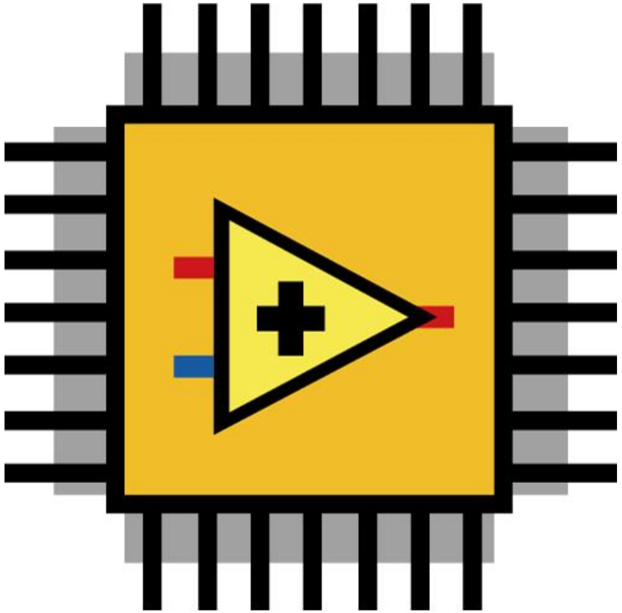
Automated Test of LabVIEW FPGA Code: CI & Jenkins 2 Pipelines

Ching-Hwa Yu
Software Engineering Manager
CLA, CTA, CJE

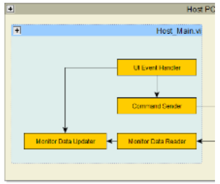
Jianhua Liu
Validation Engineer
CLA, CTD



