XAPP369 (v1.0) December 5, 2001

Handheld 1553 Bus Data Analyzer

Summary

This document describes the implementation of the Handheld 1553 Bus Data Analyzer design submitted to the recently publicized "Cool Module Design Contest". All development for this contest was performed using the Insight Springboard™ development platform which allows for rapid development of Handspring™ modules. This development platform incorporates the reprogrammable Xilinx CoolRunner™ XPLA3 CPLD and uses the Handspring Visor PDA expansion slot. Low power CoolRunner CPLDs are the ideal programmable logic solution for portable, handheld applications.

The bus data analyzer is a Springboard module that tests the operating functionality of the standard military 1553 bus. The CPLD and Handspring design files for the Handheld 1553 Bus Data Analyzer are available and can be found at **Download Pack**, page 10. A visual presentation of this bus analyzer is provided to show operation and functionality. This on-demand video is available at: **http://www.xilinx.com/apps/video.htm**.

Introduction

In the winter of 2001, Xilinx, Handspring and Portable Design Magazine collaborated on a design contest to highlight the use of Xilinx CoolRunner XPLA3 CPLDs to quickly develop Springboard modules for the Handspring Visor PDA. About 250 contest registrations were taken and nearly 100 contest submissions were received. From the submissions, ten were chosen to receive a Handspring Visor and an Insight Electronics Springboard Development kit. The ten finalists were initially given two months to complete their designs and compete for a “winner takes all” grand prize. Submissions included a written description of the project, all design files, and the necessary software to make the Visor PDA operate. All the finalists did a great job. This application note is derived from the submission supplied by David Lott and Fred Jones of Titan Systems, Inc. at Point Mugu, California.

Additional information is included. **Appendix A**, page 10 outlines existing Xilinx application notes that are appropriate for understanding this application note. These are available on the Xilinx Website: **www.xilinx.com**. **Appendix B: Overview of Military Standard 1553 and 1773 Data Bus**, page 11 is a summary of the Mil-Std-1553/1773 standard which is provided with permission from the Electronic Warfare Handbook developed by the Point Mugu Naval Weapons Center. It is available on the internet at: **http://ewhdbks.mugu.navy.mil/1553-bus.htm**.

Design Description

The Handheld 1553 Data Bus Analyzer utilizes a 16MHz clock to analyze the Manchester serial signal of a MIL-STD-1553 data message. The MIL-STD-1553 data bus is widely used in military aircraft and development laboratories. It is also used on tanks, ships, satellites, missiles, and even space station Freedom. The Xilinx CPLD takes readings of the signal, based on a State Machine implementation, and stores 16-bit data words into memory. The Handspring software then reads, formats, and displays the data for the user. The PocketC compiler was used to develop the Handspring software, and the CPLD was programmed with VHDL using the Xilinx Project Navigator, Fitter, and Programmer.

As Figure 1 illustrates, the design was successfully implemented as a plug and play Springboard Device.

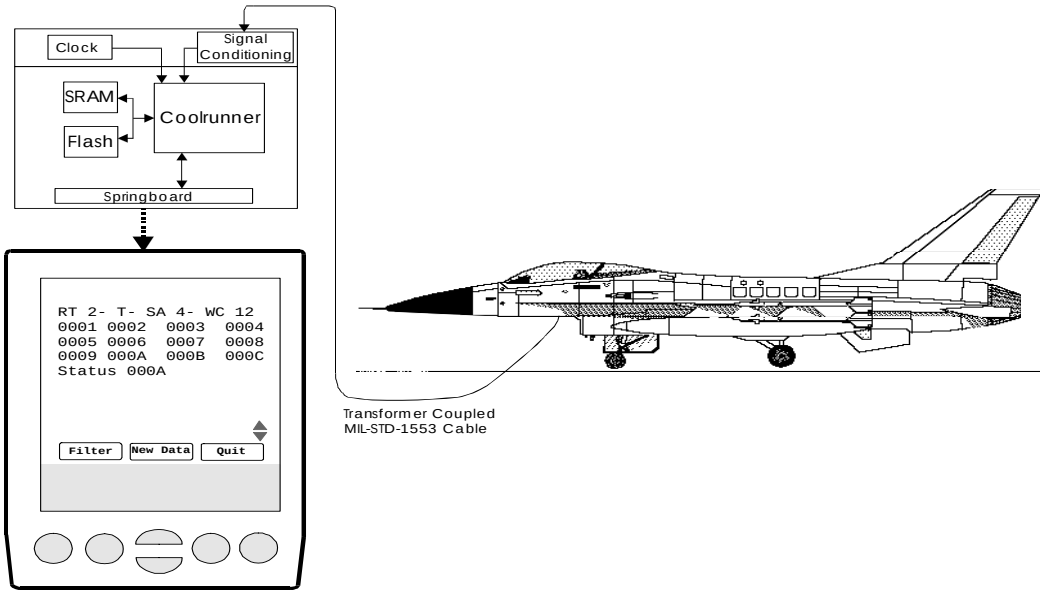

Figure 1: **High Level Diagram**

Implemented Features

The Handheld 1553 Bus Data Analyzer design includes all of the following, working features:

1. Capability to monitor network data on a MIL-STD-1553 data bus.
2. Remote network monitoring without need of a 60 Hz power source such as on a flight line.
3. Utilization of the CoolRunner CPLD to directly input the Manchester signal into the handheld device for serial to parallel conversion.
4. Capacity to store 16-bit data words on the resident SRAM for access by the Visor processor.
5. Able to run on the Handspring Visor with two AAA batteries.
6. Ability to determine filter criteria including the Remote Terminal number, Sub-Address number, and transmit/receive message type.
7. Displays Command Word, Data Words, and Status words for each message on the primary or secondary data bus.
8. Scrolls through messages and data.
9. Allows users to change the filter and access the same data set multiple times.
10. Provides for upgrades such as PC data upload, data interpretation, and message time stamping.

