

Supporting Serial Bus Interfaces with PXI Digital Instrumentation

Dale Johnson
Geotest-Marvin Test Systems
Irvine, CA 92614
dalej@geotestinc.com

Introduction

As the electronics industry has evolved, so too have the number of methods for transferring information between electronic components and subassemblies. Today there are many protocols and methods for communicating between components, circuit boards, subsystems, LRUs and systems. While system level communication has seen a variety of both parallel and serial communication standards evolve; board-level, and especially component level communication has primarily adopted serial data transfer methods. Using a generic digital test instrument with sufficient flexibility and control can provide the needed functionality to emulate and control common serial bus protocols. This paper reviews three serial bus protocols, the Serial Peripheral Interface (SPI) bus, the Inter-Integrated Circuit bus (I²C) bus and a JTAG bus, and discusses how one might use a general-purpose PXI digital instrument to emulate these busses.

SPI Bus

One of the simplest serial bus protocols is the Serial Peripheral Interface bus, or SPI Bus. The SPI bus is a synchronous serial data transfer standard popularized by Motorola. Devices on the SPI bus transfer information in a full duplex mode and communicate in a master/slave configuration, where the master initiates the data transfer between itself and one or multiple slaves.

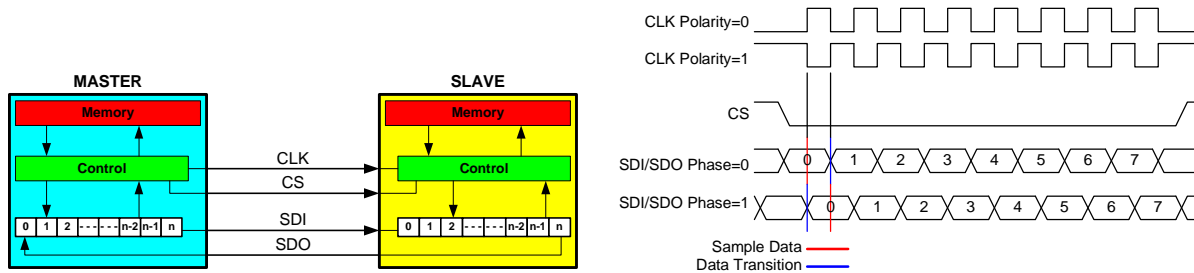


Figure 1 – SPI Bus

The SPI bus uses four signals to affect the transfer of information between devices (Figure 1). These signals are the Serial Clock (CLK), Serial Data Output (SDO), Serial Data Input (SDI) and Chip Select (CS). Data is transferred from the MSB of the Master I/O register to the LSB of the selected Slave I/O register, and from the MSB of the selected Slave I/O register to the LSB of the Master I/O register. As data is shifted out of the master device's I/O register, the data is being shifted into the slave device's I/O register. Simultaneously, the slave device is shifting data out of its I/O register and into the master I/O register. Data is transferred on either the rising or falling edge of the CLK signal, depending on the settings of the Clock Polarity and Phase parameters, and the master controls which slave device to communicate with by asserting the appropriate chip select signal. The data transfer clock typically operates at rates from 1.0 MHz to 70.0 MHz, providing data transfers rates up to 8.75MB/S.

I²C Bus

The Inter-Integrated Circuit bus (I²C) is a serial computer bus invented by Philips. Supporting multiple bus masters, I²C is used to communicate between low-speed peripherals and embedded systems. Information is transferred using a simple protocol consisting of three basic message types; a master write to slave message, a master read from slave message, and combination messages that involve multiple reads and/or writes between the master and a slave. I²C uses only two signals to transfer information between components. They are Serial Data (SDA) and Serial Clock

(SCL), which are open-drain signals with external pull-up resistors (Figure 2). Using an open-drain signal allows multiple bus masters to exist on the bus, and often the role of Bus Master and Bus Slave can be dynamically switched between devices on the bus.