Incorporating automated tests can significantly accelerate development

Continuous Integration

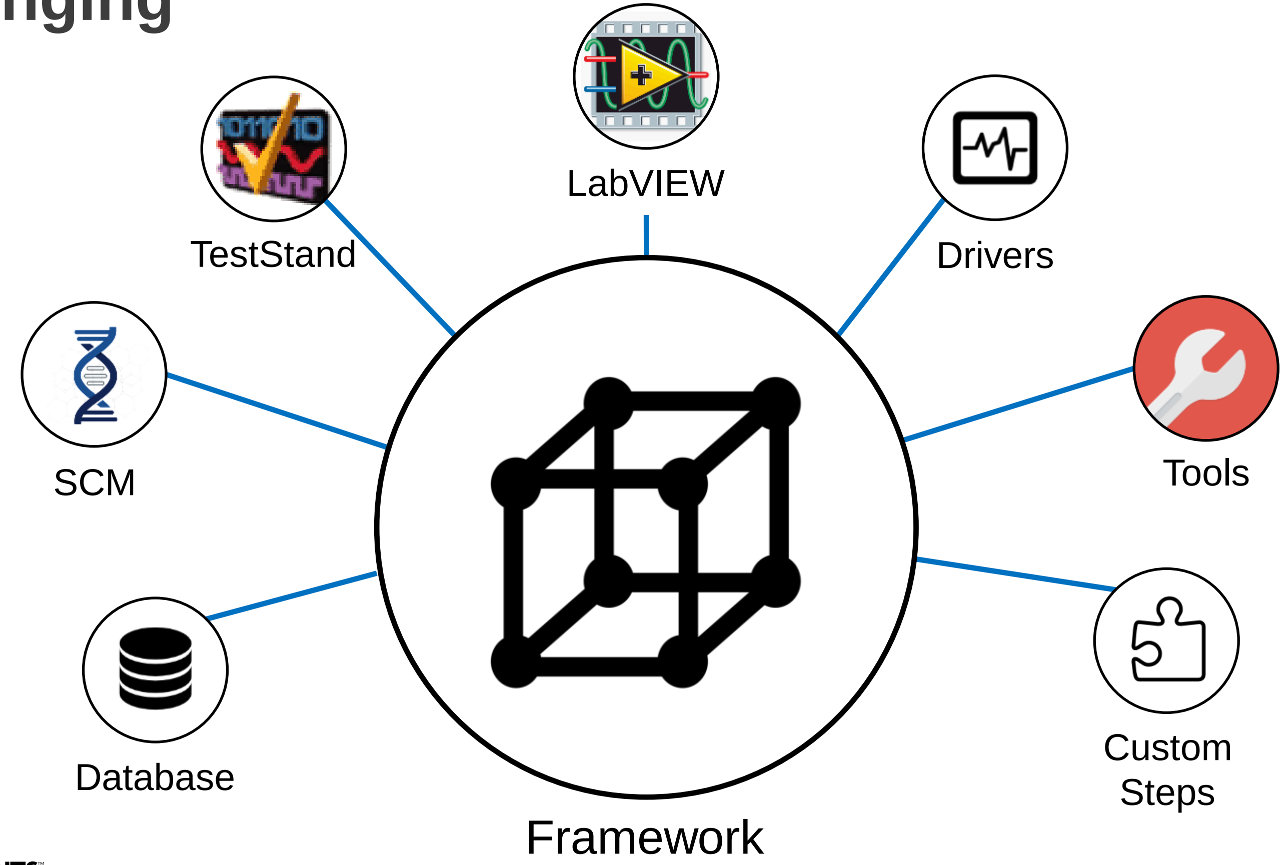
Jenkins Automation



Example: Digital Pattern Generator

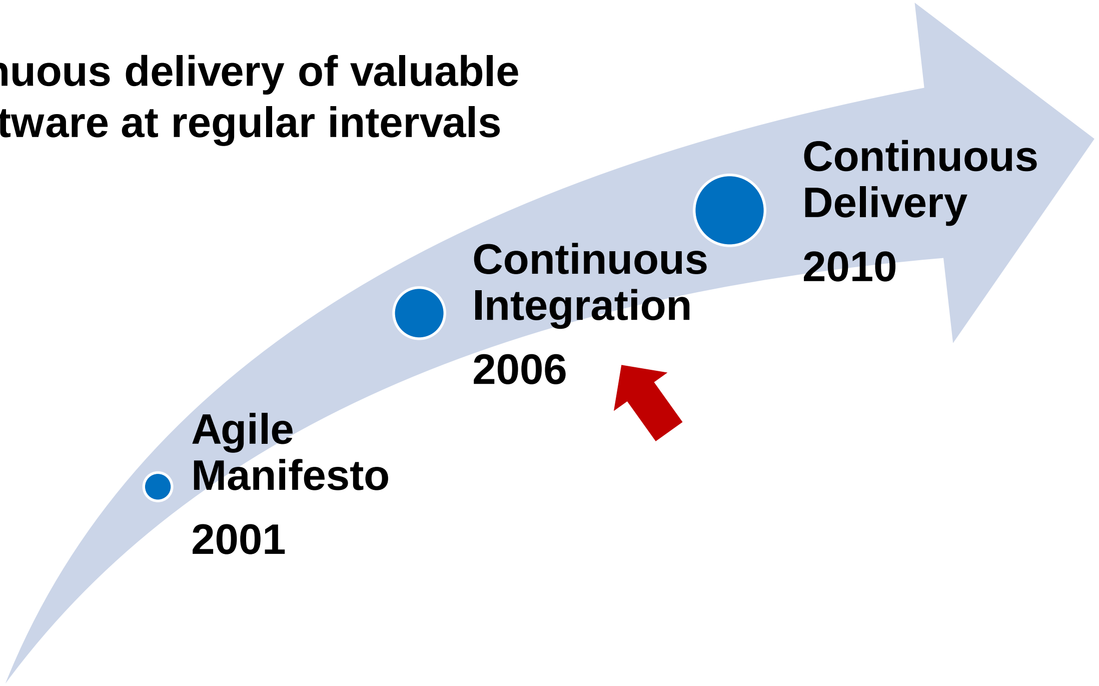
LabVIEW FPGA Test Methodology

Developing software at a fast pace to scale is challenging



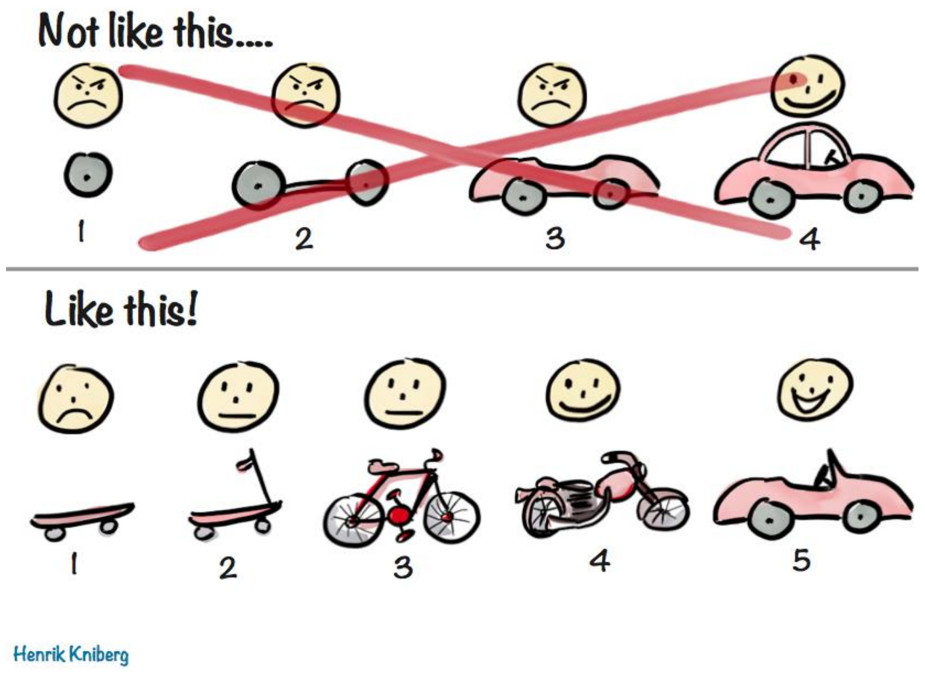
Agile software development has transformed the way software is developed and delivered

Early, continuous delivery of valuable working software at regular intervals

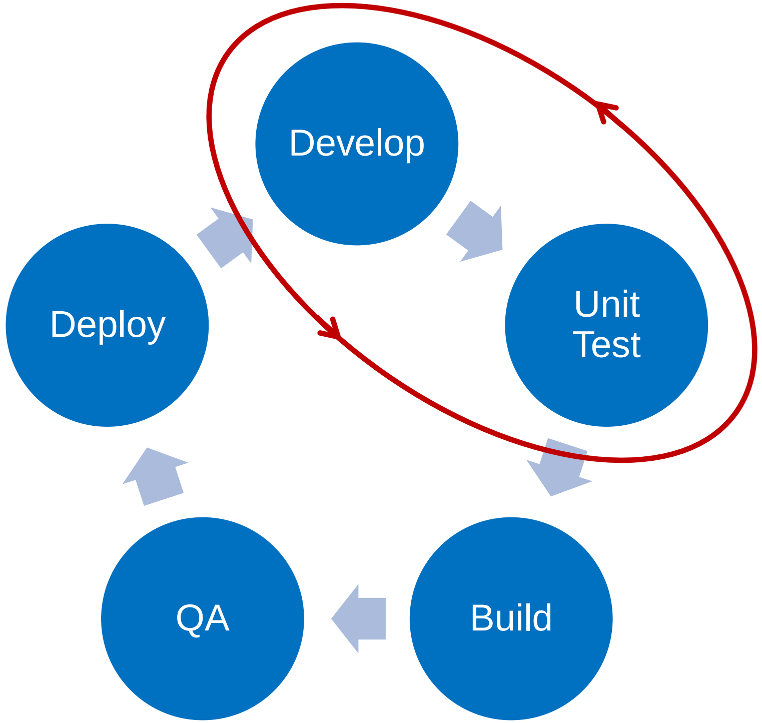


NATIONAL INSTRUMENTS™

The concept of a Minimum Viable Product (MVP) is a way to visualize the Agile process



Software Development Life Cycle (SDLC) for FPGA development includes multiple stages