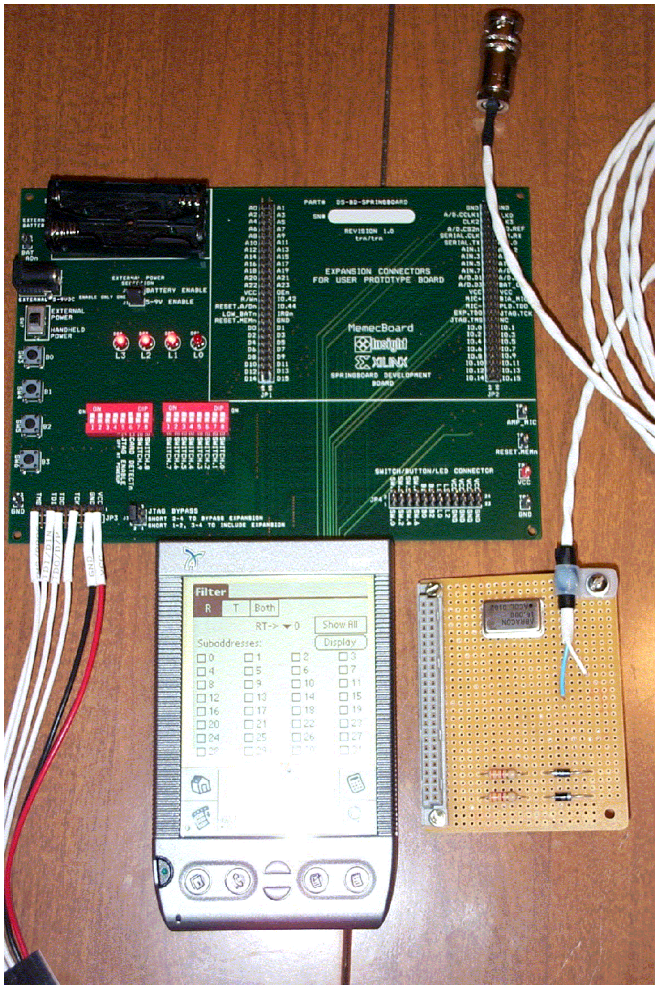

Figure 2: **Visor on the Filter Page with Interface Card and Cable**

Most 1553 networks are dual redundant and have a secondary bus that can be used if there is an error on the primary. A secondary interface can be easily added with an OR gate and logic to determine which bus is in use. The Handheld 1553 Bus Data Analyzer design presented here attaches to the main bus using a short stub connection. An expanded version would attach as either a short stub or as a long stub (transformer-coupled) connection, which requires the addition of an amplifer circuit to the interface card. In the version described in this application note, several functions that will be integrated in a version with a larger CPLD part have been ommitted. These include the checksum display, a Manchester error bit, the end message pointer, and data word count field.

The current 1553 bus data analyzer design allows for approximately 25,000 messages to be stored, and has proven successful in initial real-world tests (although it has not been tested at the limits of MIL-STD-1553). Figure 3 shows a more detailed block diagram of the target functionality.