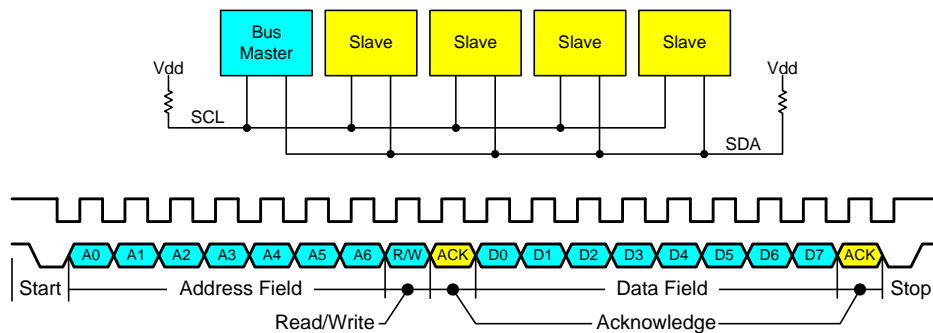


Figure 2 – I²C Bus

A typical message begins by the bus master transmitting a start bit – which is a high-to-low transition on the SDA line while the SCL line is held high. A 7-bit address field that uniquely identifies the slave for which the message is intended follows the start bit. A single bit identifying the message as a read or a write follows the address field. Data may transition on the SDA line only while the SCL line is held low. Data is transferred by a low-to-high and high-to-low transition on the SCL line. The slave, if present, responds with an ACK bit immediately after the Read/Write bit, and the master then continues with the remainder of the message, either transmitting or receiving data, depending on the type of message. The message is terminated by a low-to-high transition on SDA while SCL is held high. I²C is capable of transferring data at 100 Kb/S to 3.4 Mb/S, although the message protocol overhead reduces the throughput by approximately one-half.

JTAG 1149.1 Bus

JTAG (Joint Test Action Group) and Boundary Scan are the more common names for the IEEE 1149.1 - Standard Test Access Port and Boundary-Scan Architecture, which defines a test access port for detecting common manufacturing connectivity defects, such as PCB shorts and opens, and IC bonding failures. JTAG has evolved from a robust manufacturing test tool to a test bed for probing the internals of complex integrated circuits as well as performing functional tests on logic blocks inside of such devices, at less than full speed. Boundary scan is also commonly used for flashing (programming) PROM's, and programming CPLD's, FPGA's and other embedded components.

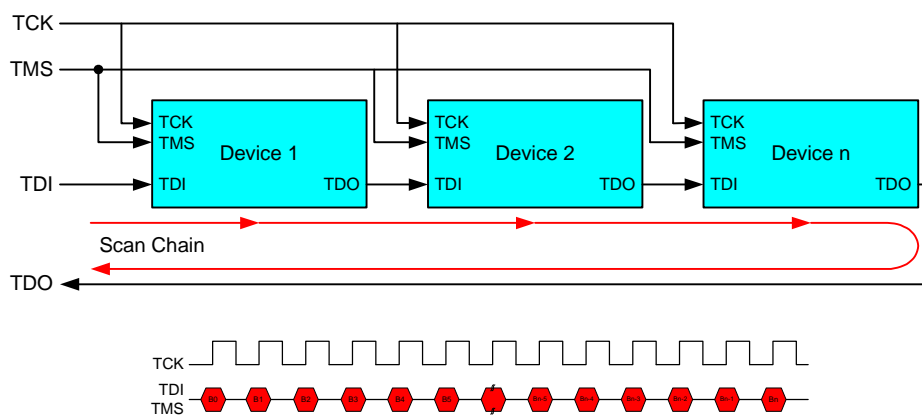


Figure 3- JTAG 1149.1 Bus

A basic Boundary Scan system utilizes either four or five signals, depending on whether the optional Reset signal is used. The four remaining signals are the Clock Input (TCK), Test Mode Select (TMS), Test Data Input (TDI) and Test Data Output (TDO) (Figure 3). Clocking a serial bit pattern into the internal state machine (commands) via the TMS pin configures all parts on the same scan chain in parallel. Data is clocked through parts in a Daisy Chain fashion using the TDI and TDO. The TDI signal on the first device in the scan chain is the primary data input. The TDO of the first device connects to the TDI of the second device, and its TDO connects to the TDI of the third device. This continues until the last device of the scan chain, who's TDO is the primary data output. Typical clock rates for TCK are 10 MHz to 100 MHz, although the slowest device in a scan chain dictates the fastest possible clock speed for the entire chain. Other timing parameters to be considered are the data valid-to-clock setup time and the clock-to-data invalid hold time. Additionally, to perform a comprehensive test via the JTAG interface, it is not unusual to require a sequence of vectors that might be 10 Megabits deep or more.

