

Debugging a Mixed Signal Design with a Tektronix Mixed Signal Oscilloscope

Application Note

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 I²C and SPI are frequently used for chip-to-chip communication, but cannot replace parallel buses for all applications.

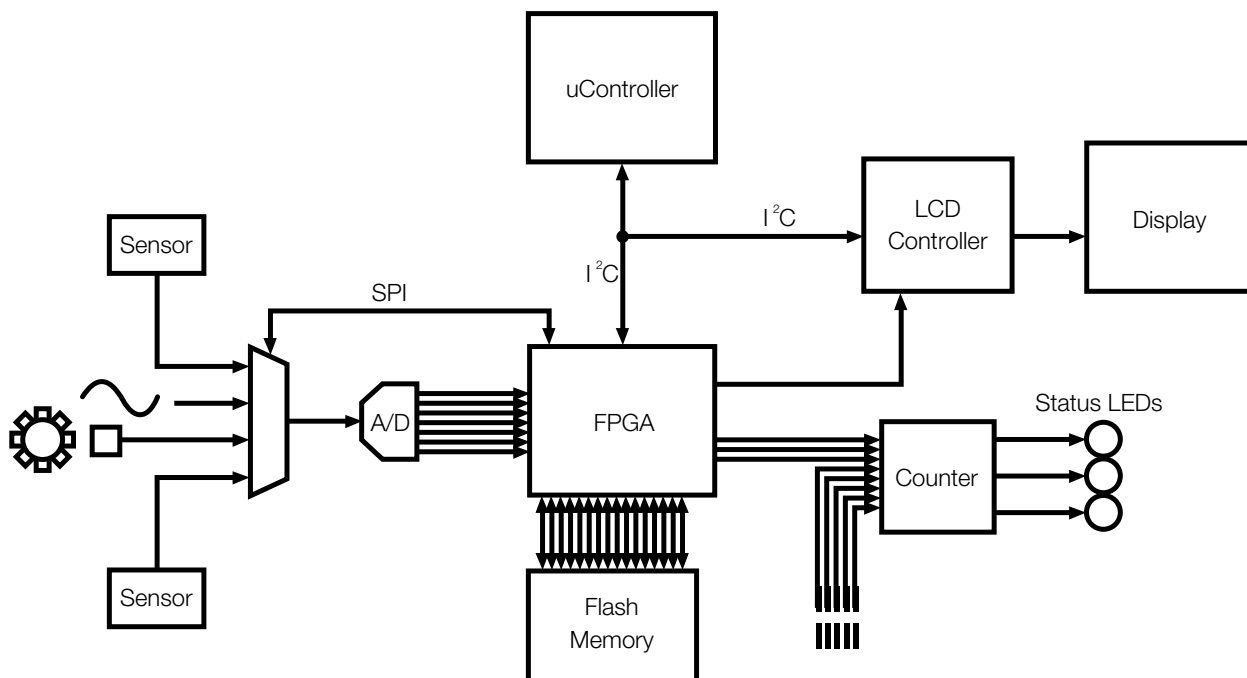


Figure 1. Mixed Signal Embedded System.

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 MSO5000, MSO4000, MSO3000 and MSO2000 Series Mixed Signal Oscilloscopes (MSO) and MDO4000 Series Mixed Domain Oscilloscopes (MDO) 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. The Tektronix MDO Series adds a dedicated RF input to the MSO Series platform to provide a built-in spectrum analyzer. This application note demonstrates the industry-leading capabilities of the Tektronix MSO5000, MDO4000, MSO4000, MSO3000 and MSO2000 Series by walking through the debug of a mixed-signal embedded design.

Using the MSO/MDO Series to Debug Multiple Serial Protocols Simultaneously

Embedded design engineers commonly use serial protocols such as I²C 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.