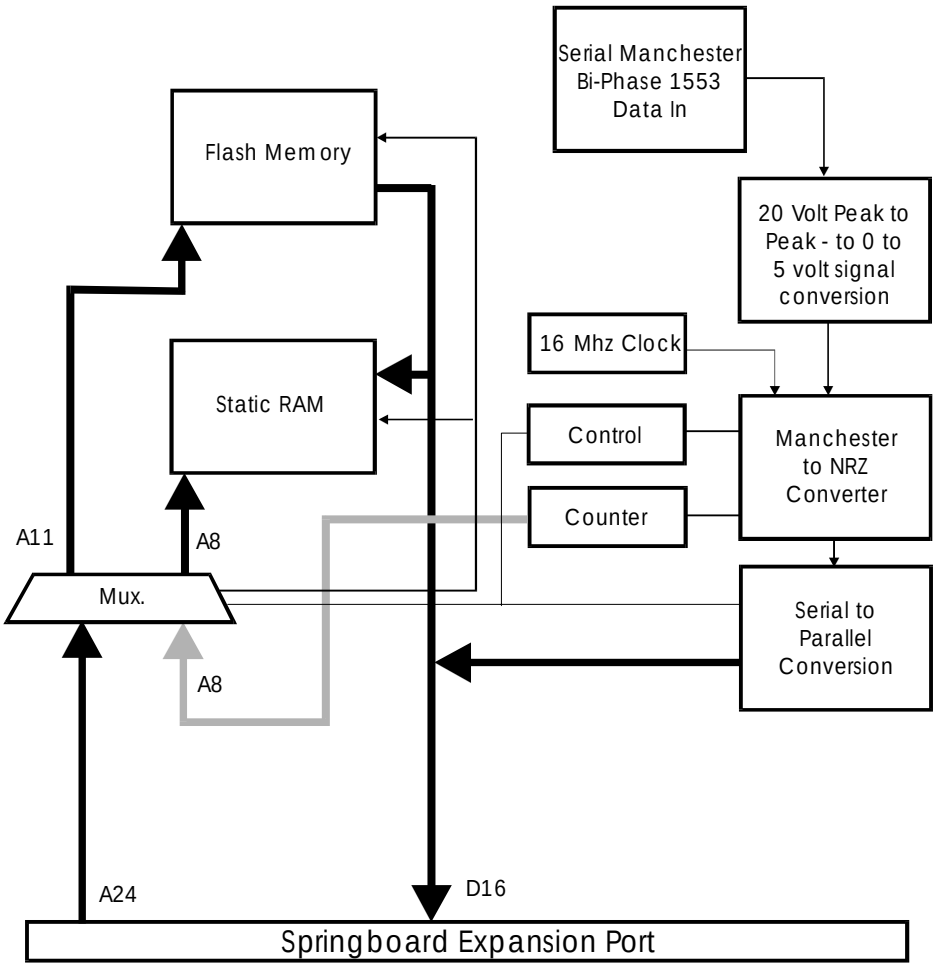

Figure 3: **CoolRunner Target Functionality**

Operation Instructions

1. Plug the 1553 Data Bus Analyzer (1553DBA) Springboard into a Visor, which has the IOLib native library installed on it. Verify that the 1553DBA launch icon is now visible on the main menu of the Visor. Also verify that the interface board is mounted on connector J2 of the development board (See Figure 4).
2. Connect the 1553DBA Springboard 1553 coupling connector to that of the 1553 data bus to be analyzed.
3. From the Main Menu on the Visor, launch the 1553DBA application. The Opening page of the application will come up.
4. When ready to gather 1553 data, hit the large “START” button in the center of the Visor display. This button will then darken and read “STOP.”
5. When finished gathering 1553 data, hit the large darkened “STOP” button. An alert will pop up indicating that the user may then disconnect the 1553DBA Springboard 1553 coupling connector from the bus. Hit "OK" and the application will change to the Filter page.
6. At this point the Filter page is displayed. The user may now select which messages to view on the Display page. In the top left corner is a 3-way radio button control reading “R” (Receive commands), “T” (Transmit commands) or “Both.” The Remote Terminal number selection is available on the popup control. Multiple subaddress selections can be made by checking the appropriate checkbox controls (if no subaddress selection is made, messages with any subaddress value will be displayed). A switch control reading “Show All” can be selected and all messages will be displayed. When finished with selections,

either tap the “Display” button or hit one of the up or down hard buttons. The application will switch to the Display page.

Figure 4: **Photos Showing the Interface Board Installed and the Program Running**

Hardware Description

The external circuitry is shown in Figure 5 below. The 1553 bus signal is input across two parallel diodes and resistors. The signal is split by the diodes into the positive and negative part of the signal. The AC signal is required for transformer coupling on the bus. This design only uses the positive part of the signal as a direct input to the PLD. The signals are in the 3V to 5V range, so further conditioning is unnecessary. A transformer-coupled version of the monitor would need some signal amplification before inputting to the PLD logic. The clock generates the needed 16 MHz signal and inputs it into one of the PLD clock inputs.

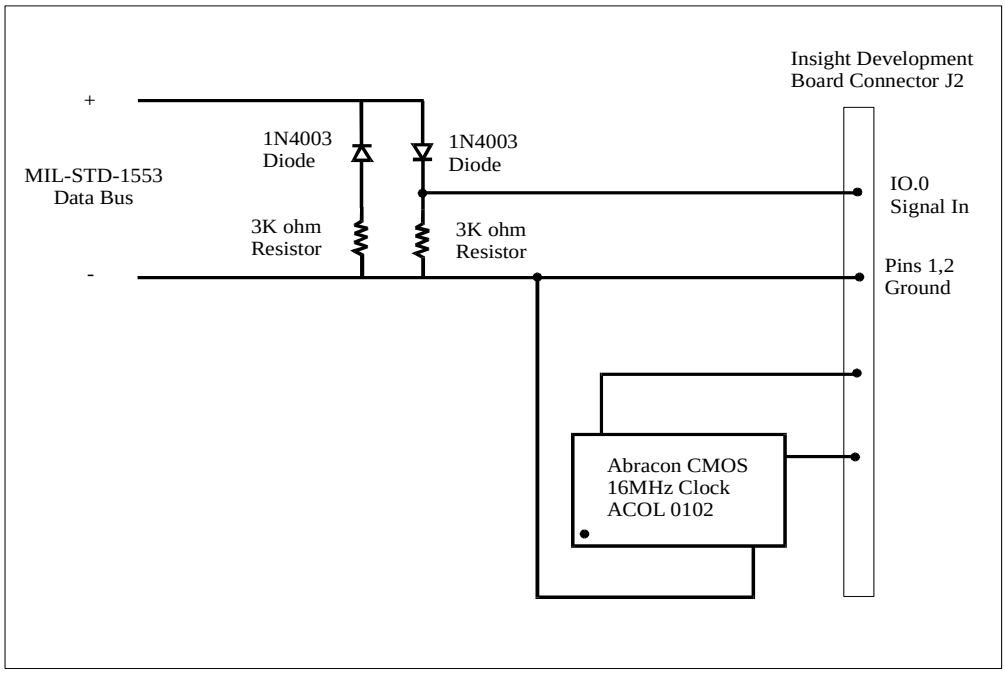

Figure 5: **Additional Circuitry for the Interface**

CPLD Software Description

The CPLD design is contained in two modules. The first module, labeled top_level.vhd, contains the definitions of the IO signals, the logic to allow SRAM and FLASH interface, the control of the execution from the Handspring software, and the Instantiation of the State Machine. The second module, decode_man.vhd, does the actual Manchester decoding and writing to the SRAM.

The CPLD design is based on a state machine that is shown in Figure 6. The CPLD states are described below:

IDLE — The Idle state is invoked with the command from the Handspring software. The next state is the New Message State, which happens when the data capture is enabled.

NEW MESSAGE — The New Message state increments the pointer by SRAM_INCREMENT to be ready for new message data.

BETWEEN MESSAGES — This state waits for a 0 to 1 transition on the input. When received, it initializes variables and goes to the time tag state.

TIME TAG — Writes the value of the clock counter to memory to time tag the message. Next state is the preamble check.

BETWEEN WORDS — Performs the same function as between messages for each subsequent word in the message. If too long a gap is seen, the next state is new message. When a 0 to 1 is seen, the next state is preamble check.

PREAMBLE CHECK — Determines if this is a command/status word or a data word. A command/status word has 1 bit followed by a 0 bit. A data word has a zero bit followed by a 1 bit. The next state is preamble wait for a command/status word, and read data for a data word. The first bit is read for a data word.

PREAMBLE WAIT — Waits for the first data bit and records it. Next state is read data.

READ DATA — Reads each of the remaining 15 bits of data and stores them. The Manchester read algorithm is read within each bit at the ¼ and ¾ points to determine if a 0 to 1 transition or a 1 to 0 transition has occurred. This transition converts the signal to a Non Return to Zero type signal which is stored in a 16-bit word.

CHECKSUM — Reads the checksum bit in the same manner as read data and stores it.

OUPUT — The output state writes the data word to SRAM.

MESSAGE COMPLETE — Writes the second status word to memory. This second status word is not written to memory due to the size of the program.

