BOUNDARY SCAN TECHNOLOGY

By Charles Armijo and Semra Bekele

We have been assigned the task of designing a printed circuit board. The purpose of the board is to test the circuit integrity of the Muon Trigger Card (MTCXX) board. The Crate Manager is a highly sophisticated printed circuit board measuring about 9U or 400mm square. This board slides into a mounting frame and is connected to a 6U backplane, or backboard. It attaches through two VME (Versa Module Eurocard) bus connectors. One connector consists of 96 pins while the other has 160 pins.

At first the idea of designing a board to test circuit integrity seems quite daunting. However, it would be considerably more difficult to manually test the circuits. Consider for a moment the task of manually testing circuit integrity on a modern printed circuit board similar to the MTCXX. This kind of board can consist of more than 20 layers of circuitry. Clearly, there is no physical way to access circuits in the interior of such a board. Additionally, the size and configuration of chips and other devices mounted on the board make examination of connections to individual pins extremely difficult. The 144 pin chips used in our project have 36 pins per side, which translates to approximately 36 pins per inch. Our task is greatly simplified by the use of JTAG boundary-scanning technology.

In the 1980s, the Joint Test Action Group (JTAG) developed an electrical engineering specification for boundary-scan testing that was later adopted as the Institute of Electrical and Electronics Engineers (IEEE) standard 1149.1. This method brings back the access to device pins by means of an internal shift register around the boundary of the device – a boundary scan register. The BST (Boundary Scan Testing) architecture offers the capability to efficiently test components on PCBs with tight pin spacing. Consider that in this project we will be using 144 pin EPM3128 ALTERA® chips. The BST architecture can test pin connections without the physical probes and it captures functional data from pin or core logic signals. Forced test data is serially shifted into the
boundary-scan cells. Captured data is serially shifted out and externally compared to expected results.

The collection of boundary-scan cells is configured into a parallel-in, parallel-out shift register. A parallel load operation – called a Capture operation – causes signal values on device input pins to be loaded into cells, and signal values passing from internal logic device output pins to be loaded into output cells. A parallel unload operation - called an Update operation – causes signal values already present in the output scan cells to be passed out through the device output pins. Signal values already present in the input cells will be passed into the internal logic.
Data can also be shifted around the shift register, in serial mode, starting from a dedicated device input pin called Test Data In (TDI) and terminating at a dedicated device output pin called Test Data Out (TDO). The test clock, TCK, is fed in via yet another dedicated device input pin and various modes of operation are controlled by a dedicated Test Mode Select (TMS) serial mode signal. Normally, if there is an edge-connector input called TDI connected to the TDI of the first device, the TDO from the first device is permanently connected to TDI of the second device, the TDO of the second device is connected to the TDI of the third device and so on.

At the device level, the boundary-scan element contributes nothing to the functionality of the internal logic. It is independent of the function of the device. This means that we can design a printed circuit board using the JTAG Boundary-Scan...
functionality incorporated into the ALTERA® programmable gate array chips without knowing anything about the logic circuits on the Muon Trigger Card.

![Diagram](image)

**Fig. 3**

Captured data is serially shifted out and externally compared to expected results.

VME bus is a computer bus architecture. As it is states above the term VME stands for Versa Module Eurocard and was first coined in 1981 by the group of manufacturers who defined it. This group was composed of people from Motorola, Mostek and Signetics corporations who were cooperating to define the standard. The term "bus" is a generic term describing a computer data path, hence the name VME bus. VME bus address and data buses are up to 64-bits wide, it performs multiprocessing and can handle up to seven interrupt levels. Both the address and data busses can be dynamically configured (they change widths automatically). This allows system expansion as microcomputer technology grows. It also handles data transfers at speeds up to 80Mbytes/second.

VME bus uses master-slave architecture. Functional modules called master transfer data to and from modules called slaves. Since many masters can reside on the bus it is called a multiprocessing bus. Before a master can transfer data it must first
acquire the bus using a central arbiter. This arbiter is part of the module called the 
system controller. Its function is to determine which master gets the next access to the 
bus. All bus activity takes place on the four sub-busses. These are called the Data 
Transfer Bus, the Data Transfer Arbitration Bus, the Priority Interrupt bus and the Utility 
bus.

VME bus is asynchronous (meaning that no clocks are used to coordinate data 
transfer). Data is passed between modules using interlocked handshaking signals. Cycle 
speed is set by the slowest module participating in the cycle. Microcomputer buses 
usually fall into one of the two general categories: multiplexed and non-multiplexed. 
Multiplexed buses share the same set of pins for address and data lines. A two-part bus 
cycle is required to transfer data. During the first portion of the cycle an address is 
passed over the pins. Data is then moved during the second half of the cycle. Non-
multiplexed busses have separate pins for both address and data lines. Unfortunately this 
also means more bus interface circuitry and signal pins are required. VME bus uses both 
multiplexed and non-multiplexed architecture. For address and data transfers of up to 32-
bits, VME bus uses multiplexed architecture on 32 address and 32 data pins. The 
Revision D version of the VME bus specification (VME64) allows 64-bit address and 64-
bit data transfers. This is accomplished by multiplexing over the same set of pins. For 
this reason VME bus is considered a hybrid architecture.
Our first step in doing this project was to layout the schematic of the test board on PowerLogic. PowerLogic is a multi-sheet, schematic program, which allows the user to layout the design schematic for a printed circuit board. We were given a schematic that has the MTCXX VME connectors with the control bus connector. The VME connectors have data lines, address lines, control signals, power and ground pins. We laid out that same schematic on Power Logic along with four 144 pin Altera chips. The pins on the Altera chips correspond to the pins on the VME connectors. The JTAG pins are also located on the four chips- namely TDI, TMS, TCK and TDO on pins 4, 20, 89 and 104 respectively.