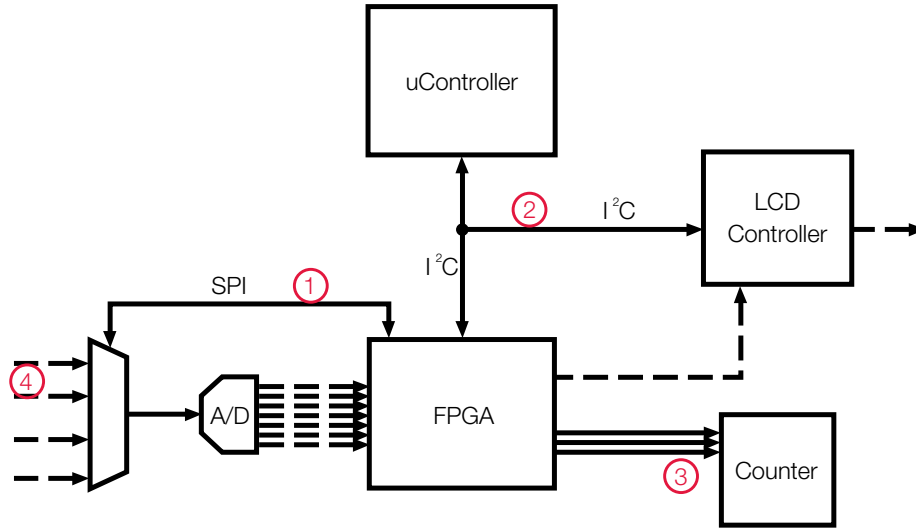


Figure 2. Subset of acquisition system with probe points.

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/MDO Series combines the serial trigger and decode power of the DPO Series with 16 additional digital channels. In addition to I²C, SPI, CAN, LIN, RS-232 and other serial bus standards, the MSO/MDO Series supports triggering and decoding parallel buses. With the MSO/MDO Series of oscilloscopes, engineers can probe and decode multiple serial buses along with custom parallel buses, all at the same time. In the following example, an MSO/MDO 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/MDO Series was connected to the signals labeled 1-4 in Figure 2.

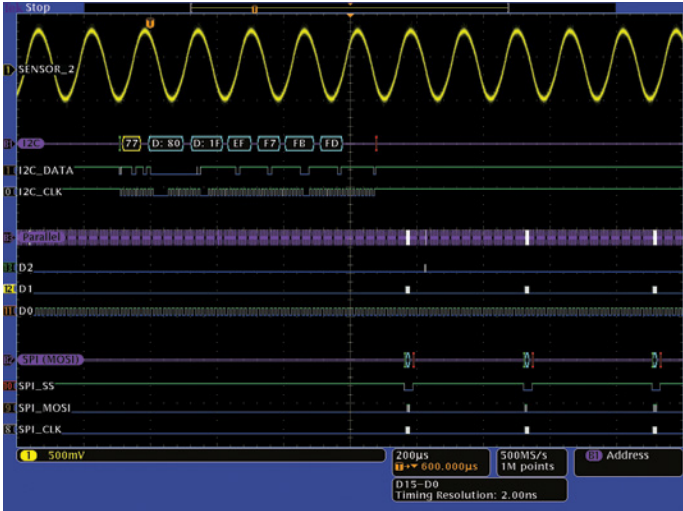


Figure 3. MSO4000 displaying I²C, SPI, and parallel buses with CH1 analog.

Figure 3 shows a screenshot from 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 oscilloscope 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/MDO 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 oscilloscope triggered on and decoded the I²C data being written from the microcontroller. 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 oscilloscope had already decoded the value of the I²C data, the engineer could see that the microcontroller had written I²C data to address

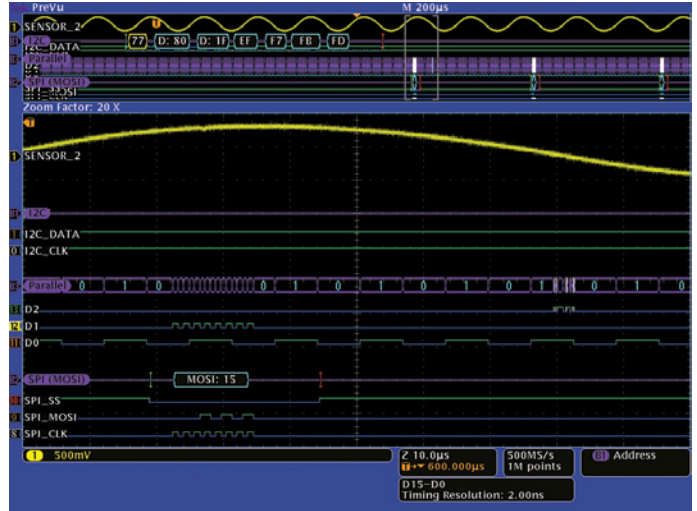


Figure 4. Wave Inspector[®] is used to zoom in and show packet detail.