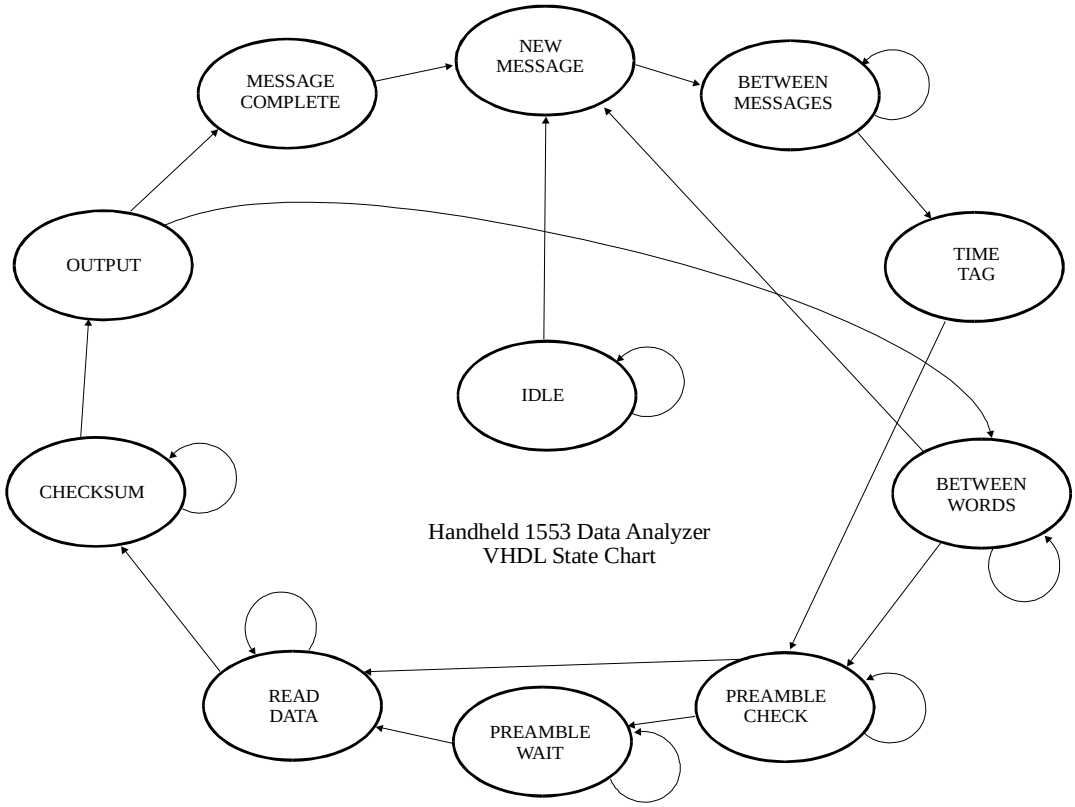

Figure 6: **VHDL State Diagram**

Handspring Software Description

The 1553 Data Bus Analyzer Handspring Application was created with PocketC (Desktop Edition v4.0.2). The application's GUI controls were initially created on the Visor itself using CControls/CEditor. Because the application utilizes the IOLib Native Library for accessing the Springboard SRAM, the IOLib Palm Application must be resident on the host Visor.

The Palm-MakeRom utility of the Handspring GNU tools was used to convert the Palm application into a binary file suitable for flashing the Springboard Toshiba flash memory. The Palm OS Debugger was used to do the flash memory programming.

The application consists of three pages:

- The Opening page, with a switch to start/stop the gathering of 1553 messages by the CPLD program.
- The Filter page, which allows the user to select which kinds of 1553 messages to view.
- The Display page, which displays the data and allows the user to scroll through the data. The Display page has menu selections which allow the user to go back to either of the other two pages or exit the application.

The Main routine of the application consists of a main outer loop with three inner loops for each page of the application. When a page is accessed it is initialized, drawn, and its loop entered. The loop continuously monitors and interprets user actions.

A extremely brief description of a MIL-STD-1553 message may be in order at this point. See **Appendix B: Overview of Military Standard 1553 and 1773 Data Bus**, page 11 for a more detailed summary. Each 1553 message consists of a command word, a status word and up to 32 words of data. For our purposes there are two types of 1553 messages, transmit commands and receive commands. The command word contains the Remote Terminal (RT)

number, type (transmit/receive or simply “T” or “R”), the subaddress and the word count of data words in the message. A remote terminal is simply another box (e.g. a GPS receiver), one of many which may be connected to the bus. A transmit command is essentially telling the RT “Send me this data.” A receive command is telling the RT “Take this data.” The subaddress number may be thought of as a location in the RT to get/put data. The data words in the 1553 message are the data which was transmitted or is to be received by the RT. The status word indicates various errors which may have occurred during the transaction.

It is the command word of the 1553 message which contains all the important information of the message (other than the data itself).

The 1553 Data Bus Analyzer, when gathering data, puts the command words of the 1553 messages at known locations in SRAM. In other words, the 1553 records are fixed-length. This makes it easy to search for messages with certain remote terminal numbers, types or subaddress numbers, which the user has specified on the filter page. The 1553 Data Bus Analyzer searches through SRAM in this manner. When a command word with matching characteristics is found, the rest of the message words are extracted from SRAM, buffered and displayed.

Memory Specification used by the 1553 Data Bus Analyzer

Tx (transmit) Cmd Record Format

i+70d Data word 32

.

.

.

i+8 Data word 1

i+6 Status word

i+4 Command word

i+2 Time stamp

i+0 Time stamp

Rx (receive) Cmd Record Format

i+70d Status word

i+68d Data word 32

.

.

.

i+6 Data word 1

i+4 Command word

i+2 Time stamp

i+0 Time stamp

where the first word of the first gathered record is located at i = 0x29000000 and the last word of the last gathered record is located at no higher an address than 0x29079998.

- Record formats are fixed in length at 36 words per record.
- Start/Stop Gather register located at 0x2907FFFE, contents=0 =>Stop gather, contents=1=>Start gather.
- address of the first word of the final record gathered located at 0x2907FFFC.