A PXI – Based, Serial Bus Test Solution

A basic dynamic digital PXI test instrument consists of a timing resource, typically a PLL-based programmable clock source, control logic, a pattern sequencer/address generator, and pattern memory. For basic serial bus emulation, pattern memory is used to store the output pattern which is sent to the UUT, a Tristate Control memory provides dynamic tri-state capability for bi-directional I/O control, and a Record Memory is used for capturing UUT responses. A host computer is connected to the digital instrument for programmatic control of the test hardware. Figure 4 details this basic instrument's architecture as well as the PXI bus connection to a host computer.

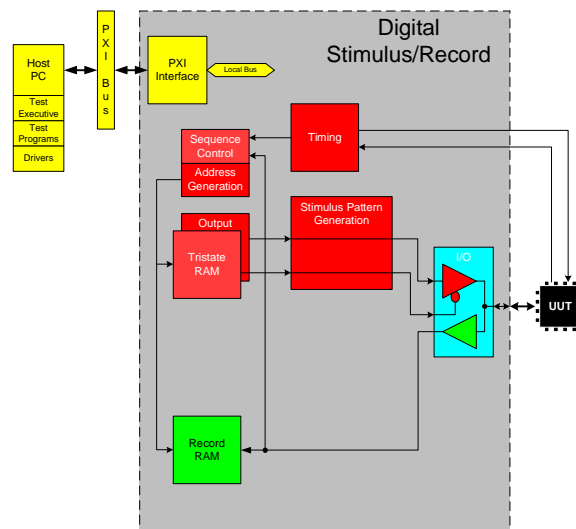


Figure 4 - Basic Digital Bus Test Instrument

Supporting software consists of instrument specific drivers and a high-level test program for generating and loading stimulus patterns, executing the test sequence and analyzing the UUT's response captured by the digital instrument. Controlling or programming a device via a serial link typically involves writing or reading the UUT's registers. The registers may be byte, word or even double word sizes, but regardless of their widths, the process is similar for accessing the register. The creation of a simple command interface (Figure 5) using a development environment such as ATEasy provides the primitive bus commands for reading or writing to the UUT registers – simplifying control and monitoring of the serial interface. The bus commands are compiled into serial patterns for downloading to the digital test instrument, with the pattern compilation being specific for the bus interface of interest. For example, if the commands need to support the I²C bus, then the commands' bit pattern will include a header which contains the seven-bit device address, whereas, an SPI command includes just clock and data patterns for shifting data to the UUT's register.

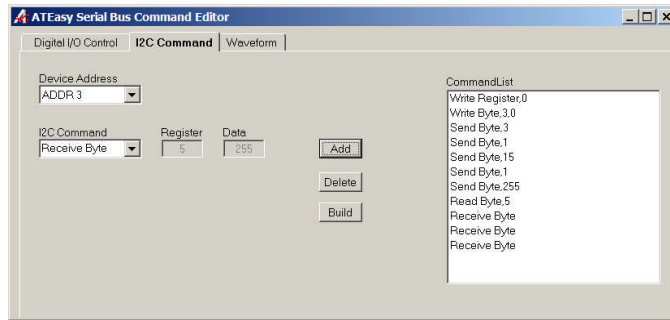


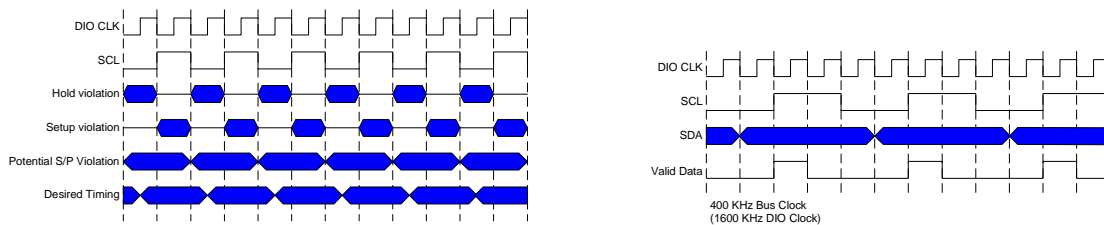
Figure 5 – Command Interface

The commands allow high-level configuration and control of the UUT regardless of the interface selected. The test patterns can be loaded directly to the test hardware, or stored to a file for later use in an ATE program. As the need to test devices with new serial interfaces are developed, only the functions for translating the primitive commands to the new serial format need to be developed. The actual commands do not need to be changed. High-level commands to support standard JTAG instructions (BYPASS, EXTEST, SAMPLE/PRELOAD) can also be added, including the ability to import binary files, such as Serial Vector Format files, which supports the programming of embedded EPROMS, CPLD's, FPGA's or other devices. The flexibility of the digital instrument allows many scenarios to be incorporated into the tester, or added to the tester as the need arises.

