Communication Type Identifying and Processing Probe

We intend to design and build a probe that will be able to interface with devices that have serial ports. For the initial functionality, the probe should be able to discern between JTAG and RS232 protocols but in the future, we plan on expanding the capability of the probe to more protocols such as; SPI, I2C and more. Beyond discerning between the two initial protocols, the probe should be able to read the data coming out of the devices being probed and also be able to send data signals to these devices using the appropriate protocol.

With this kind of access, we will be able to gain insight into the functionality of the device and this could further be used to perform penetration tests as well as reverse engineering of the device.

Given today’s societal dependence on devices that have embedded systems in them, it is paramount that these devices are protected from malicious usage. Technology today, is headed towards a new era known as the Internet of Things (IoT), and with this kind of intercommunication across devices, leakage of information could be detrimental and even fatal. There are a lot of devices today that still have open communication port headers that served as test ports in the design and testing phase. These can provide attackers access to the device’s firmware which could be flashed with modifications made to the functionality of the device.

Future goals of our project would be to determine ways in which these vulnerable devices can be protected from attackers. This would require further analysis into the communication signals being transmitted and how to make them harder to access. This way, the functionality and internal makeup of the device is kept proprietary and has less chances to be exploited.

Due to the uncertainty of target connector pin numbers, we have to consider the greatest possible number of pins which is roughly the sum of RS232 (usually no more than 14 pins) and one or two serial communication pins. Based on this situation, we plan on using a 16:1 multiplexer. Note that voltage level for RS232, JTAG and other protocols varies from 1.8V, 2.5V, 3.3V, 5V and 12V. Our FPGA, however, aiming on detecting the type of unknown input protocol can only receive and transmit signals within chip power level, 3.3V. The solution is to use both comparators and level shifters. Comparator is a device that compares two voltages or currents and outputs a digital signal indicating which is larger. By setting one of the analog input to be reference voltage, we can tell whether the input voltage level reaches that reference voltage or not. Then, 5 parallel connected comparators with a
common reference voltage will be used to determine the exact voltage level of the input. This will work in conjunction with a priority encoder so that if for example, all the digital outputs of comparators are ones except for the 12V input, it can be inferred that the input signal is operating through a 12V protocol. One consequential factor of the comparator is that the sampling rate should be 10 times the input frequency based on the digital control theorem in order to interpret input signals correctly. The level shifters require a DC power supply, connecting the output of the comparator so that digital outputs can be shifted to a desired 3.3V. Besides voltage restriction, the current going through the FPGA should meet the requirements provided by the data sheet, which is no more than 0.5A. After signals are modified to 3.3V and less than 0.5A, the FPGA will execute our program and identify the functions of the connected pins such as GND, DATA IN, DATA OUT, and so on.

**Project Phases:**
Design  
Assembly  
Testing/Debugging  
Synthesis of hardware and software (FPGA)  
Combine with debugger tool  
Retrieve flash from chip on device  
Finalizing project

**Project Timeline:**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>11/11/15</td>
</tr>
<tr>
<td>Assembly</td>
<td>1/29/16</td>
</tr>
<tr>
<td>Testing/Debugging</td>
<td>2/12/16</td>
</tr>
<tr>
<td>Synthesis of Hardware and Software</td>
<td>2/19/16</td>
</tr>
<tr>
<td>Combine Debugger Tool</td>
<td>2/26/16</td>
</tr>
<tr>
<td>Retrieve Flash</td>
<td>3/11/16</td>
</tr>
<tr>
<td>Finalizing Project</td>
<td>4/1/16</td>
</tr>
</tbody>
</table>
**Project Milestones:**
Design finished
Hardware assembled
Hardware testing and debugging
Software coded
Software debugged
Hardware and software combined
Probed tested on devices
Attempt to combine probe with debuggers in order to read data from pins
Use debugger to pull flash from chip on device
Design PCB
Solder parts to PCB
Testing/Debugging of PCB design

**Parts List**

1x 16:1 Mux
   - [Digikey](http://www.digikey.com/scripts/DkSearch/dksus.dll?Detail&itemSeq=182840464&uq=635817543644409324) - 0.92$ ea

3x 8:1 Mux/Demux - [Digikey](http://www.digikey.com/product-detail/en/TC74HC4051APF/TC74HC4051APF-ND/870508) 0.74$ ea

4x Resistors

4x Capacitors

1x Nexys4

8x LM393 - [Digikey](http://www.digikey.com/product-detail/en/LM393AP/296-6609-5-ND/372797) 0.45$ ea