Development of a Programming Adapter
For a Classified-Data Processor Device

Dominic A. Peres
New Mexico Institute of Mining and Technology
Space Processor Engineering Department 05721, Sandia National Laboratories
Albuquerque, New Mexico 87185-0980, U.S.A.

Submitted August 4, 2000
Science & Technology Outreach Symposium
Sandia National Laboratories

Abstract:

This summer I was tasked to define and design a programming adapter (PA) that
interfaces a data processor device to a specialized cluster of computers at an
U.S. Air Force programming station. The PA allows classified serial data to pass
to target devices without being observed or recorded, ensuring the data will not
be compromised. The adapter is a command/response system translating
commands from the programming station into the device programming and
verification functions.

I am utilizing many computer aided engineering tools (schematic capture,
programmable logic designs and simulations) to develop a prototype of the
required adapter. The prototype is a scaled version of the PA to manage a single
target device but it contains the same logic that the full-scale PA will use to
manage multiple target devices. The prototype demonstrates functional and
security concepts and will be a model for the implementation of the final version
of the PA.
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Programming Adapter Design Concept and Systems</td>
<td>3</td>
</tr>
<tr>
<td>● Power System</td>
<td></td>
</tr>
<tr>
<td>● Power Switching Subsystem</td>
<td></td>
</tr>
<tr>
<td>● Parallel Port Discrete Logic Control System (PPDLCS)</td>
<td></td>
</tr>
<tr>
<td>● Target Device Matrix Subsystem</td>
<td></td>
</tr>
<tr>
<td>● Self-Test Subsystem</td>
<td></td>
</tr>
<tr>
<td>Development of the Programming Adapter Prototype</td>
<td>3</td>
</tr>
<tr>
<td>Software Development</td>
<td>5</td>
</tr>
<tr>
<td>Hardware Development</td>
<td>6</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>7</td>
</tr>
<tr>
<td>Prototype Evaluation</td>
<td>7</td>
</tr>
<tr>
<td>Design Recommendations</td>
<td>7</td>
</tr>
<tr>
<td>Conclusion</td>
<td>8</td>
</tr>
</tbody>
</table>

Appendix A: Project Overview
Appendix B: Power System
Appendix C: Parallel Port Discrete Logic Control System (PPDLCS)
Appendix D: Power Switching Subsystem
Appendix E: Target Device Matrix Subsystem (Full Scale)
Appendix F: Target Device Matrix Subsystem (Prototype Scale)
Introduction

This summer, I was tasked with the development of a design and prototype for a Programming Adapter (PA). This device must interface to a specialized cluster of computers at an U.S. Air Force programming station. The PA is a command/response system capable of recognizing commands from a host Programming Computer (PC) generating a response to these commands according to design requirements. The PA must also route classified serial data between a programming station and any target devices on the PA without compromising the data (see Appendix A). In this manner, classified data can pass through the adapter, but when data transfer is complete, the PA can be handled as an unclassified piece of hardware.

Programming Adapter Design Concept and Systems

The development of the PA was carefully guided by a set of standard specifications and design requirements provided by the Air Force programming station. Analyzing these design requirements, the PA was broken down into smaller sub-systems and a block diagram was made to aid the development process. This schematic can be seen in Appendix B through Appendix F. The systems identified were the Power System, the Power Switching subsystem, the Parallel Port Discrete Logic System (PPDLCS), the Self-Test Subsystem and the Target Device Matrix Subsystem.

• Power System (Appendix A)

This system is responsible for taking 110V AC input and converting it to 5V and 3.3V DC. There are two types of hardware internal to the PA, 5.0V hardware and 3.3V hardware. During operation, PA must power its internal logic.
systems and any target devices connected to it. The input current during PA operation must not exceed 10A at 5.0V and 3.3V. The PA must be capable of detecting a potentially destructive power condition and compensate for it by use of a fuse or a resetable circuit breaker. As one of the tests included in the Self-Test subsystem, a dangerous power condition will be induced to ensure detection and response.