The way the JTAG signals travel through the devices is as follows – the edge connector input (TDI_1) is connected to the TDI_2 of the second device, the TDI_2 Connects to the TDI_3 of the third chip, the TDI_3 of the third chip is connected to the TDI_4 of the forth chip. Finally, TDI_4 goes into the J1 connector and goes out as TDO. The TMS and TCK also connect to the J1B connector. The JTAG signals can connect to any of the VME connector pins as long as there are available unused pins.
We have two sets of VME connectors (J1 and J2) each having 160 pins. J1 has data lines, address lines, control signals, power and ground pins. J2 has additional data lines, address lines and extra power and ground pins. These pins are connected to the corresponding pins on the four chips.

We have six layers on our board.

1) Primary Component Side

2) GND 1st plane

3) +3.3V Power Plane

4) +5V Power Plane

5) ±12V Voltage Plane

6) Secondary Component Side

Electronic noise is present in all VME bus systems and can never be eliminated. At low levels it should not cause problems, but at large levels it can render a system unusable. The amount of noise will directly correspond to the reliability of the system, and is therefore prudent to reduce it as much as possible. So to reduce the noise, we assigned .1 µf capacitors that correspond to each power (+5V, +12V, -12V, +3.3V). Since we have four EPM3128 144 pin chips that have 11 power connections on each one of them, we added 44 capacitors Thus, eleven capacitors must be physically place around each ALTERA® chip.
In addition to those capacitors, we added 4 more 10µf capacitors that correspond to each power. We have a 10 pin Test Point connector connected to ground that we could hook up a probe to if needed. We also have an 8-pin connector connected to ground and power. On each pin of the JTAG configuration, we have test pins to be able to check, for example, what kind of signal is passing through them.

We imported the Schematic to Power PCB from the schematic-capture (Power Logic.) The Power PCB shows the actual layout on the board with all the netlists. A netlist is a connection list for each signal in a design by providing part name and pin number. We actually created netlists from Power Logic almost at the end of the design process. The output format was the Power PCB format. We combined the netlists of the test board and the MTCXX board and the output was displayed in notepad as an ASCII file.
The +12V and –12V powers on the second ground plane were split using the split-layer tool in order to isolate electrical signals from each other. After splitting them, we flooded the plane. Flooding means filling a previously defined copper pour area. Originally the plane is filled with copper insulation areas around traces and pins that pass through the copper but they aren’t attached or connected to the copper.

When we checked connections using “verify design” tool, there were ten errors under the split/mixed ground plane. We removed the ground connections from that plane since there is another ground plane. We renamed that plane “voltage plane” instead of second ground plane. The errors were no longer there after removing the ground connection from that plane.

The parts were piled up together in Power PCB, so we needed to move and position them and it took a great deal of time and effort to lay out the board properly. The routing and clearance are categorized under design rules. We did some manual routing and then afterwards auto routing. The auto routing was done using the Blaze Router. Routing creates a metal etch trace of specified width between pads. The Blaze Router sequentially routes the unrouted traces until all the connections are routed. We looked at the clearance rules for each layer to make sure that there is enough space among pads, traces and vias to prevent short.

When we checked connections using “verify design” tool, there were ten errors under the split/mixed ground plane. We removed the ground connections from that plane since there is another ground plane. We renamed that plane “voltage plane” instead of second ground plane. The errors were no longer there after removing the ground connection from that plane.

At this stage of the design process, we produced a machine readable board description suitable for the PC house to manufacture the board. We used CAM (Computer Aided Manufacturing) in order to return the layout and to view the final design to its final artwork. The commands of the CAM have printing and plotting options. From PCB, we went to CAM to draw a precision plot of the “artwork”. We plotted each layer and added solder mask on the top and bottom of the design. Silk
screen pattern was also added on the top portion of the design. The solder mask is an artwork layer for a non-conductive material that covers the entire board with the exception of pad locations. It is a protective covering and prevents short. Silkscreen is also another artwork layer that comprises the reference designator and component outline of all parts used for the final board fabrication.

There is an artwork tape that tells the photoplotter how to draw a precision plot of “artwork”. It is written in “Gerber Format”. Gerber is the language that drives a photoplotter. This ASCII file language has instructions for selecting an aperture (a uniquely shaped window or hole attached to the photoplotter) moving and turning the light source on and off. After adding all layers, we previewed the plot from “CAM preview”. We attached a Gerber file (Appendix B.1) in ASCII format from layer 1 just to give you an idea on what a Gerber file looks like.