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/MDO Series oscilloscope 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.

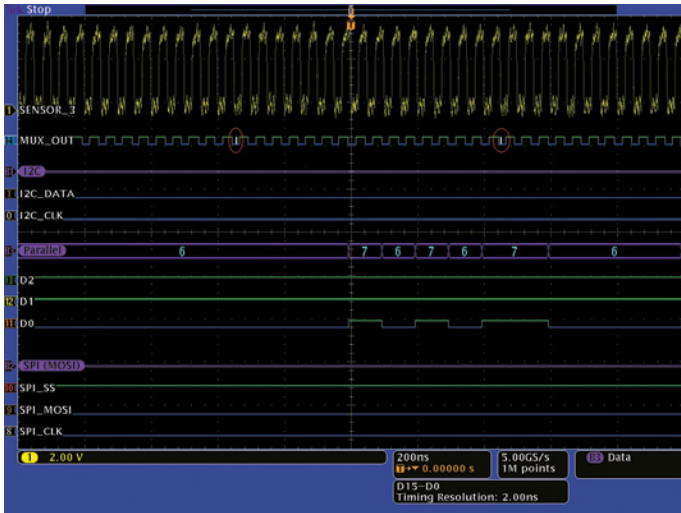


Figure 5. The white edges on MUX_OUT indicate more details are present.

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/MDO Series oscilloscope 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 oscilloscope. 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/MDO 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/MDO 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 a glitch on the MUX_OUT signal.

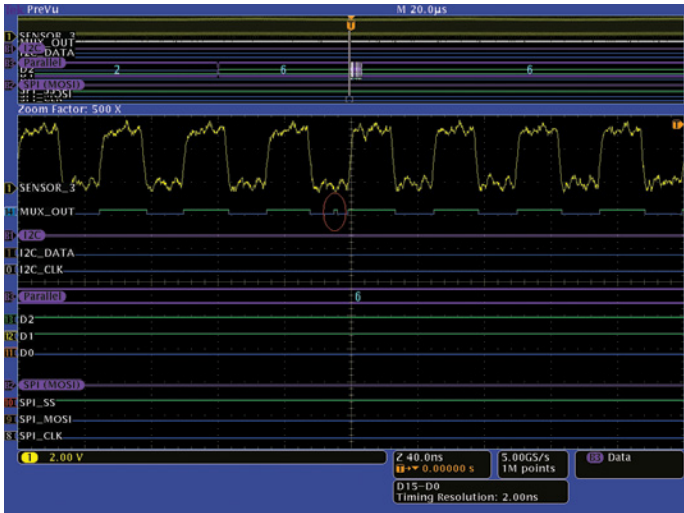


Figure 6. Wave Inspector® reveals a glitch on the MUX_OUT signal.

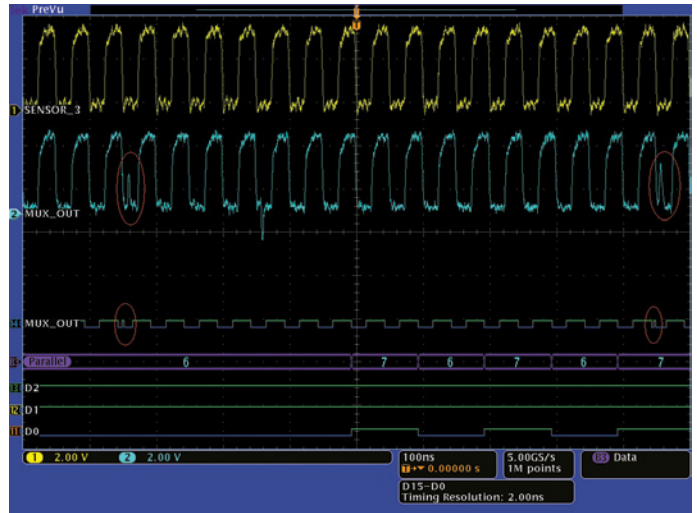


Figure 7. CH2 shows more detail for MUX_OUT.