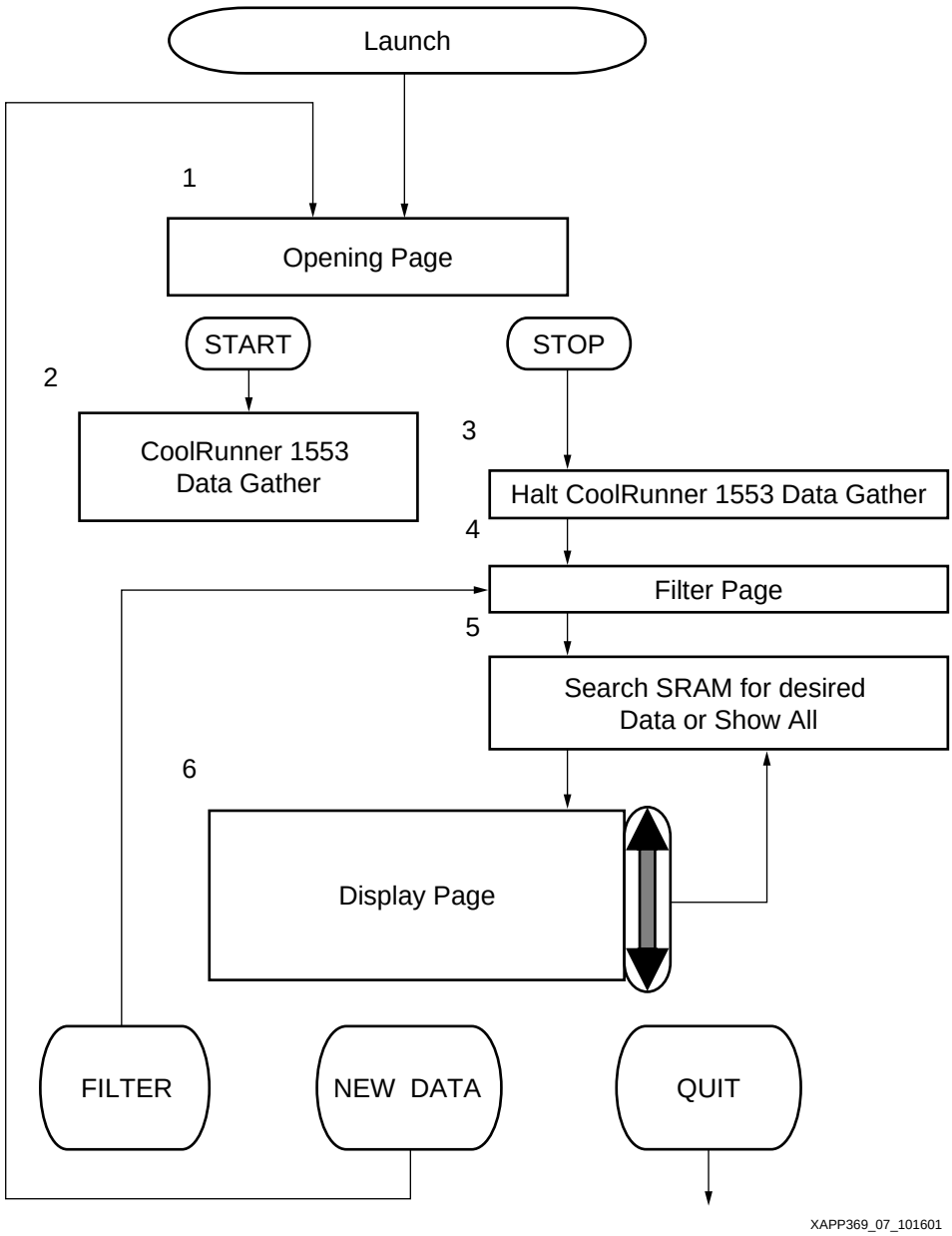

Figure 7: **Handheld 1553 DataBus Analyzer Software Flow**

Software Flow Description

1. After launching the 1553 Data Bus Analyzer, a “START” button is displayed, and the program waits for the user to tap it.
2. Once the user has connected the Springboard to the 1553 data bus being analyzed and has tapped “START” button, the CoolRunner gathers any 1553 data messages which are detected on the 1553 data bus into the Springboard SRAM. The button label changes to “STOP” and the program waits for the user to tap it.
3. The user taps the “STOP” button and the CoolRunner 1553 data gather halts. At this point the user may disconnect the Springboard from the 1553 bus.
4. The display changes to a set of checkboxes and radio buttons, which allow the user to display only those 1553 messages meeting the user’s criteria. Filter criteria shall include Remote Terminal number(s), Sub-address number(s), and transmit/receive message type. There are 32 possible Remote Terminals, and 32 possible sub-addresses for each transmit

or receive message. Data display will include message type, status word and data words. The user may select to see all messages. A single button “DISPLAY” will also be displayed. The program accepts user selections and waits for the user to tap the “DISPLAY” button or click either of the up/down hard buttons.

5. The user taps the “DISPLAY” button, clicks one of the up/down hard buttons, orscrolls the display (see step 6). Based on the user’s display criteria, the program will search the Springboard SRAM for the next matching 1553 message and display it. Should no matches be found, processing returns to step 4.
6. The first matching 1553 message is displayed to the user. The menu items “FILTER,” “NEW DATA” and “QUIT” are available, as well as scroll buttons, and the program waits for user input. The user can scroll by tapping the GUI buttons or by clicking the up/down hard buttons. Scrolling will bring processing back to step 5, selecting “FILTER” will bring processing back to step 4, selecting “NEW DATA” will bring processing back to step 1and selecting “QUIT” will terminate the program.