- **Parallel Port Discrete Logic Control System (PPDLCS) (Appendix B)**

  The PPDLCS is in control of all the other systems and subsystems since it is responsible for receiving commands from the PC, determining what the command is, and what subsystems to implement in response to the given command. This system was the most complicated aspect of this project. Its function is analogous to the communications and processing node of the PA. The PPDLCS will interface with the PC and respond only to commands it recognizes, otherwise, the PA remains in an idle-state. The communications convention to be used to facilitate data communication between the PC and the PA must follow IEEE Standard 1284-1994 interface (Standard Signaling Method for Bi-directional Parallel Peripheral Interface for Personal Computers).

- **Power Switching Subsystem (Appendix C)**

  The Power Switching subsystem has two input systems, the Power System and the PPDLCS. This subsystem is responsible for identifying which column of target devices on the PA requires power. This condition is met when the Column_Select AND Power_Control_Logic signals from the PPDLCS are both TRUE. This activates an analog MOSFET switch and power from the Power System is then connected to the corresponding devices in a column. This allows power going to the target devices to be controlled using the command/response interface of the programming station.

- **Target Device Matrix Subsystem (Appendix D and Appendix E)**

  The Target Device Matrix subsystem is made up of a maximum of four rows and five columns of devices for a total of 20 devices that could be on the adapter at one time. Four serial channel inputs make up the rows of this matrix. These serial channels carry the classified serial programming data to all the target devices connected to its respective row. To select a column of four target devices, a column select signal is used. Since there are five columns of four target devices, there are five column selects. The column select signals are determined by de-muxing a three bit input from the PC to a corresponding seven bit output. Each output bit corresponds to the binary equivalent of the input bits. A binary input of '000' corresponds to No_Column_Selected, and a binary input of '110' or '111' corresponds to output bits not used by the PA. The other five cases are used as column selects A through E. In this manner, a target device can be isolated from both serial data and power until targeted by the PC via the
command/response interface. In addition, the classified serial data arriving to the PA from the serial ports of the PC use RS233 voltage logic. This system is to level shift between RS233 voltage logic (+12V and –12V) and TTL/CMOS voltage logic (+5V and 0V) to allow communications on the serial lines between the PC and the PA.

Appendix E depicts the same design concept as Appendix D, but on a much smaller scale. Appendix E represents one of the twenty target devices that could be attached to the PA at any one time. In this case, it is in row one, column A. For this target device to be targeted, the column select A signal must be TRUE and serial data must be coming in on channel 1. This simple arrangement demonstrates that an adapter on a much larger scale, as shown in Appendix D, is feasible.

- **Self-Test Subsystem**

The Self-Test subsystem is responsible for performing a diagnostic of the PA before any target devices are attached to or initialized by the PA. This system's sole purpose is to ensure PA functionality implying a comprehensive diagnostic of all systems and their supporting subsystems. If, for any reason, the self-test fails, the PA will not accept any further commands, with the exception of the read Self-Test status command, until the PA is reset to a known initial condition. A complete power down and reset of the adapter is necessary if the diagnostic is to be performed again.

**Software Development**

The logic controlling the PA was created using several engineering tools and techniques. Some of these tools were used to program discrete logic. A software package that does this is Altera and it was a primary tool in the development of the PA. The PPDSLCS was implemented using a finite state machine inside of a chip called a Programmable Logic Device (PLD). Using Very High Speed Integrated Circuit Hardware Description Language (VHDL), Altera Software, a table of project specific commands the PA will accept, and the IEEE Communications Standard 1284-1994, all the inputs, outputs and states were defined for the state machine in the VHDL code. The standard signal inputs controlling the PA are nDstrb, nAstrb, nWrite, and the data lines. The standard output signal from the PA to the PC is nWait and is used to facilitate a handshaking communications scheme with the PC as defined by IEEE Standard 1284-1994. This code was compiled, simulated, and troubleshoot in Altera.