If the digital instrument also provides more advanced capabilities, such as edge timing placement, programmable I/O levels, or Real-Time compare, then the possibility to go beyond the mere exercising / emulating of serial busses exists. Input level sensitivity, validation of setup and hold parameters and real-time validation of the UUT response are examples of what is possible.

Using PXI Digital Instrumentation for Serial Bus Testing

Most serial busses are relatively slow, so clock speeds are generally not a problem. However, bus timing can be an issue. For example, if the bus requires adherence to strict data-to-clock setup and hold parameters, then the digital instrument needs to accommodate those requirements. A digital test instrument that allows a programmable skew offset to be applied between the internal data clock and the digital data outputs will provide the needed flexibility to meet the desired setup/hold specifications. Alternatively, if the digital instrument does not provide this functionality, then the most common approach for meeting the setup/hold specifications is to over - clock the digital instrument. For example, to represent an I²C data bit requires that the SDA signal change state only while the clock is low. Without a clock-to-data skew adjustment feature, the setup and hold time can only be met by over - clocking (over sampling) the digital instrument by a factor of 4 (minimum). For example, if the UUT's I²C bus operates at 400 KHz, then you would program the digital instrument to a 1.6 MHz data rate and use four digital instrument states to represent one I²C bus cycle. (Figure 6A & 6B) This method can be employed by virtually any digital test instrument, with two caveats: 1) The 4X clock rate is within the range of the DIO instrument. 2) A deep pattern buffer is required as the test pattern length is reduced by a factor of four (or more).



Figures 6A & 6B – Bus Timing & Over Sampling

Most digital test instruments provide many parallel channels, far more than is typically needed for serial bus test applications. The extra channels can be used to provide testing of multiple serial busses (multi-site testing) or for augmenting the capabilities of the test system. For example, if you need to trigger an oscilloscope at a particular time or event within a bus cycle – e.g. when data is valid – then an unused digital channel can become a trigger signal for an oscilloscope. You can program the strobe to occur at every valid bit within the bit stream, only when the UUT is responding, or at any other time of your choosing (Figure 6B).

Figures 7, 8 and 9 detail how a PXI digital test instrument can be used to support the SPI, I2C, and JTAG busses, including support for multiple busses.

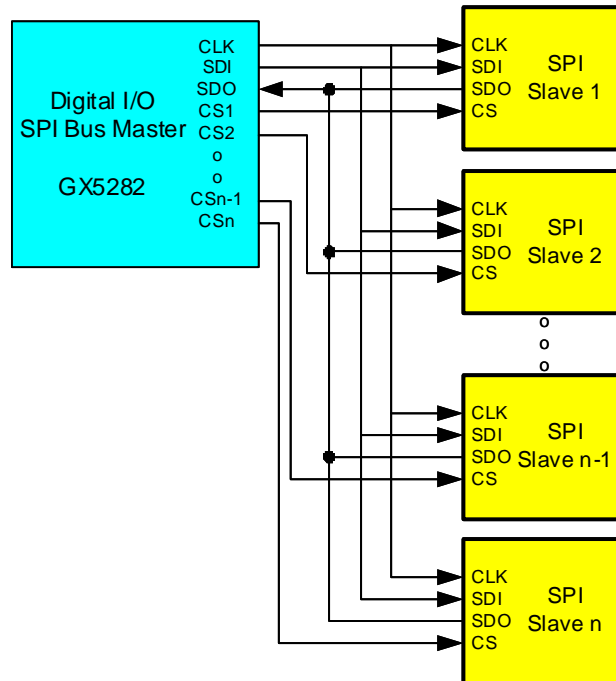


Figure 7 – Multi-channel SPI Bus Support

SPI Bus Support Key Features:

- 32 channel digital I/O supports up to 8 SPI slaves
- Programmable clock supports rates to 100 MHz
- Deep pattern memory (64 Mb / channel)

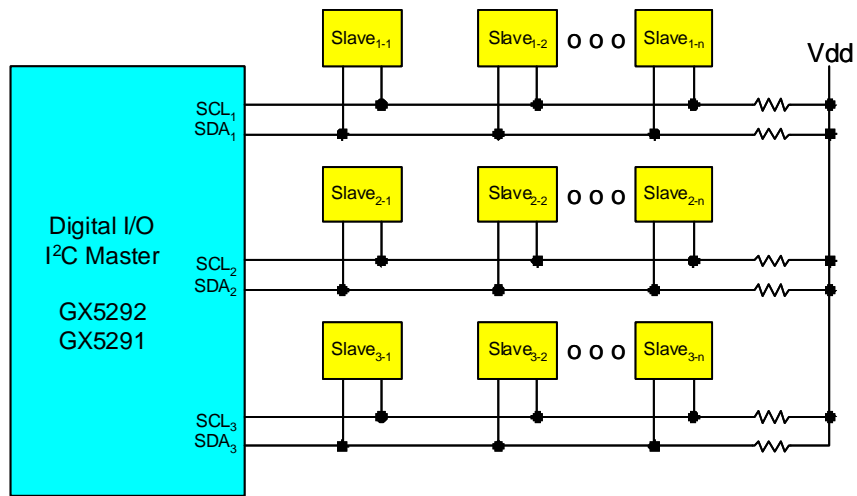


Figure 8 – Multi-channel I2C bus support

I²C Master Bus Support Key Features:

- 32 channel digital I/O supports up to 16 unique busses
- Programmable clock supports rates to 100 MHz
- Per channel / per vector direction control – required to support the bi-directional data bus

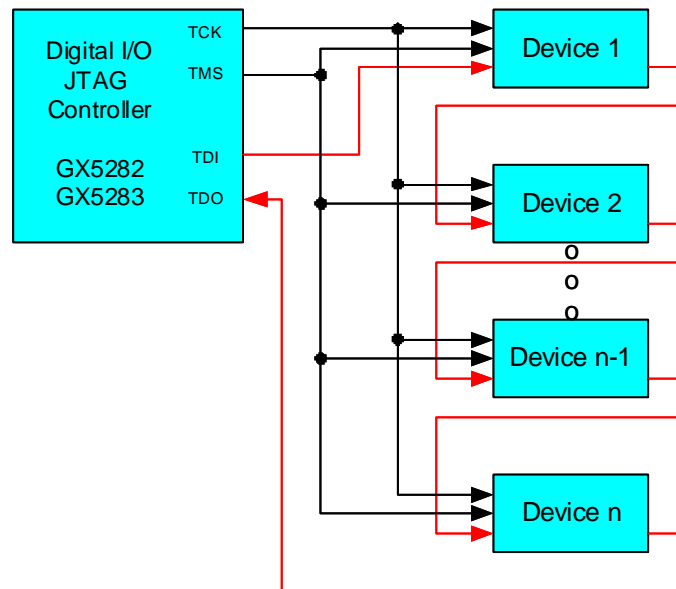


Figure 9 – JTAG Bus Emulation with a Digital Instrument

JTAG Controller Key Features:

- 32 channel digital I/O can support multiple JTAG ports or the unused digital channels can be used for synchronization or for other digital functional applications
- Programmable clock supports rates to 100 MHz or 200 MHz (depending on the instrument)
- The GX5283 offers deep pattern memory – up to 16 Gb per channel if configured for only 4 I/O channels

Summary

Using a general purpose PXI digital instrument to support serial interfaces can offer the user a large degree of flexibility when supporting both standard and non-standard protocols. Provided the PXI digital instrument has sufficient flexibility (and capabilities) and a flexible user interface is developed to simplify low-level control of the interface, a very powerful and time-saving test solution can be developed for supporting a variety of serial protocols. Additionally, since the solution is based on generic hardware and a software defined application; adapting / creating support for new protocols or variants can be easily done by upgrading the application specific software. The result is a solution of serial interfaces that is adaptable, cost-effective and future- proof.

References

http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

<http://en.wikipedia.org/wiki/I2c>

<http://en.wikipedia.org/wiki/JTAG>