry-leading capabilities of the Tektronix MSO4000 and MSO2000 Series by walking through the debug of a mixed-signal embedded design.

Using the MSO Series to Debug Multiple Serial Protocols Simultaneously

Embedded design engineers commonly use serial protocols such as I^2C and SPI to simplify communication between system blocks on a circuit board. While these serial protocols can reduce wiring complexity, the ability to debug their implementation has been cumbersome with traditional oscilloscopes. Designers are typically forced to decode the acquired serial data by hand, or export the data from the oscilloscope for post-processing and decoding. Having the oscilloscope decode the serial data can save an embedded design engineer countless hours of debugging by allowing the engineer to see the effects of both the hardware and the software in real time.

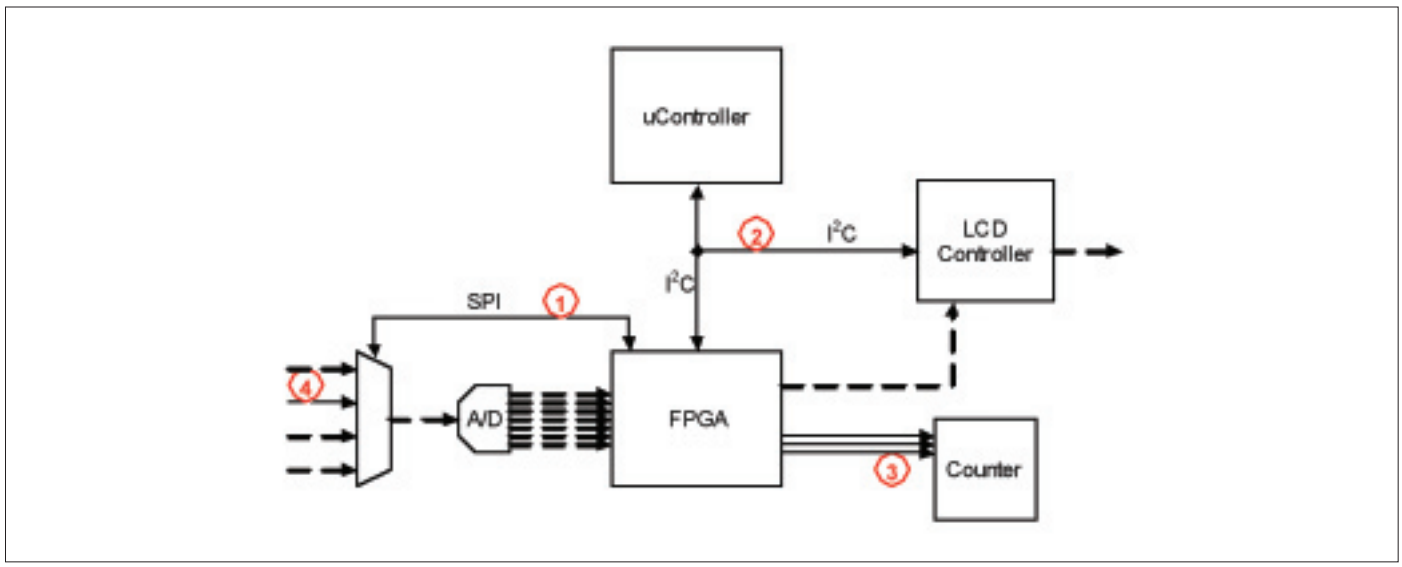

Figure 2. Subset of acquisition system with probe points.

Although DPO Series oscilloscopes can provide up to four channels for probing serial data, several of the common serial protocols require three wires or more. Engineers often need the ability to decode and display multiple serial buses at the same time, and observe their timing correlation. The Tektronix MSO Series combines the serial trigger and decode power of the DPO Series with 16 additional digital channels. In addition to I^2C, SPI, CAN, LIN, and RS-232, the MSO Series supports triggering and decoding parallel buses. The MSO4000 Series also supports triggering and decoding FlexRay buses. With the MSO Series, engineers can probe and decode multiple serial buses along with custom parallel buses, all at the same time. In the following example, the MSO Series was used to debug a complex, multi-chip communication error for the embedded design shown in Figure 1.