Download Pack

THIRD PARTIES MAY HAVE PATENTS ON HANDHELF 1553 BUS DATA ANALYZER. BY PROVIDING THIS HDL CODE AS ONE POSSIBLE IMPLEMENTATION OF THIS DESIGN, XILINX IS MAKING NO REPRESENTATION THAT THE PROVIDED IMPLEMENTATION OF THIS DESIGN IS FREE FROM ANY CLAIMS OF INFRINGEMENT BY ANY THIRD PARTY. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND XILINX SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE, THE ADEQUACY OF THE IMPLEMENTATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OR REPRESENTATION THAT THE IMPLEMENTATION IS FREE FROM CLAIMS OF ANY THIRD PARTY. FURTHERMORE, XILINX IS PROVIDING THIS REFERENCE DESIGNS "AS IS" AS A COURTESY TO YOU.

XAPP369 - **http://www.xilinx.com/products/xaw/coolvhdlq.htm**

Conclusion

This design submission for the Cool Module Design Contest is a perfect application of a low power Sprinboard module. Titan Systems, Inc.’s Handheld 1553 Bus Data Analyzer design submission is a portable, low power design that interfaces with the Handspring Springboard expansion slot. It is an ideal application for the Xilinx XPLA3 CoolRunner CPLD, as no other programmable logic device in the world is capable of delivering the Xilinx patented Fast Zero-Power (FZP) Technology, the combination of ultra-low power and uncompromised speed.

Appendix A

Appendix A lists appropriate Xilinx CoolRunner CPLD application notes. These application notes can be found by searching the Xilinx website and keying on the specific XAPP#. Many include appropriate driver software along with high level design code. All have been constructed and work.

PDA Springboard Design

XAPP147: Low Power Handspring Springboard Module Design with CoolRunner CPLDs

XAPP359: Understanding the Insight Springboard Development Kit

XAPP357: CoolRunner Visor Springboard LED Test

XAPP355: Serial ADC Interface Using a CoolRunner CPLD

XAPP146: Designing an Eight Channel Digital Volt Meter with the Insight Springboard Kit

XAPP149: Designing an Oscilloscope for the Insight Springboard Development Kit

Additional Application Notes

XAPP339: Manchester Encoder-Decoder for Xilinx CPLDs

XAPP341: UARTs in Xilinx CPLDs

References

Handspring website: http://www.handspring.com/

Appendix B: Overview of Military Standard 1553 and 1773 Data Bus

The following discussion is reprinted with permission from the Electronic Warfare Handbook published on the internet by Point Mugu Naval Weapons Center. The URL where additional material can be found is: http://ewhdbks.mugu.navy.mil/1553-bus.htm.

MIL-STD-1553 & 1773 DATA BUS

In recent years, the use of digital techniques in aircraft equipment has greatly increased, as have the number of avionics subsystems and the volume of data processed by them.

Because analog point-to-point wire bundles are inefficient and cumbersome means of interconnecting the sensors, computers, actuators, indicators, and other equipment onboard the modern military vehicle, a serial digital multiplex data bus was developed. MIL-STD-1553 defines all aspects of the bus, therefore, many groups working with the military tri-services have chosen to adopt it.

The 1553 multiplex data bus provides integrated, centralized system control and a standard interface for all equipment connected to the bus. The bus concept provides a means by which all bus traffic is available to be accessed with a single connection for testing and interfacing with the system. The standard defines operation of a serial data bus that interconnects multiple devices via a twisted, shielded pair of wires. The system implements a command-response format.

MIL-STD-1553, "Aircraft Internal Time-Division Command/Response Multiplex Data Bus," has been in use since 1973 and is widely applied. MIL-STD-1553 is referred to as "1553" with the appropriate revision letter (A or B) as a suffix. The basic difference between the 1553A and the 1553B is that in the 1553B, the options are defined rather than being left for the user to define as required. It was found that when the standard did not define an item, there was no coordination in its use. Hardware and software had to be redesigned for each new application. The primary goal of the 1553B was to provide flexibility without creating new designs for each new user. This was accomplished by specifying the electrical interfaces explicitly so that compatibility between designs by different manufacturers could be electrically interchangeable.

The Department of Defense chose multiplexing because of the following advantages:

- Weight reduction
- Simplicity
- Standardization
- Flexibility

Some 1553 applications utilize more than one data bus on a vehicle. This is often done, for example, to isolate a Stores bus from a Communications bus or to construct a bus system capable of interconnecting more terminals than a single bus could accommodate. When multiple buses are used, some terminals may connect to both buses, allowing for communication between them.

MULTIPLEXING

Multiplexing facilitates the transmission of information along the data flow. It permits the transmission of several signal sources through one communications system.

BUS

The bus is made up of twisted-shielded pairs of wires to maintain message integrity. MIL-STD-1553 specifies that all devices in the system will connect to a redundant pair of buses. This provides a second path for bus traffic should one of the buses be damaged. Signals are only allowed to appear on one of the two buses at a time. If a message cannot be completed on one bus, the bus controller may switch to the other bus. In some applications more than one 1553 bus may be implemented on a given vehicle. Some terminals on the bus may actually connect to both buses.

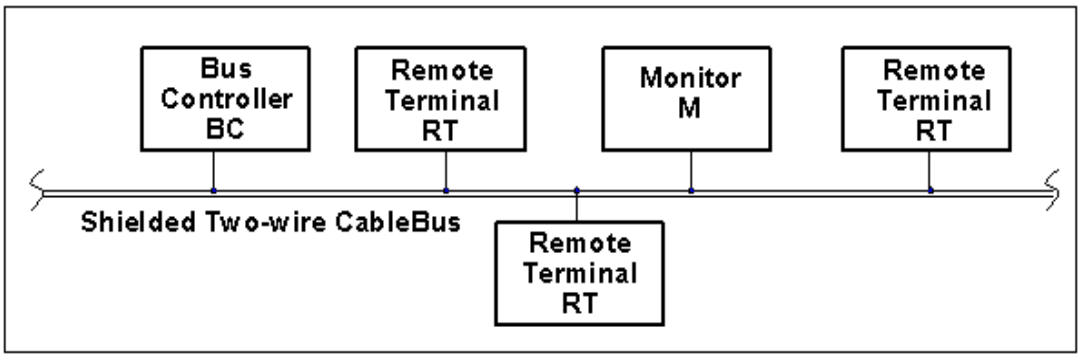

Figure 8: **1553 Bus Structure**

Bus Components

There are only three functional modes of terminals allowed on the data bus: the bus controller, the bus monitor, and the remote terminal. Devices may be capable of more than one function. Figure 8 illustrates a typical bus configuration.