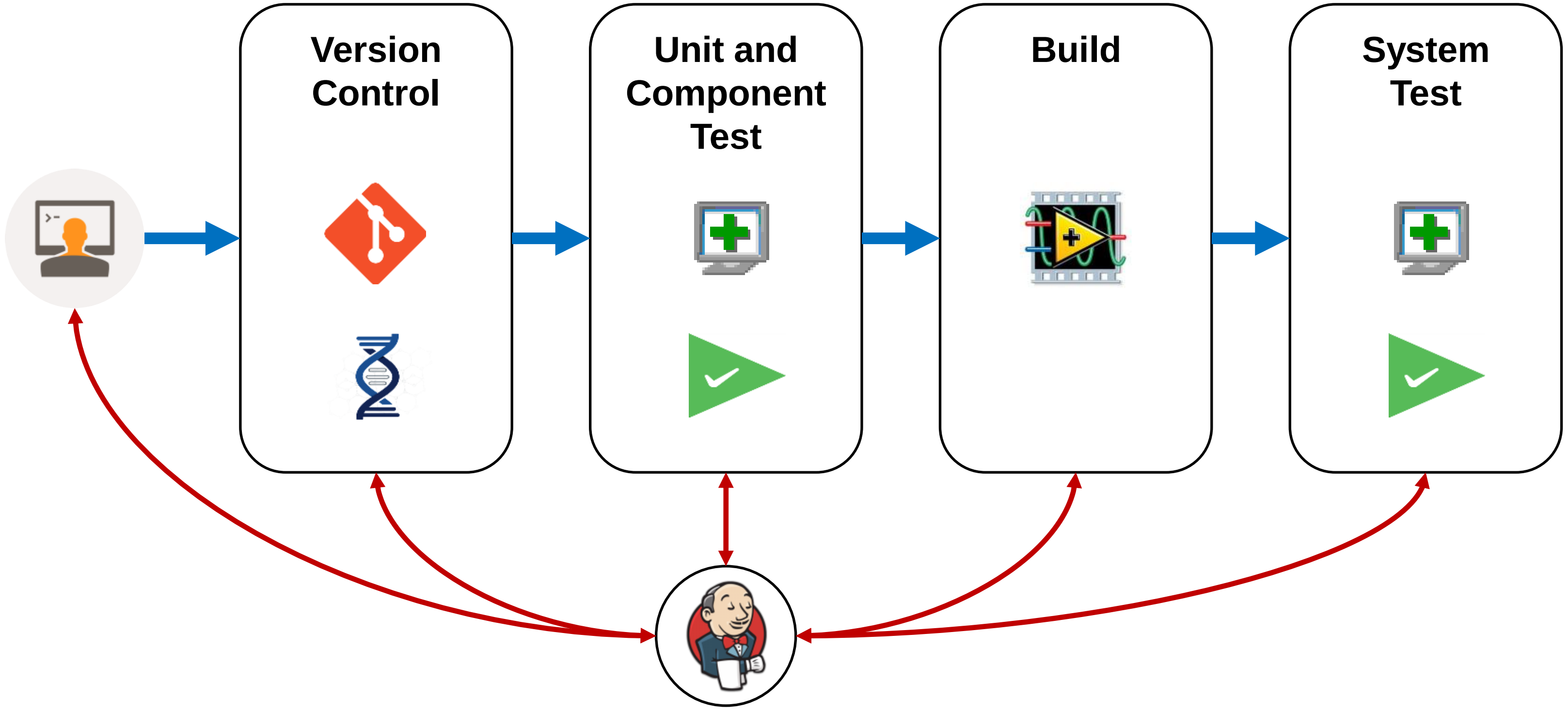


A continuous feedback loop is needed to catch issues before building bit files

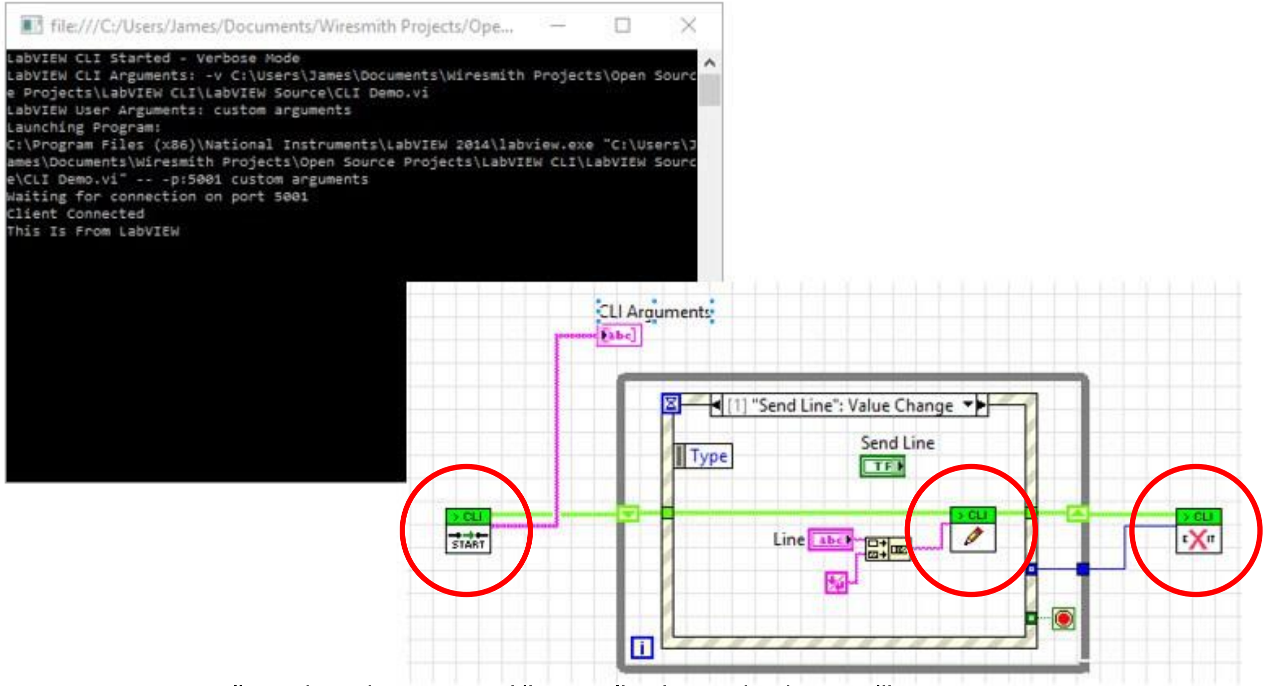
Continuous Integration:

Continuously verify newly integrated code by automating tests to ensure all functionality is working as expected

The steps in testing FPGA software can be automated using a Continuous Integration system called Jenkins