During the initial debug of the system (shown in Figure 1), the system occasionally encountered a condition where the status LEDs on the circuit board indicated a failure. The ambiguous error reported by the status LEDs left the system engineer unsure if the problem was hardware or software related. A similar error had previously been the result of poor signal quality at the input of the analog MUX, but the hardware that caused the signal fidelity problem had been successfully replaced. Since the system engineer was suspicious that the error could be coming from a source other than the MUX input, he decided to get a broad system view by probing the analog input to the MUX as well as several digital buses. With 4 analog and 16 digital channels available for debugging, the MSO Series was connected to the signals labeled 1-4 in Figure 2.

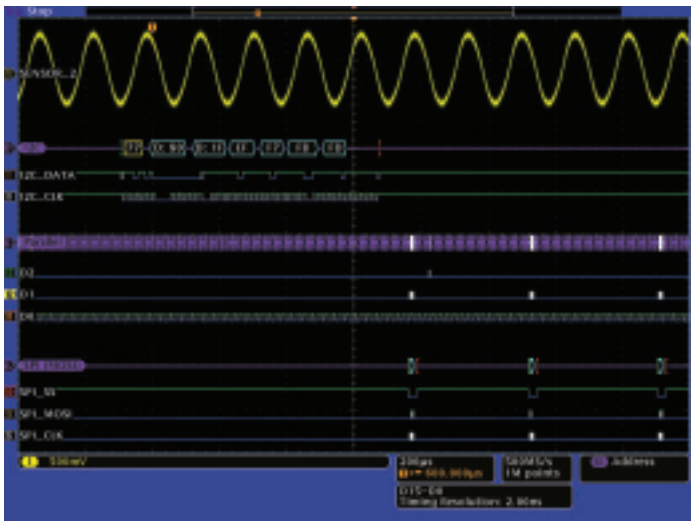

Figure 3. MSO4000 displaying I²C, SPI, and parallel buses with CH1 analog.

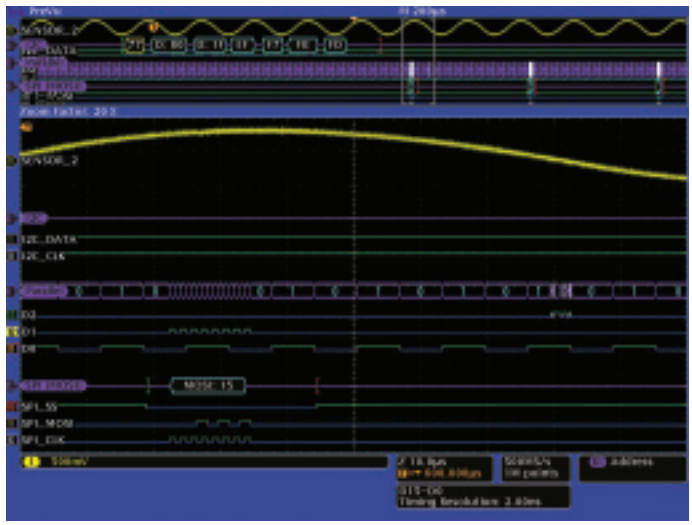

Figure 4. Wave Inspector® is used to zoom in and show packet detail.

Figure 3 shows a screenshot from the MSO Series probing the SPI bus (1), I²C bus (2), 3-bit parallel bus (3), and analog input (4) at the same time. Since the error could be isolated to a particular subroutine, the MSO Series was configured to take a single acquisition that triggered on particular I²C activity. Setting the record length to 1M points guaranteed that all the useful information around the event on the I²C bus would be accurately captured. The engineer ran the subroutine and quickly referred to the MSO Series to see what had happened in the system. The clean analog waveform at the input of the MUX shown on CH1 confirmed the engineer s suspicion that the hardware had been fixed, and the error was occurring elsewhere. The MSO Series triggered on and decoded the I²C data being written from the microprocessor. The engineer noticed activity on the SPI bus and the displayed signals labeled D1 and D2 shortly after the I²C data was transmitted. The activity on these buses was suspicious because the executed function was supposed to primarily involve the LCD controller. Because the MSO Series had already decoded the value of the I²C data, the engineer could see that the microprocessor had written I²C data to address 0x77. Address 0x77 is the address of the FPGA, but the subroutine was supposed to write data to address 0x76 instead, which is the address of the LCD controller.