Bus Controller — The bus controller (BC) is the terminal that initiates information transfers on the data bus. It sends commands to the remote terminals which reply with a response. The bus will support multiple controllers, but only one may be active at a time. Other requirements, according to 1553, are: (1) it is "the key part of the data bus system," and (2) "the sole control of information transmission on the bus shall reside with the bus controller."

Bus Monitor — 1553 defines the bus monitor as "the terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time." Bus monitors are frequently used for instrumentation.

Remote Terminal — Any terminal not operating in either the bus controller or bus monitor mode is operating in the remote terminal (RT) mode. Remote terminals are the largest group of bus components.

Modulation

The signal is transferred over the data bus using serial digital pulse code modulation.

Data Encoding

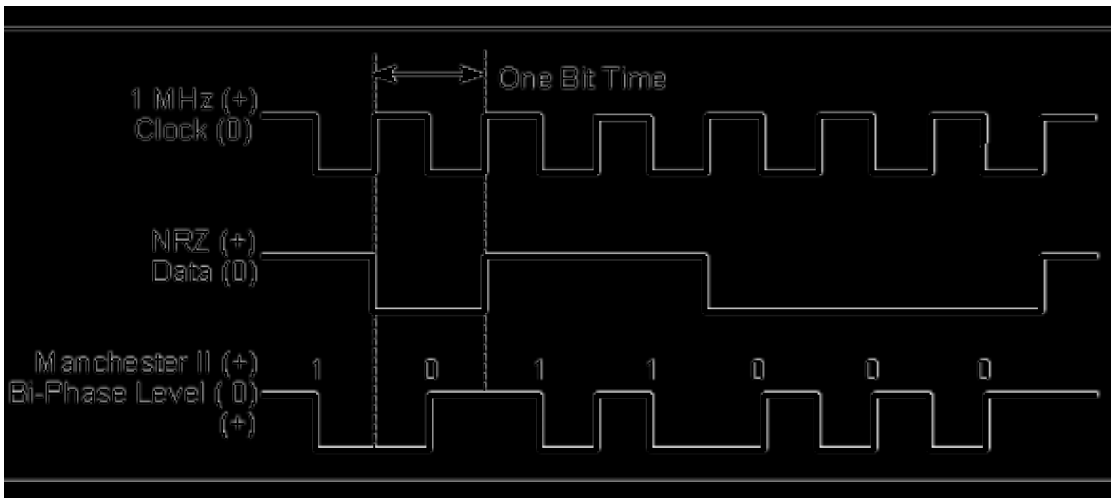

Figure 9: **Data Encoding**

The type of data encoding used by 1553 is Manchester II biphase.

A logic one (1) is transmitted as a bipolar coded signal 1/0 (in other words, a positive pulse followed by a negative pulse). A logic zero (0) is a bipolar coded signal 0/1 (i.e., a negative pulse followed by a positive pulse).

A transition through zero occurs at the midpoint of each bit, whether the rate is a logic 1 or a logic 0. Figure 9 compares a commonly used Non Return to Zero (NRZ) code with the Manchester II biphase level code, in conjunction with a 1 MHz clock.

Bit Transmission Rate

The bit transmission rate on the bus is 1.0 megabit per second with a combined accuracy and long-term stability of +/- 0.1%. The short-term stability is less than 0.01%.

There are 20 1.0-microsecond bit times allocated for each word. All words include a 3 bit-time sync pattern, a 16-bit data field that is specified differently for each word type, and 1 parity check bit.

Word Formats

Bus traffic or communications travels along the bus in words. A word in MIL-STD-1553 is a sequence of 20 bit times consisting of a 3 bit-time sync waveform, 16 bits of data, and 1 parity check bit. This is the word as it is transmitted on the bus; 1553 terminals add the sync and parity before transmission and remove them during reception. Therefore, the nominal word size is 16 bits, with the most significant bit (MSB) first.

There are three types of words: command, status, and data. A packet is defined to have no intermessage gaps. The time between the last word of acontroller message and the return of the terminal status byte is 4-12 microseconds. The time between status byte and the next controller message is undefined. Figure 10 illustrates these three formats.

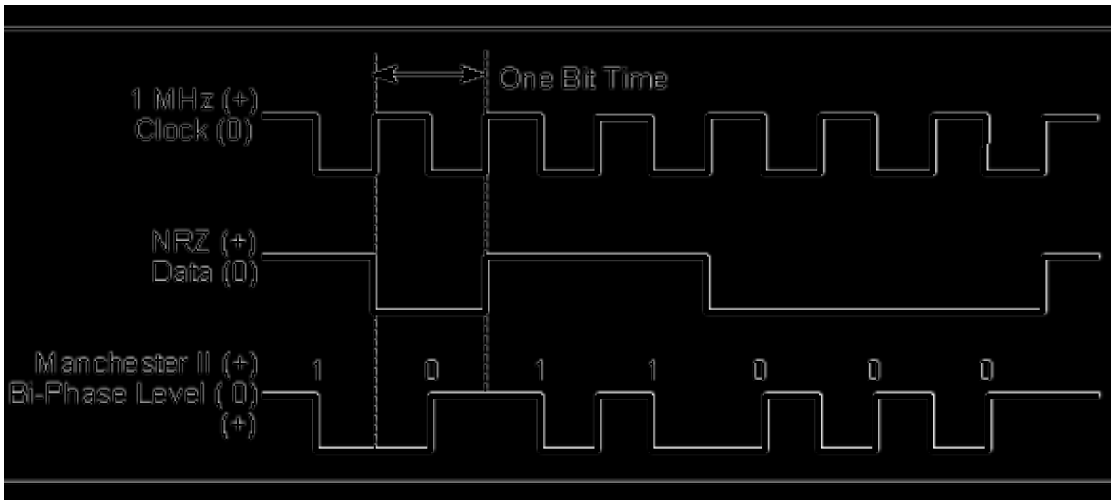

Figure 10: **Data Encoding**

Command Word

Command words are transmitted only by the bus controller and always consist of:

- 3 bit-time sync pattern
- 5-bit RT address field
- 1 Transmit/Receive (T/R) field
- 5-bit subaddress/mode field
- 5-bit word count/mode code field
- 1 parity check bit.