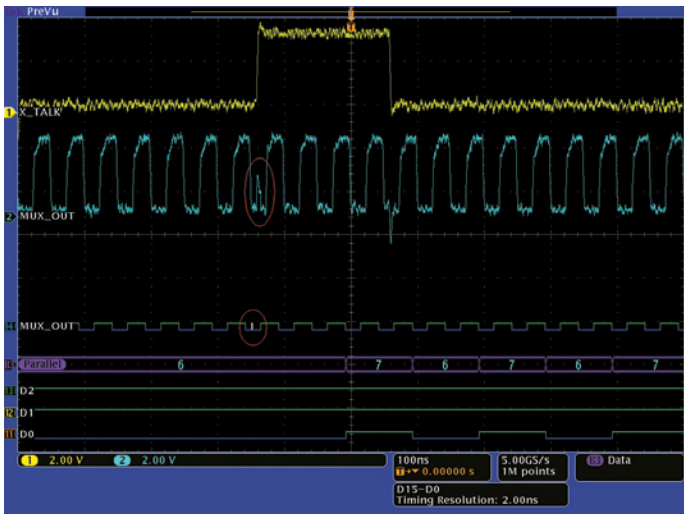


Figure 8. CH1 shows the source of crosstalk into the MUX_OUT signal.

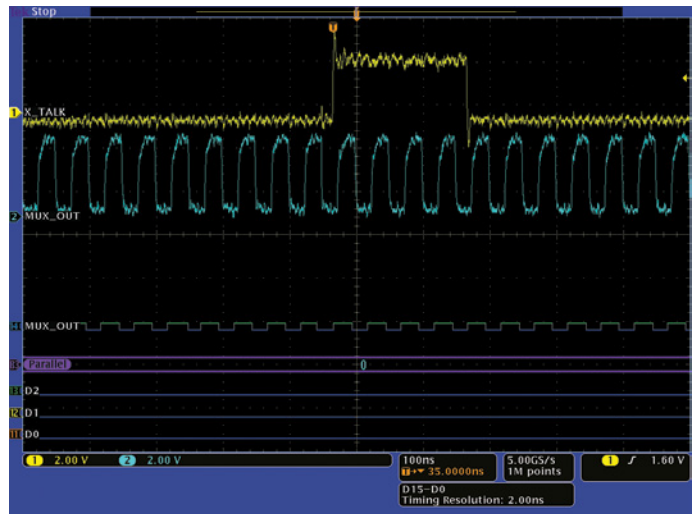


Figure 9. The PCB modification removes the crosstalk.

Figure 6 shows the timing correlation between the analog and digital channels in the MSO/MDO 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 oscilloscope still configured to trigger on the parallel bus value 0x7. The waveforms for the SPI and I²C 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, 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/MDO Series oscilloscope to trigger on CH1. Figure 9 shows that the MSO/MDO 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 – MSO5000, MSO4000, MSO3000 and MSO2000 – and MDO Series MDO4000 – 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. The MDO Series builds on the MSO series by adding a dedicated RF input; which can capture time-correlated analog, digital, and RF signals acquisitions in single instrument. Rather than searching for multiple oscilloscopes or learning how to operate a logic analyzer or spectrum analyzer, engineers will now reach for the MSO or MDO Series.

The MSO and MDO Series' ability to trigger on and decode both parallel buses and serial standards such as I²C, SPI, USB, Ethernet, CAN, LIN, FlexRay, RS-232/422/485/UART, MIL-STD-1553, and other serial standards is invaluable to engineers who evaluate the complex interaction of hardware and software in today's embedded designs.