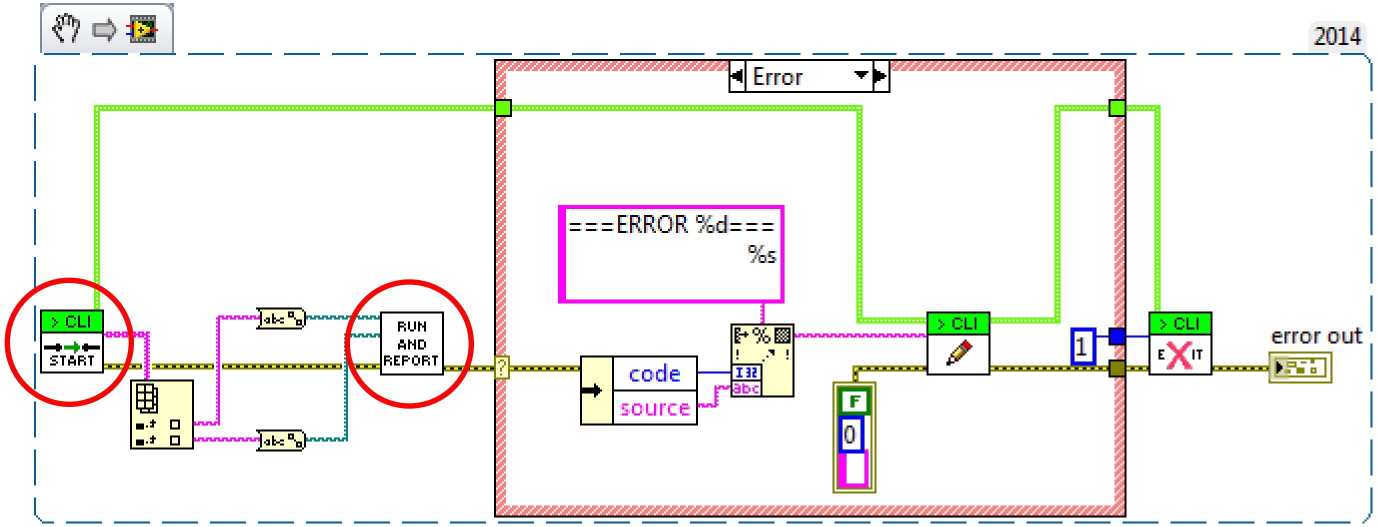
The LabVIEW Command Line Interface (LabVIEW-CLI) can be used to make LabVIEW behave like a real CLI


file:///C:/Users/James/Documents/Wiresmith Projects/Ope...

```
LabVIEW CLI Started - Verbose Mode
LabVIEW CLI Arguments: -v C:\Users\James\Documents\Wiresmith Projects\Open Sourc
e Projects\LabVIEW CLI\LabVIEW Source\CLI Demo.vi
LabVIEW User Arguments: custom arguments
Launching Program:
C:\Program Files (x86)\National Instruments\LabVIEW 2014\labview.exe "C:\Users\J
ames\Documents\Wiresmith Projects\Open Source Projects\LabVIEW CLI\LabVIEW Sourc
e\CLI Demo.vi" -- -p:5001 custom arguments
Waiting for connection on port 5001
Client Connected
This Is From LabVIEW
```

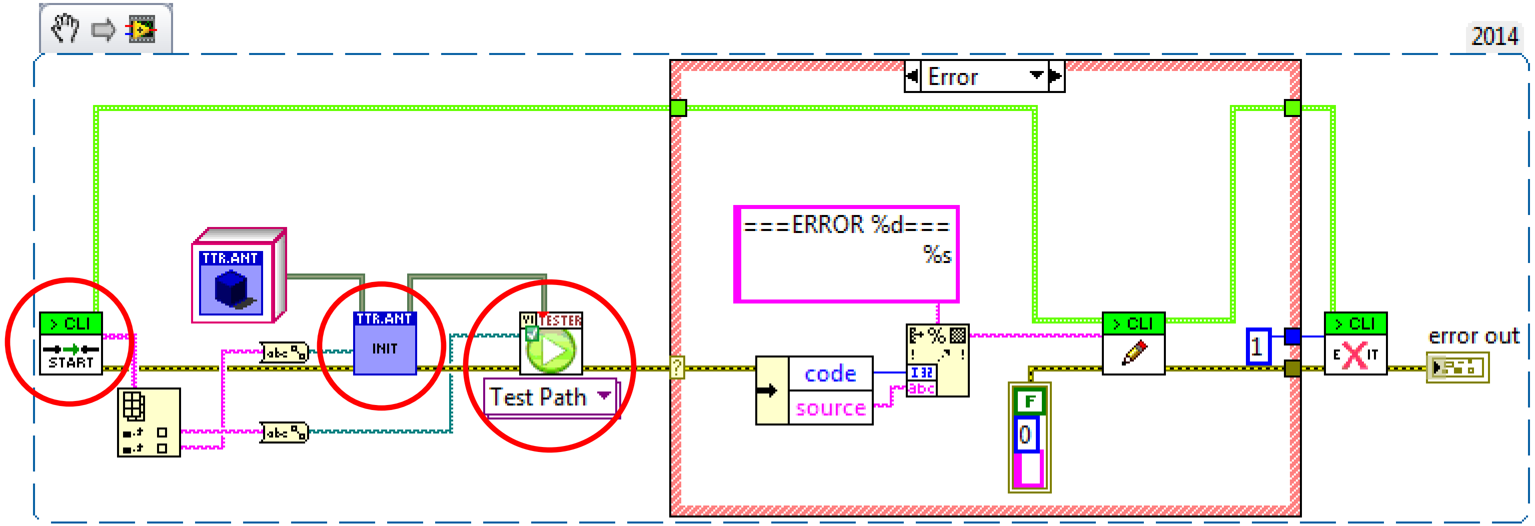
McNally, J. (2016). Command line application and LabVIEW library