Data Word

Data words are transmitted either by the BC or by the RT in response to a BC request. The standard allows a maximum of 32 data words to be sent in a packet with a command word before a status response must be returned. Data words always consist of:

- 3 bit-time sync pattern (opposite in polarity from command and status words)
- 16-bit data field
- 1 parity check bit.

Status Word

Status words are transmitted by the RT in response to command messages from the BC and consist of:

- 3 bit-time sync pattern (same as for a command word)
- 5-bit address of the responding RT
- 11-bit status field
- 1 parity check bit.

The 11 bits in the status field are used to notify the BC of the operating condition of the RT and subsystem.

Information Transfer

Three basic types of information transfers are defined by 1553:

1. Bus Controller to Remote Terminal transfers
2. Remote Terminal to Bus Controller transfers
3. Remote Terminal to Remote Terminal transfers

These transfers are related to the data flow and are referred to as messages. The basic formats of these messages are shown in Figure 11.

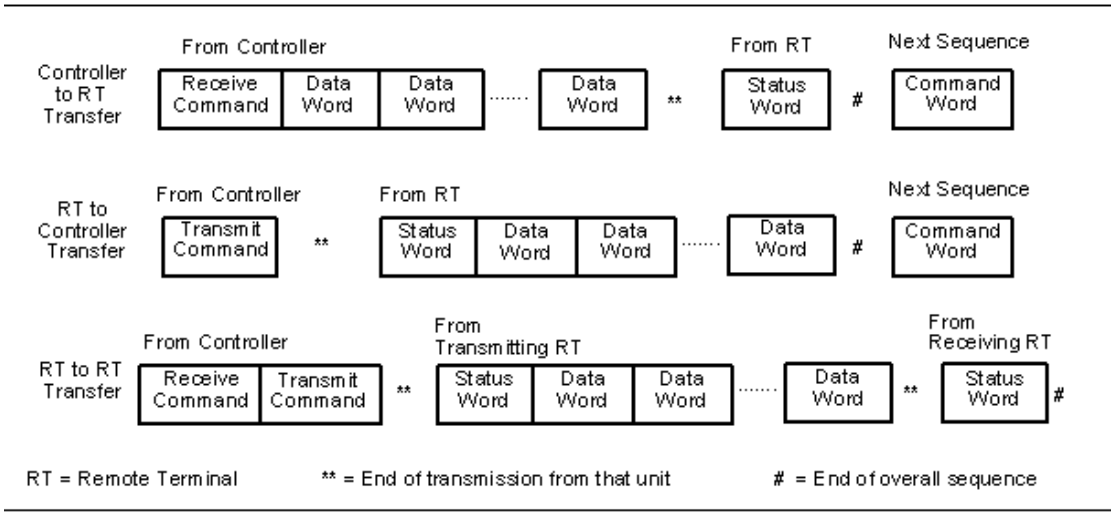

Figure 11: **1553 Data Message Format**

The normal command/response operation involves the transmission of a command from the BC to a selected RT address. The RT either accepts or transmits data depending on the type (receive/transmit) of command issued by the BC. A status word is transmitted by the RT in response to the BC command if the transmission is received without error and is not illegal.

Figure 12 illustrates the 1553B Bus Architecture in a typical aircraft.

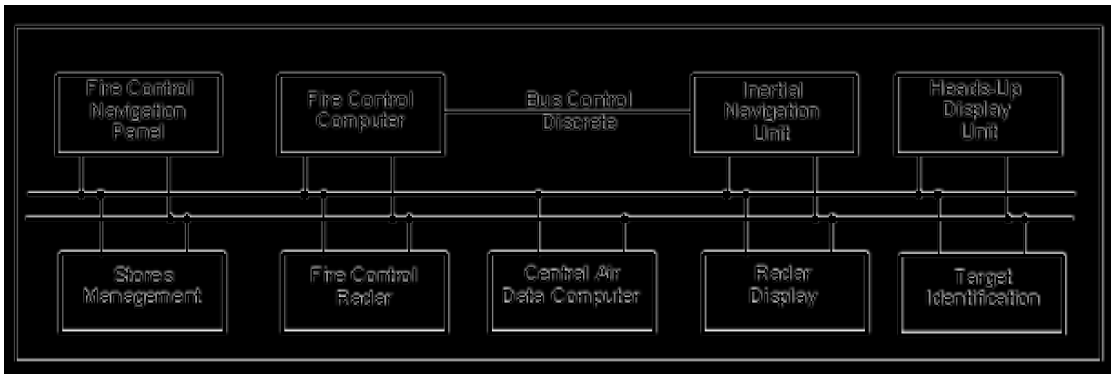

Figure 12: **Typical Bus Architecture**

MIL-STD-1773

MIL-STD-1773 contains the requirements for utilizing a fiber optic "cabling" system as a transmission medium for the MIL-STD-1553B bus protocol. As such, the standard repeats MIL-STD-1553 nearly word-for-word. The standard does not specify power levels, noise levels, spectral characteristics, optical wavelength, electrical/optical isolation or means of distributing optical power. These must be contained in separate specifications for each intended use.

Data encoding and word format are identical to MIL-STD-1553, with the exception that pulses are defined as transitions between 0 (off) and 1 (on) rather than between + and - voltage transitions since light cannot have a negative value.

Since the standard applies to cabling only, the bus operates at the same speed as it would utilizing wire. Additionally, data error rate requirements are unchanged.

Different environmental considerations must be given to fiber optic systems. Altitude, humidity, temperature, and age affects fiber optics differently than wire conductors. Power is divided evenly at junctions which branch and connectors have losses just as wire connectors do.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/05/01	1.0	Initial Xilinx release.