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

	MSO5000 Series	MDO4000 Series	MSO4000B Series	MSO3000 Series	MSO2000B Series
Bandwidth	2 GHz, 1 GHz, 500 MHz, 350 MHz	1 GHz, 500 MHz, 350 MHz, 100 MHz	1 GHz, 500 MHz, 350 MHz, 100 MHz	500 MHz, 300 MHz, 100 MHz	200 MHz, 100 MHz, 70 MHz
Channels	4 analog, 16 digital	4 analog, 16 digital, 1 RF	2 or 4 analog, 16 digital	2 or 4 analog, 16 digital	2 or 4 analog, 16 digital
Record Length (All Channels)	12.5 M standard, up to 125 M optional	20 M	Up to 20 M	5 M	1 M
Sample Rate (Analog)	Up to 10 GS/s	5 GS/s, 2.5 GS/s	Up to 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)	500 MS/s (Full Record Length), 16.5 GS/s (10 k Points Centered on The Trigger)	500 MS/s (Full Record Length), 16.5 GS/s (10 k Points Centered on The Trigger)	500 MS/s (Full Record Length), 8.25 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	10.4 in. XGA	10.4 in. XGA	9 in. WVGA	7 in. WQVGA
Parallel Bus Analysis	Yes	Yes	Yes	Yes	Yes
Serial Bus Triggering and Analysis Application Modules	SR-EMBD: I ² C, SPI SR-COMP: RS-232/422/485/UART SR-USB: USB 2.0 SR-AUTO: CAN, LIN, FlexRay SR-AERO: MIL-STD-1553 SR-810B: 8b/10b Analysis SR-DPHY: MIPI D-PHY Analysis VNM: CAN, LIN decoding	DPO4EMBD: I ² C, SPI DPO4USB: USB DPO4ENET: 10BASE-T, 100BASE-TX DPO4COMP: RS-232/422/485/UART DPO4AUTO: CAN, LIN DPO4AUTOMAX: CAN, LIN, FlexRay DPO4AERO: MIL-STD-1553 DPO4AUDIO: I ² S/LJ/RJ/TDM	DPO4EMBD: I ² C, SPI DPO4USB: USB DPO4ENET: 10BASE-T, 100BASE-TX DPO4COMP: RS-232/422/485/UART DPO4AUTO: CAN, LIN DPO4AUTOMAX: CAN, LIN, FlexRay DPO4AERO: MIL-STD-1553 DPO4AUDIO: I ² S/LJ/RJ/TDM	DPO3EMBD: I ² C, SPI DPO3COMP: RS-232/422/485/UART DPO3AUTO: CAN, LIN DPO3FLEX: FlexRay DPO3AUDIO: I ² S/LJ/RJ/TDM DPO3AERO: MIL-STD-1553	DPO2EMBD: I ² C, SPI DPO2COMP: RS-232/422/485/UART DPO2AUTO: CAN, LIN
Additional Application Support	Power Analysis Limit and Mask Testing Jitter and Timing USB 2.0 Compliance Ethernet Compliance MOST Compliance Visual Triggering	Limit and Mask Testing, Power Analysis, Advanced RF Power Level Triggering, HDTV & Custom Video	Limit and Mask Testing, Power Analysis, HDTV & Custom Video	Power Analysis, HDTV & Custom Video	-

Contact Tektronix:

- ASEAN / Australasia (65) 6356 3900
- Austria* 00800 2255 4835
- Balkans, Israel, South Africa and other ISE Countries +41 52 675 3777
- Belgium* 00800 2255 4835
- Brazil +55 (11) 3759 7627
- Canada 1 (800) 833-9200
- Central East Europe 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* 00800 2255 4835
- Germany* 00800 2255 4835
- Hong Kong 400-820-5835
- India 000-800-650-1835
- Italy* 00800 2255 4835
- Japan 81 (3) 6714-3010
- Luxembourg +41 52 675 3777
- Mexico, Central/South America & Caribbean 52 (55) 56 04 50 90
- Middle East, Asia and North Africa +41 52 675 3777
- The Netherlands* 00800 2255 4835
- Norway 800 16098
- People's Republic of China 400-820-5835
- Poland +41 52 675 3777
- Portugal 80 08 12370
- Republic of Korea 001-800-8255-2835
- Russia & CIS +7 (495) 7484900
- South Africa +27 11 206 8360
- Spain* 00800 2255 4835
- Sweden* 00800 2255 4835
- Switzerland* 00800 2255 4835
- Taiwan 886 (2) 2722-9622
- United Kingdom & Ireland* 00800 2255 4835
- USA 1 (800) 833-9200

*** If the European phone number above is not accessible, please call +41 52 675 3777**

Contact List Updated 10 February 2011

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



Copyright © 2012, Tektronix. 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.

10/12 EA/WWW

3GW-20215-6