Figure 4 shows the same acquisition while using Wave Inspector® Navigation and Search to zoom in on the details of the SPI and parallel buses. The SPI data is decoded on screen as a write from the master (FPGA) to the slave (MUX) with a data value of 0x15. This SPI command instructed the input MUX to change the input it was using for the signal path. This unexpected change in the input signal caused the FPGA to send an error code to the status LEDs on the parallel bus. The activity on signal D2 indicating the error code and the decoding of the parallel bus can be observed in Figure 4 as well.

The embedded design engineer was quickly able to determine that a software bug had caused the malfunction in the system because the MSO Series was able to view and decode all the signals of interest simultaneously. The software programmer had mistakenly written I²C data from the microcontroller to the FPGA when the packet was intended for the LCD controller.

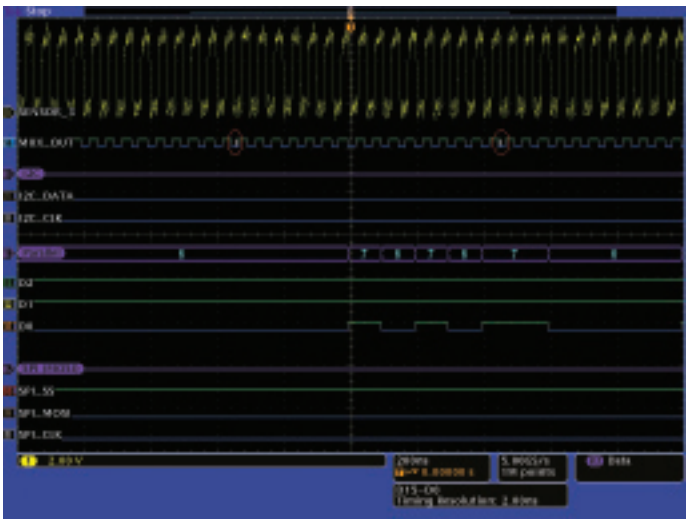

Figure 5. The white edges on MUX_OUT indicate more details are present.

Next Generation Digital Waveform Display Reveals a Problem

After modifying the system software to correct the addressing bug described in the previous section, the embedded design engineer continued to exercise more functionality of the system. While doing this, he noticed occasional errors on the status LEDs. Unlike the error described in the previous section, the engineer was unsure how to recreate the errors he was seeing. The errors appeared to be random in nature, and could not be isolated to a particular function or subroutine of the system.

The embedded design engineer was puzzled by the random nature of the errors, and was not sure where to begin looking for the source. One option to find the error source was to use the oscilloscope to randomly probe around the system in hopes of capturing the random event. Although the engineer had used this method in the past, he knew that a properly configured MSO Series connected to all the signals of interest could find the error in considerably less time. Most of the probe points from the previous section were still connected to the MSO Series. The CH1 probe was moved to the active MUX input, which is a digital signal from sensor 3. In addition to these 4 probe points, a digital channel was used to probe the MUX output.

The FPGA transmits value 0x7 over the 3-bit parallel bus to indicate an error has occurred. In order to isolate the problem, the MSO Series was configured to capture a single acquisition with the trigger event set to parallel bus value 0x7. Figure 5 shows the resulting acquisition. In this case, the parallel bus decoding and trigger saved time and confusion because the error condition was easily isolated. The 1M record length used during this acquisition allowed the engineer to observe key details of the signals before and after the trigger event.

At first glance, the signals shown in Figure 5 appear to be well behaved, but the system engineer quickly identified two edge transitions on the MUX_OUT signal that appeared unique. The white transitions on the MUX_OUT signal shown in Figure 5 are an indication to the user that more information exists within those portions of the signal. The MSO Series multiple edge detection capability highlights areas of the waveform where zooming may reveal higher frequency digital pulses. Figure 6 reveals the details behind the first white transition when Wave Inspector® is used to zoom in on the waveform details. The portion of the signal that was drawn as the white transition in Figure 5 is actually a glitch on the MUX_OUT signal.

Figure 6. Wave Inspector® reveals a glitch on the MUX_OUT signal.