![Fig. 6 The design artwork in Viewmate (above)](image)
In order to view the plot for every layer, we had to use a program called Lavenir Viewmate. It is a PCB data viewer. We run the layer document from the default CAM directory in the c drive as Gerber files and imported them to Viewmate. Above is part of the design artwork from Viewmate.

Now the photoplot is ready to be sent to the PC board fabricators. This is almost the final stage of the process. After the board is fabricated, we will be soldering the Altera chips, the capacitors and the other devices on the board. Then the test board can be mounted onto the MTCXX board through the two connectors and is ready for testing.

Max +Plus II®

The other program we learned to use was the Altera Multiple Array Matrix Programmable Logic User System or Max +Plus II. It is a great design tool that provides a multiple-platform, architecture-independent design environment that suits the user’s design needs.

This software creates logic designs for Altera Programmable logic devices like Max 5000, Max 7000 and Max 9000. It has a full spectrum of logic capabilities: a variety of design entry, methods of hierarchical designs, powerful logic synthesis, timing-driven compilation, partitioning, functional and timing simulation, linked multiple-device simulation, timing analysis, automatic error location, and device programming and verification. Max +Plus II both reads and writes Altera Hardware Description Language (AHDL) and Very High-speed integrated circuits Description Language (VHDL) files.

As an example, one of the authors (Bekele) used the following equation, \( Y = A*/B+/C \) to illustrate the use of Max Plus II. I used a two input AND gate, a two input OR gate and two inverters. The multiplication operator indicates there is an AND gate, and the addition operator implies there is an OR gate. The negations on B and C call for inverters. A, B, and C are the input variables and Y is the output variable. The inputs are defined to count through the binary numbers 001 to 111 (ABC where A is the most significant bit and C is the least significant bit.) She did the schematic and simulation in Max +Plus II as shown in Appendix B.2.
Here is the truth table for the above function.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>/B</th>
<th>/C</th>
<th>A*/B</th>
<th>A*/B+/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Truth Table for the function $A*/B + /C = Y$

It is the same as the simulation results. For example, where the input was 101 for ABC respectively, the output was 1. On the simulation this was noticed between 400ns to 500ns. The time scale is in increments of 100ns on the waveform editor. There is a small delay when the inputs change and when the output changes to the expected value. If, for example, we look at the time segment from 500 to 600ns, the inputs for C and B are 0 and 1 respectively. The resulting output changes from 1 to its stable value 0 in a short time after the 600ns mark. This is an indication of a glitch in which there is a short time where the output goes through an unsteady state value.

**Quartus II®**

Altera introduced Quartus II after Max Plus II®. It is the advanced form of Max Plus II®. It is a tool that provides the development of devices with high levels of
integration, including designs with entire systems on a single device—systems on a programmable chip.

Quartus II has advanced features such as timing-driven placement, timing-driven routing, enhanced timing analysis, power analysis, and incremental design capabilities.

One of the authors (Bekele) was given the task of importing Max +Plus II assignments to Quartus II settings. It was a timing encoder. She used the import tool in Quartus II. After importing the design, she made sure that the timing encoder family and target devices were correct. The pin assignments were also the same as Max Plus II. The pin numbers are listed on the schematic in Max Plus II whereas in Quartus II they are not, so she had to go to the pin assignment dropdown menu to view the pin numbers to make sure they were assigned correctly. Please look at Appendix B.3. Finally, she checked if the design simulated in Quartus II. The simulation tests the design thoroughly to ensure that it correctly responds to every possible situation before the device is programmed or configured.

**Testing Receivers on Serial Daughter Board**

The same author was assigned to test receivers on a serial daughter board. She plugged the receiver onto the Xbert receiver adapter board and turned on the Xbert and scope. The Xbert is a program that tests the receivers. Normally after Max +Plus II compiler has processed a project, it generates one or more programming files. She opened up the Max +Plus II programming window and verified if the receivers were programmed to the processed project program file, which is the serial daughter board receiver program file.

She programmed the ones that weren’t programmed and proceeded to test the receivers. We have a 10ft and 200ft cables and the test is done through both cables. She let the Xbert run until $10^9$ count, which is the error count. If it runs without errors it is a good receiver, and if error is noticed then the receiver has a problem.
During these eight weeks period we gained a great deal of knowledge from learning to use different software that are very useful in engineering applications to soldering devices on boards. Our work at the University of Arizona Physics lab has given us a tremendous experience and we are very lucky to be working as interns here at the lab. We are very sure that this research experience will open doors for other similar research project intern positions and going forward for our future career.
Acknowledgements

We very much thank Dr. Anthony Pitucco for giving us the opportunity to work as interns this summer.

We also would like to thank James Little, John Lawvere, and Dr. Erwin Sucipto. for introducing us to the different labs and for giving us introductory lectures before getting into this lab.

We would like to extend our appreciation to Dr. William Bickle for teaching us to use different machine shop tools.

We would like to thank Joel Steinberg who has been a great help in guiding us accomplish our project, Dr. Ken Johns, and Chris Leeman. Without their help we wouldn’t have achieved such an experience.