The Target Device Matrix subsystem was created using Altera Hardware Description Language (AHDL). AHDL is like VHDL because both are languages representing an analytical expression of a graphical solution. The Target Device Matrix is made up of tri-state buffers. A tri-state buffer is a device that will connect its input to the output only when its enable signal is TRUE. The PA uses
these buffers to connect or isolate a target device from the serial lines. For instance, if the column select signal for column A is TRUE, then all the serial lines connected to the devices in that column are directly connected to their corresponding devices. If the column select signal is FALSE, then no direct connection exists between the serial data lines and the target devices in that column. In this manner, the data lines of four different serial channels can be isolated from the corresponding target devices in a column.

Other logic required to facilitate internal communication between the PPDLCS and its subsystems was done using a graphical representation of the logic. A circuit schematic is like linking many smaller blocks of logic together to create high level logic. This logic is necessary to decipher incoming commands into data the PA’s subsystems will understand. The registers, de-muxing chips and simple logic gates were used to create the logic for the column select signal and the target device power enable signal. These pieces were laid out, connected, and simulated in Altera. The two signals generated by this system are responsible for controlling the data and the power to a target column.

The three large blocks of VHDL, AHDL, and graphical logic combine to create one large logic block representing the internal control for the PA. This block was then used by Altera to program a PLD. The inside of the PLD is "wired" to execute a specified hardware task represented in the software descriptions above. The inputs and outputs of this chip are connected to the corresponding inputs and outputs from the supporting hardware of the test environment.

Hardware Development

After reviewing the specifications, the design concept of the hardware making up the PA was developed. This adapter could be done using a microprocessor, a PLD or both to control the PA’s discrete logic. The ease of use and flexibility of the microcontroller were sacrificed for a comparable design using only a PLD and an input clock signal to drive the internal logic of the PLD. This decision was made since memory in the microcontroller posed a security concern to the classified algorithms on the serial lines. To implement the PA using a PLD, a test environment for the design had to be created. This test environment is made up of a bread board, signal and data connectors, power connectors, filtering elements, switches, digital and analog devices, and the seats for chips to be put into. This test environment is the PA prototype. The prototype is a scaled version of the PA design, but it contains the same logic that the full-scale PA would use to function and demonstrates the feasibility of the design. Hardware components used to support both the PA and the interface to an external host:
• Instead of an internal supply, power will be supplied by means of an external power supply. This is brought into the board by three connectors (+5V, +3.3V, GND).

• The clock signal driving the circuit will also be externally supplied by input from a function generator.

• A double pole; single throw switch is used to initialize the +5V and the +3.3V inputs simultaneously allowing the PA to power up in a more stable manner.

• Light Emitting Diodes (LEDs) are connected to the input of the board to indicated power for the +5V connector and the +3.3V connector. An LED indicator is also used to report if MOSFET power for column A is active.

• In series with the load of the both the +5V and +3.3V inputs are 0.1-Ohm resistors with two filtering capacitors, in parallel with the load, after this. These capacitors are used to keep noise off the inputs of the PA by filtering out AC components.

• A DIP switch will be used to simulate the Self-Test by returning to the PLD a binary ‘0’ for Go and a binary ‘1’ for No_Go. Recommendations for tests involving the full-scale version will be discussed later in this paper.

• The serial input of the PA will transfer data to and from the selected target devices through a voltage level shifter. The voltage level shifter will be used to shift from RS232 levels to TTL/CMOS levels. The voltage level shifter used in the prototype is an ADM233L. This shifter is capable of shifting two bits from RS232 to TTL/CMOS and another two bits from TTL/COMS to RS232.

• A parallel port 26-pin connector carries the discrete logic commands of the command/response interface of the PA. These lines are already at TTL/CMOS levels, so no shifting is necessary. Each signal on the port will be wired to its corresponding pin on the PLD.

• Instead of interfacing the PA to up to 20 target devices, the prototype will interface with only one target device represented by row one, column A. This will still demonstrate the PA is capable of deciphering the address of the column select and supplying both data and power to that device.