The LabVIEW-CLI can be used to automate the launch of NI Unit Test Framework



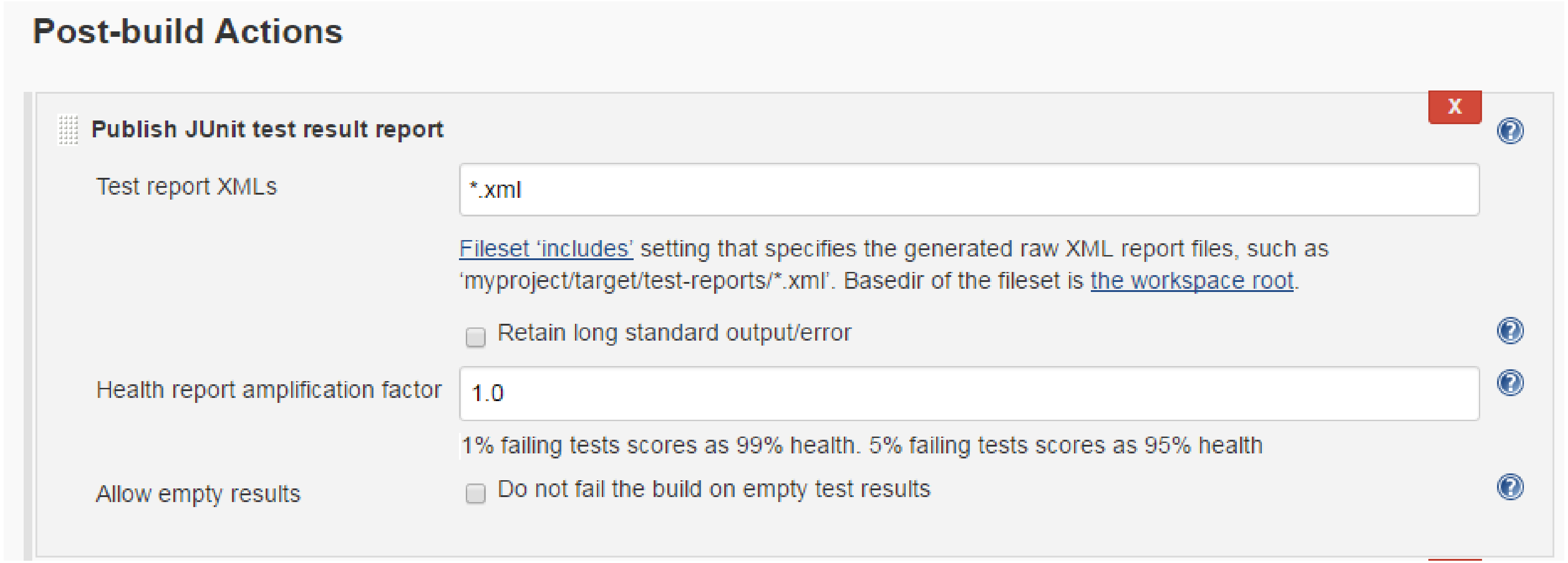
```
>labview-cli --kill --lv-ver 2014 “Run UTF Tests.vi” -- “MyProject.lvproj” “UTF.xml”
```

The LabVIEW-CLI can be used to automate the launch of JKI VI Tester



```
>labview-cli --kill --lv-ver 2014 "Run VI Tester Tests.vi" -- "MyProject.lvproj" "VIT.xml"
```

JUnit Jenkins plugin can be used to consume test results



Post-build Actions

Publish JUnit test result report

Test report XMLs: *.xml

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is the workspace root.

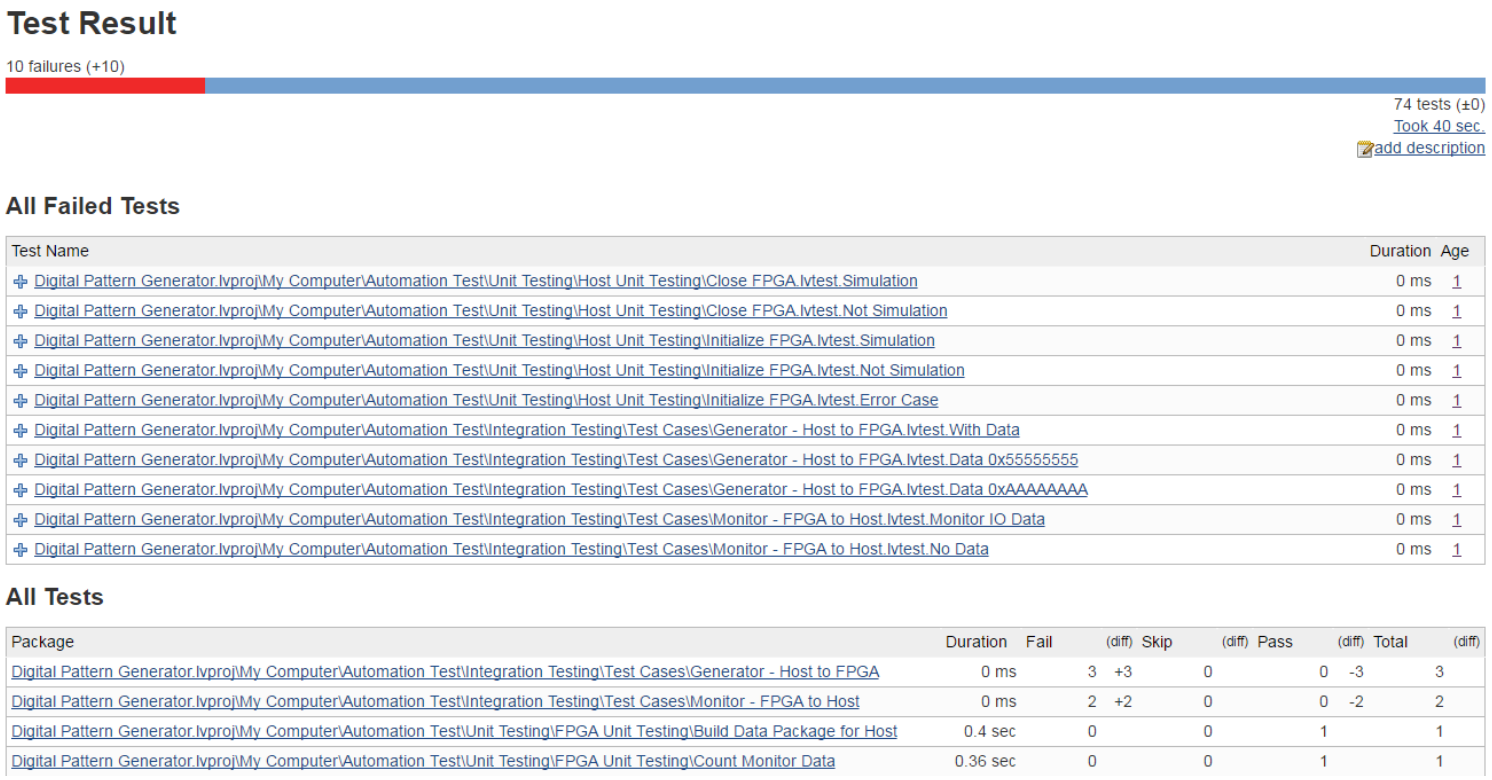
☐ Retain long standard output/error

Health report amplification factor: 1.0

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results: ☐ Do not fail the build on empty test results

JUnit Jenkins plugin can be used to publish test results



Test Result

10 failures (+10)

