Thank you for attending NIWeek 2017.
Anand Krishnan
Ganesh Devaraj

Soliton Technologies

Architecting a PXI Platform for USB-PD, I2C, and Reliability Testing

anand.krishnan@solitontech.com  ganesh.devaraj@solitontech.com
Agenda

- Session Introduction
- Customer Technical Requirements and ‘Scaling Plan’
- Considerations for PXI Instrument Stackup selection
- Software Scalability and Firmware Regression considerations
- Suggested Solution for Maximum ROI from investment in PXI
- Conclusion
Architecting a validation platform that scales better across Semiconductor Product Lines
Tests:

- Load and Line Sweeps on a Power Regulator
- Dynamic Response of power rails to change in load

Requirement 1:

Product Family:
Low Dropout Regulator
Tests:

- Digital protocol Validation
  - Timing Sweeps on I2C, SPI Lines
  - Voltage Sweeps on I2C, SPI Lines
- Fault tolerance and Recovery of I2C, SPI

Requirement 2:

Product Family:
Low Dropout Regulator
Tests:

- Reliability Test across Temperature, Ionizing Radiation etc
- Lower sample rate continuous acquisition of power rails and digital test signals
- Context based fine logging upon configurable ‘failure events’
Tests:
- Power Supply DC and AC tests
- Timing Response tests to USB_PD Commands
- PD spec compliance power negotiation (CC) lines
- Physical layer tests
- Firmware Validation
Instrument Stackup Decision Points

- List of SMU channels, voltage, current levels, hardware-timed acquisition memory, sample rate
- List of Analog Outputs, Ranges and Update Rate
- List of Analog Inputs, Sample Rate
- Need for synchronization between the SMU channels and the Analog Inputs and Outputs for performing timing measurements
- Granularity of timing skew required for protocol timing characterization
- Voltage level control on protocol digital lines
- Current limit control on digital lines (for control of rise and fall times)
### Short Term versus Long term Considerations

#### Hardware

<table>
<thead>
<tr>
<th>Kind of IO</th>
<th>Short Term Instrument Choice Requirement 1-3</th>
<th>Longer Term instrument Choice Requirements 1 through 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Rails (Input and Output)-+/- 60V, +/-10A pulsed</td>
<td>4139, 4145</td>
<td>4139, 4145 (assuming V,I levels to not change from DUT to DUT)</td>
</tr>
<tr>
<td>Digital protocol lines</td>
<td>8451 (for 1 only)</td>
<td>7965R with 6581 FAM for timing validation and Transceiver FAM for USBPD</td>
</tr>
<tr>
<td></td>
<td>65XX (for 1,2,3)</td>
<td></td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>6115 (for 1 and 3)</td>
<td>7965R with 578X FAM for USBPD and power rail logging</td>
</tr>
<tr>
<td>Analog Outputs</td>
<td>Not needed</td>
<td>7965R with 578X FAM for USBPD</td>
</tr>
</tbody>
</table>
# Short Term versus Long term Considerations

## Software

<table>
<thead>
<tr>
<th>Design Decision</th>
<th>Short Term Design Choice Requirement 1-3</th>
<th>Longer Term Design Choice Requirements 1 through 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUT Control Module</td>
<td>NI-HSDIO based driver with Register Map Module</td>
<td>PXle-7965R + 6581 based driver with Register Map Module</td>
</tr>
<tr>
<td>Buffered AI Acquisition Module</td>
<td>Requirement 1- Finite Multifunction IO/SMU Capture Requirement 3- Buffered MultiFunction IO capture</td>
<td>Buffered continuous stream using 7965R+Transceiver FAM. Not limited by memory on 7965R</td>
</tr>
<tr>
<td>Sequencing Engine</td>
<td>LabVIEW GUI Based</td>
<td>Test Stand Based</td>
</tr>
<tr>
<td>Test Condition Management</td>
<td>Iterator GUI called in LabVIEW</td>
<td>Iterator GUI called in TestStand</td>
</tr>
</tbody>
</table>
Firmware Validation on PMIC/USBPD - Use Case Based

- Firmware exposes bunch of API calls - Each device mode is configured by grouping a set of API calls (Macros)
- In Firmware functional validation, the often-used modes are set up for functional verification (stimulus response)
- A generic test sequencer runs each of these modes and logs data using standard data logging APIs

Firmware Validation- API-regression Based

- The previous approach covers dominant use cases of the firmware but does not cover each parameter combination for each API
- Definition Engines can be built that will auto populate different API input combinations for each API exposed by the firmware (through parsing the API header file)
- User can then select which APIs and parameter combinations to pick and which to ignore
- The resulting definition file can be executed for every release version of the firmware/ APIs
PXI Stackup for MultiFunction Test Platform

I2C DIGITAL INTERFACE VALIDATION, FAULT TOLERANCE

FLEXRIO

SMU

TRANSCEIVER FAM

USB PD VALIDATION

MULTIFUNCTION IO

DIGITIZER FAM

RELIABILITY TEST

RELIABILITY TEST
## Cost Advantages from Forethought in Scaling Needs

<table>
<thead>
<tr>
<th>Approach</th>
<th>Instrument List</th>
<th>Approx HW Cost (USD)</th>
<th>Approx SW Cost (USD)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragmented</td>
<td>4139, 4145, 8451, 65XX, 6115, 79XXR, Transceiver</td>
<td>46,000</td>
<td>50,000</td>
<td>Fully automated, Software Triggered, 1 Real Time FPGA Module without Sequencer leverage</td>
</tr>
<tr>
<td>Consolidated across product lines</td>
<td>4139, 4145, 79XXR(x2), Transceiver, Digitizer FAM</td>
<td>43,000</td>
<td>33,000</td>
<td>Fully Automated, Hardware Triggered, 2 Real Time FPGA Modules with full Sequencer leverage</td>
</tr>
</tbody>
</table>
Questions
Before you go, take the survey.
Stay Connected During and After NIWeek

ni.com/niweekcommunity
facebook.com/National Instruments
twitter.com/niglobal
youtube.com/nationalinstruments

Please provide feedback on this session via the NIWeek Mobile App