• Communication between one target device and the serial lines of a PC is facilitated through a project specific board to and from which data and control signals can be sent between the PA’s active serial line and the devices targeted by a column select.

Results and Discussion
Prototype Evaluation

The PA prototype is currently under construction. The simulations of the logic in the PLD have been good. I am confident the adapter will perform as designed. When tests on the prototype are complete, a more thorough review of the PA design will be made along with recommendations for implementing the design in the future.

Design Recommendations

The PA prototype did not implement any self-tests. Some recommendations for several tests that could be conducted are over-voltage testing in which an over-voltage condition is induced and the PA's self-test must recognize this condition. If the PA fails to see the induced test, the self-test must report the PA as inoperable. Other possible tests include over-current tests, temperature tests, and discrete logic reliability. Most of these tests are feasible with the exception of the discrete logic reliability of the Target Device Matrix subsystem. Since this subsystem touches the serial lines the classified data is on, any intrusion into this system to test the tri-state buffers could be seen as a security concern since the objective of the PA is to isolate these lines. Although the Air Force programming station left these tests to be defined by the developer, the more simple the tests are, the more likely it will be to the required security approval.

Conclusion

This definition and development of the PA was an exercise in determining the best way to meet a project's specifications. This includes security concerns, hardware challenges, software challenges, planning circuit schematic layouts, and troubleshooting high level logic. This project has exposed me to many new tools such as Mentor Graphic's Very Best (Schematic Capture), VHDL, board layouts, and hierarchical schematic drawings of the PA prototype. The prototype will be used to demonstrate the PA design and to troubleshoot design errors in an isolated, low-risk environment until a reliable model is available for future reference when a full-scale PA is needed.
Appendix A: Power System

AC/DC Power Supply w/ dual outputs of +5V and +3.3V

+5V Hardware
- RS232 to TTL/COMS Voltage Level Shifter
- Other Analog Circuitry

+3.3V Hardware
- MOSFET
- Altera PLD
- Other Analog Circuitry

Power Filter
- Analog Circuit designed to filter out AC components of the inputs
- Circuit Logic for POWER ON RESET.
Appendix B: Parallel Port Discrete Logic Control System (PPDLCS)

- Column Select Logic
- IEEE-1284 Communication Logic
- Adapter ID Logic
- All Command/Response Logic

Command/Response Logic in a PLD.
Appendix C: Power Switching Subsystem

Parallel Port
Discrete Logic Control System

Demuxed Output for CS

Power Control Logic

AND Gate Logic in a PLD

Logic 0 or 1

MOSFET

PWR_A

Logic 0 or 1

MOSFET

PWR_B

Logic 0 or 1

MOSFET

PWR_C

Logic 0 or 1

MOSFET

PWR_D

+3.3V

Power to Targeted Column of Devices

CS_A = Column Select A

Power Control Logic is 0 for OFF and 1 for ON

Logic 0 = OFF Logic 1 = ON

PWR_A = +3.3V to Column A

Power System
Appendix E: Target Device Matrix Subsystem

Figure of Column A Channel One Only

* NOTE: Clock Signal not shown

Serial Interface to the PC

Voltage Level Shifter
12V <-> 0V
-12V <-> 5V

Rx1

Voltage Level Shifter
12V <-> 0V
-12V <-> 5V

Tx1

Voltage Level Shifter
12V <-> 0V
-12V <-> 5V

FCO1

Voltage Level Shifter
12V <-> 0V
-12V <-> 5V

FCI 1

Targeted Device

Tri-State Buffer

Tri-State Buffer

Tri-State Buffer

Tri-State Buffer

Tri-State Buffer Logic in a PLD

Program Discrete Logic

Parallel Port Discrete Logic Control System

PWR_A

* Note: This diagram does not illustrate the integrated test circuitry for the BIT Subsystem

* This diagram illustrates the relationship between the serial port and parallel port signals routed to the Target Device. This system will be implemented on a scale 20 times this size. Only one target device will be used for the PA prototype.