74 tests (±0)
Took 40 sec.
add description

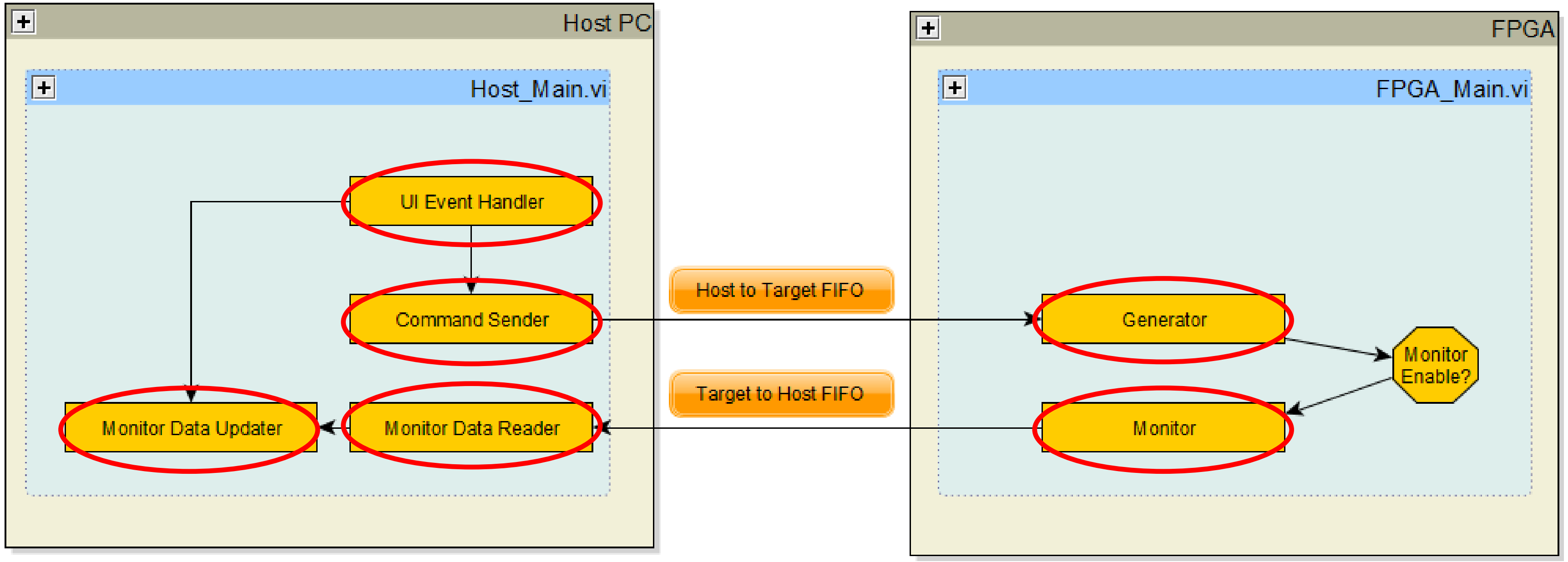
All Failed Tests

Test Name	Duration	Age
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\Host Unit Testing\Close FPGA.lvtest.Simulation	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\Host Unit Testing\Close FPGA.lvtest.Not Simulation	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\Host Unit Testing\Initialize FPGA.lvtest.Simulation	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\Host Unit Testing\Initialize FPGA.lvtest.Not Simulation	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\Host Unit Testing\Initialize FPGA.lvtest.Error Case	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Generator - Host to FPGA.lvtest.With Data	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Generator - Host to FPGA.lvtest.Data 0x55555555	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Generator - Host to FPGA.lvtest.Data 0xAAAAAAAA	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Monitor - FPGA to Host.lvtest.Monitor IO Data	0 ms	1
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Monitor - FPGA to Host.lvtest.No Data	0 ms	1

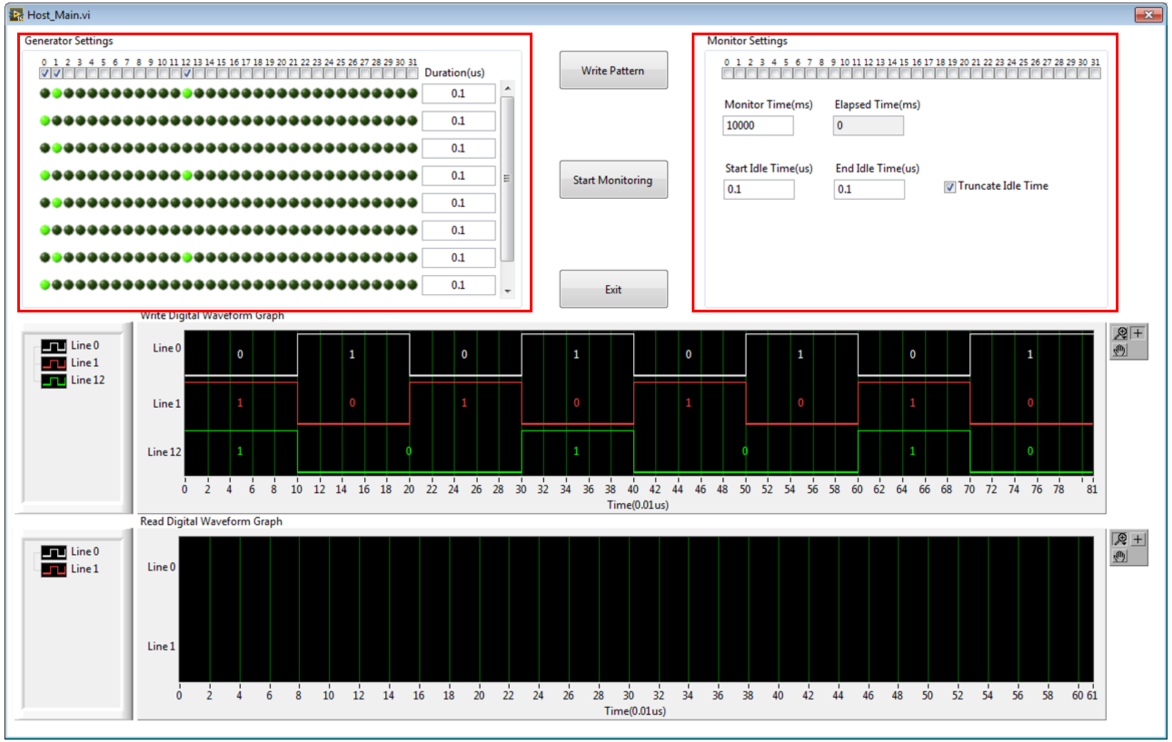
All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Generator - Host to FPGA	0 ms	3	+3	0		0	-3	3	
Digital Pattern Generator.lvproj\My Computer\Automation Test\Integration Testing\Test Cases\Monitor - FPGA to Host	0 ms	2	+2	0		0	-2	2	
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\FPGA Unit Testing\Build Data Package for Host	0.4 sec	0		0		1		1	
Digital Pattern Generator.lvproj\My Computer\Automation Test\Unit Testing\FPGA Unit Testing\Count Monitor Data	0.36 sec	0		0		1		1	

Software testing of an FPGA application can be complex due to the layers involved