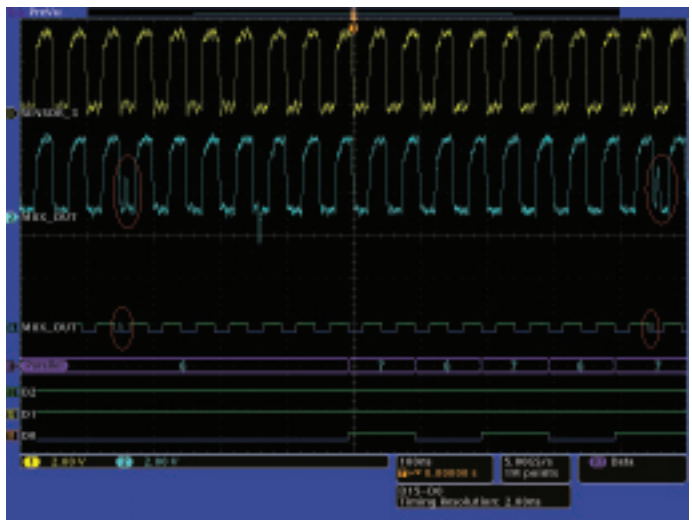

Figure 7. CH2 shows more detail for MUX_OUT.

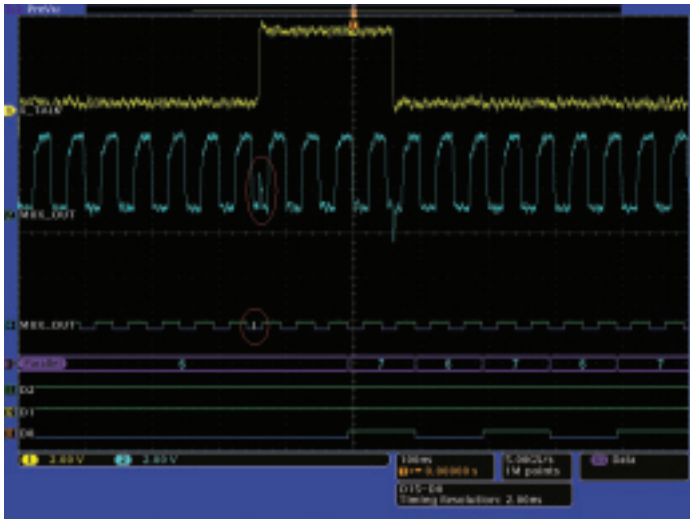

Figure 8. CH1 shows the source of crosstalk into the MUX_OUT signal.

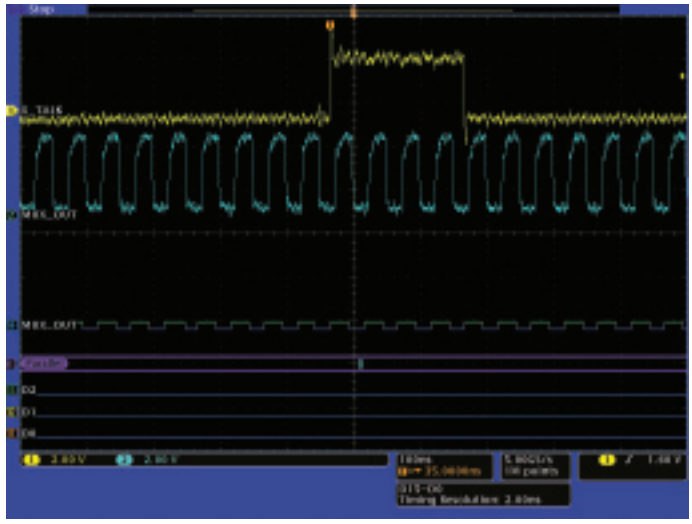

Figure 9. The PCB modification removes the crosstalk.

Figure 6 shows the timing correlation between the analog and digital channels in the MSO Series. The input to the MUX is displayed on CH1 (SENSOR_3) while the MUX output is observed on digital channel D14 (MUX_OUT). The engineer noticed that although the MUX output had a glitch, the input to the MUX appeared glitch free. After identifying the glitch with the digital channel, the engineer decided to take a closer look by connecting CH2 to the MUX output. Figure 7 shows the resulting acquisition with the MSO Series still configured to trigger on the parallel bus value 0x7. The waveforms for the SPI and I^2C buses have been turned off in Figure 7 to focus on the primary signals of interest. Using the analog probes on the MUX input and output revealed that glitches present on the output are not present on the input signal. Figure 7 shows that a glitch on the MUX_OUT signal appeared a short time before the FPGA sent the error code. The timing relationship between these two signals indicated that the glitch could be the problem the engineer had observed. The engineer repeated the acquisition several times with the same configuration, and observed behavior similar to Figure 7 in every case.

After studying the screenshots from the MSO Series, the embedded design engineer was suspicious that crosstalk could be the source of the glitch on the MUX_OUT signal. None of the signals he monitored in Figure 5 were identified as the source of crosstalk. While examining the circuit board layout in more detail, the engineer found a via located next to the MUX_OUT trace on the printed circuit board (PCB). The engineer probed the via on the PCB with CH1, and waited for another trigger from the parallel bus. The resulting screenshot is shown in Figure 8. Figure 8 shows that a low to high transition for the signal captured on CH1 directly correlates in time with a positive glitch on the MUX_OUT signal. Correspondingly, the high to low transition directly relates to a negative glitch on the MUX_OUT signal.

After spending some time to reroute the offending signal on the circuit board, the engineer configured the MSO Series to trigger on CH1. Figure 9 shows that the MSO Series triggered on the CH1 transition, but does not show a glitch on the MUX_OUT signal. Since the MUX_OUT signal did not have a glitch present, the parallel bus did not produce an error condition. The circuit board modification removed the crosstalk, and allowed the embedded design engineer to complete his system evaluation.

Summary

As this application note has shown, the MSO Series — MSO4000 and MSO2000 — are extremely powerful tools for engineers developing and debugging embedded designs. The MSO Series combines 16 time-correlated digital channels with the trusted performance and intuitive interface of a 4- channel Tektronix oscilloscope. Rather than searching for multiple oscilloscopes or learning how to operate a logic analyzer, engineers will now reach for the MSO Series.

The MSO Series ability to trigger on and decode both parallel buses and serial standards such as I^2C, SPI, CAN, LIN, and RS-232 is invaluable to engineers who evaluate the complex interaction of hardware and software in today s embedded designs.

The MSO Series offers a range of models to meet your needs and your budget:

	MSO4000 Series	MSO2000 Series
Bandwidth	1 GHz, 500 MHz, 350 MHz	200 MHz, 100 MHz
Channels	2 or 4 analog, 16 digital	2 or 4 analog, 16 digital
Record Length (All Channels)	10 M	1 M
Sample Rate (Analog)	5 GS/s*, 2.5 GS/s	1 GS/s
Sample Rate (Digital)	500 MS/s (Full Record Length) 16.5 GS/s (10 k Points Centered on The Trigger)	1 GS/s (Using Any of Channels: D7 - D0) 500 MS/s (Using Any of Channels: D15 - D8)
Color Display	10.4 in. XGA	7 in. WQVGA
Serial Bus Triggering and Analysis Application Modules	DPO4EMBD: I^2C, SPI DPO4COMP: RS-232/422/485/UART DPO4AUTO: CAN, LIN DPO4AUTOMAX: CAN, LIN, FlexRay	DPO2EMBD: I^2C, SPI DPO2COMP: RS-232/422/485/UART DPO2AUTO: CAN, LIN

* 1 GHz bandwidth models.

Contact Tektronix:

ASEAN / Australasia (65) 6356 3900
Austria +41 52 675 3777
Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
Belgium 07 81 60166
Brazil & South America (11) 40669400
Canada 1 (800) 661-5625
Central East Europe, Ukraine and the Baltics +41 52 675 3777
Central Europe & Greece +41 52 675 3777
Denmark +45 80 88 1401
Finland +41 52 675 3777
France +33 (0) 1 69 86 81 81
Germany +49 (221) 94 77 400
Hong Kong (852) 2585-6688
India (91) 80-22275577
Italy +39 (02) 25086 1
Japan 81 (3) 6714-3010
Luxembourg +44 (0) 1344 392400
Mexico, Central America & Caribbean 52 (55) 5424700
Middle East, Asia and North Africa +41 52 675 3777
The Netherlands 090 02 021797
Norway 800 16098
People's Republic of China 86 (10) 6235 1230
Poland +41 52 675 3777
Portugal 80 08 12370
Republic of Korea 82 (2) 6917-5000
Russia & CIS +7 (495) 7484900
South Africa +27 11 206 8360
Spain (+34) 901 988 054
Sweden 020 08 80371
Switzerland +41 52 675 3777
Taiwan 886 (2) 2722-9622
United Kingdom & Eire +44 (0) 1344 392400
USA 1 (800) 426-2200
For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

Updated 12 November 2007

For Further 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 **www.tektronix.com**

10/08 EA/ 3GW-20215-1

Tektronix®