Testing ensures that commands and parameters originated on an user interface is properly translated to an FPGA



Host_Main.vi

Generator Settings

Duration(us): 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1

Monitor Settings

Monitor Time(ms): 10000

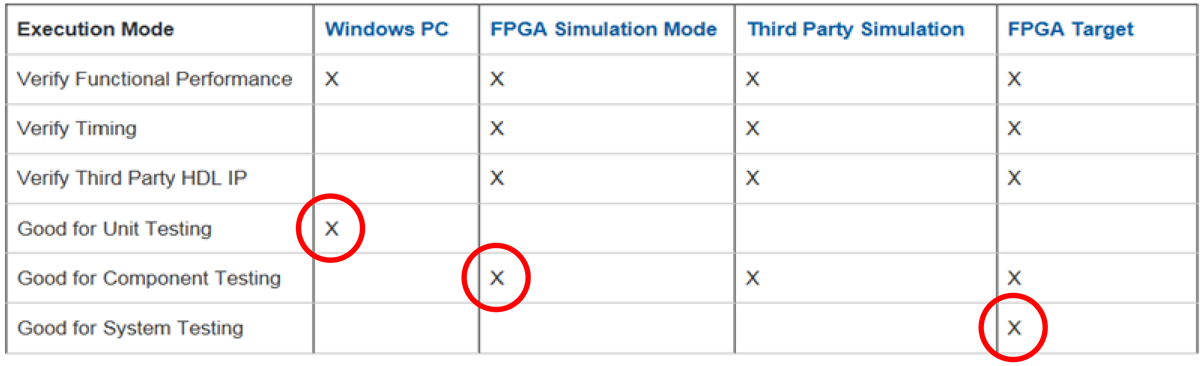
Elapsed Time(ms): 0

Start Idle Time(us): 0.1

End Idle Time(us): 0.1

☑ Truncate Idle Time

The LabVIEW FPGA testing strategy has multiple methods



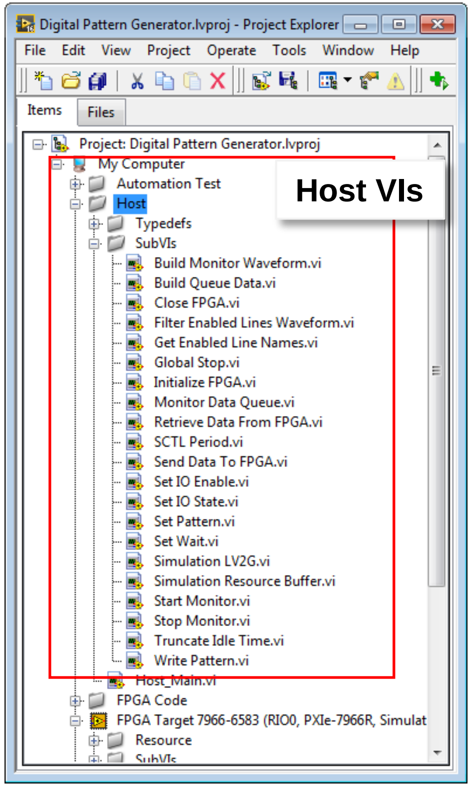
Execution Mode	Windows PC	FPGA Simulation Mode	Third Party Simulation	FPGA Target
Verify Functional Performance	X	X	X	X
Verify Timing		X	X	X
Verify Third Party HDL IP		X	X	X
Good for Unit Testing	X			
Good for Component Testing		X	X	X
Good for System Testing				X

http://www.ni.com/tutorial/51862/en/

Unit Testing	Component Testing	System Testing
Host VIs FPGA VIs	Host and FPGA VIs work together. FPGA VIs execute on Development Computer with Simulated I/O (use Custom VI for FPGA I/O)	Host and FPGA VIs work together. FPGA VIs (bit file) execute on the FPGA Target, real hardware (FPGA Target, Adaptor Module)
FPGA VIs execute on Windows context.		

*This demo will focus on Unit Testing and Component Testing

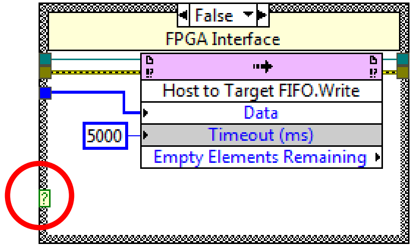
Case structures can be used to bypass FPGA code to test UI related functionality

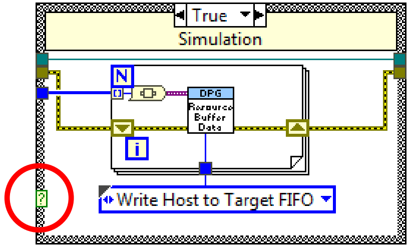

Digital Pattern Generator.lvproj - Project Explorer

File Edit View Project Operate Tools Window Help

Items Files

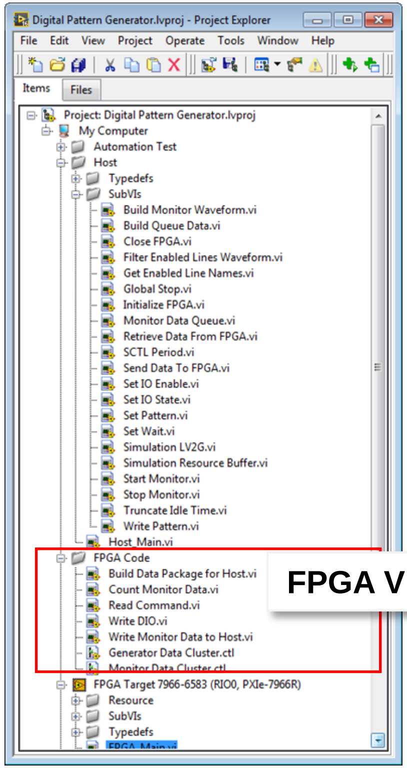
- Project: Digital Pattern Generator.lvproj
  - My Computer
    - Automation Test
    - Host
      - Typedefs
      - SubVIs
        - Build Monitor Waveform.vi
        - Build Queue Data.vi
        - Close FPGA.vi
        - Filter Enabled Lines Waveform.vi
        - Get Enabled Line Names.vi
        - Global Stop.vi
        - Initialize FPGA.vi
        - Monitor Data Queue.vi
        - Retrieve Data From FPGA.vi
        - SCTL Period.vi
        - Send Data To FPGA.vi
        - Set IO Enable.vi
        - Set IO State.vi
        - Set Pattern.vi
        - Set Wait.vi
        - Simulation LV2G.vi
        - Simulation Resource Buffer.vi
        - Start Monitor.vi
        - Stop Monitor.vi
        - Truncate Idle Time.vi
        - Write Pattern.vi
      - Host_Main.vi
    - FPGA Code
    - FPGA Target 7966-6583 (RIO0, PXIe-7966R, Simulat
      - Resource
      - SubVIs





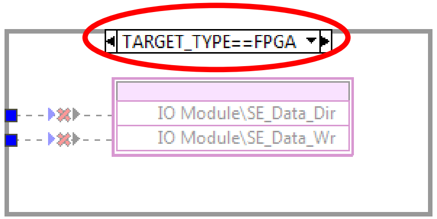
Use Functional Global VI to set and get defined testing data

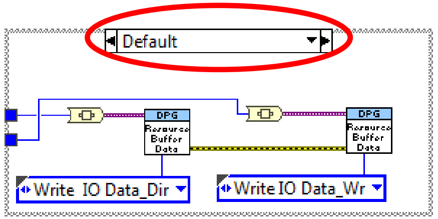
Conditional disable structures can be used to bypass FPGA target specific code



```
Digital Pattern Generator.lvproj - Project Explorer
File  Edit  View  Project  Operate  Tools  Window  Help
Items  Files
Project: Digital Pattern Generator.lvproj
  My Computer
    Automation Test
    Host
      Typedefs
      SubVIs
        Build Monitor Waveform.vi
        Build Queue Data.vi
        Close FPGA.vi
        Filter Enabled Lines Waveform.vi
        Get Enabled Line Names.vi
        Global Stop.vi
        Initialize FPGA.vi
        Monitor Data Queue.vi
        Retrieve Data From FPGA.vi
        SCTL Period.vi
        Send Data To FPGA.vi
        Set IO Enable.vi
        Set IO State.vi
        Set Pattern.vi
        Set Wait.vi
        Simulation LV2G.vi
        Simulation Resource Buffer.vi
        Start Monitor.vi
        Stop Monitor.vi
        Truncate Idle Time.vi
        Write Pattern.vi
      Host_Main.vi
    FPGA Code
      Build Data Package for Host.vi
      Count Monitor Data.vi
      Read Command.vi
      Write DIO.vi
      Write Monitor Data to Host.vi
      Generator Data Cluster.ctl
      Monitor Data Cluster.ctl
    FPGA Target 7966-6583 (RIO0, PXIe-7966R)
      Resource
      SubVIs
      Typedefs
      FPGA_Main.vi
```

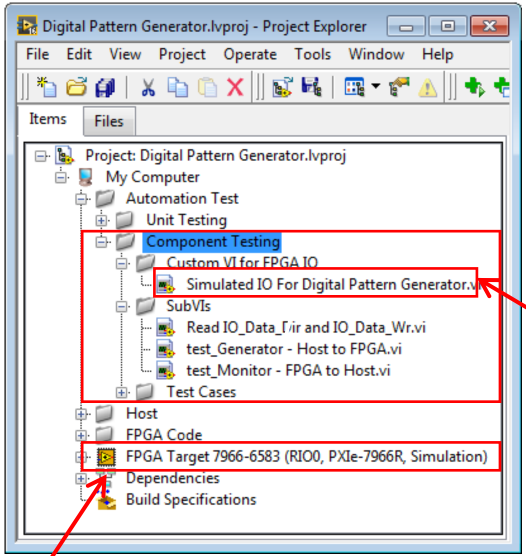
FPGA VIs





Drag VIs from the FPGA target to My Computer

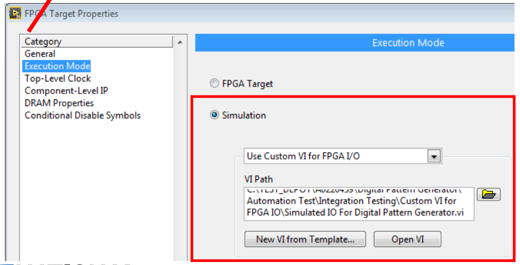
Custom VIs can be used to simulate FPGA IO input and output


Digital Pattern Generator.lvproj - Project Explorer

File Edit View Project Operate Tools Window Help

Items | Files

- Project: Digital Pattern Generator.lvproj
  - My Computer
    - Automation Test
      - Unit Testing
      - Component Testing
        - Custom VI for FPGA IO
          - Simulated IO For Digital Pattern Generator.vi
        - SubVIs
          - Read IO_Data_Dir and IO_Data_Wr.vi
          - test_Generator - Host to FPGA.vi
          - test_Monitor - FPGA to Host.vi
        - Test Cases
    - Host
    - FPGA Code
    - FPGA Target 7966-6583 (RIO0, PXIe-7966R, Simulation)
    - Dependencies
    - Build Specifications


FPGA Target Properties

Category
- General
- Execution Mode
- Top-Level Clock
- Component-Level IP
- DRAM Properties
- Conditional Disable Symbols

Execution Mode

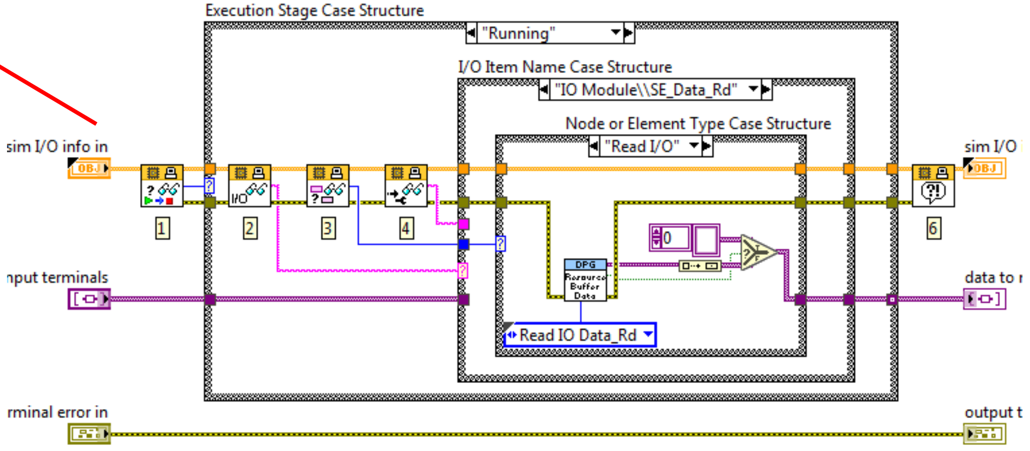
FPGA Target [✓ Simulation]

Use Custom VI for FPGA I/O

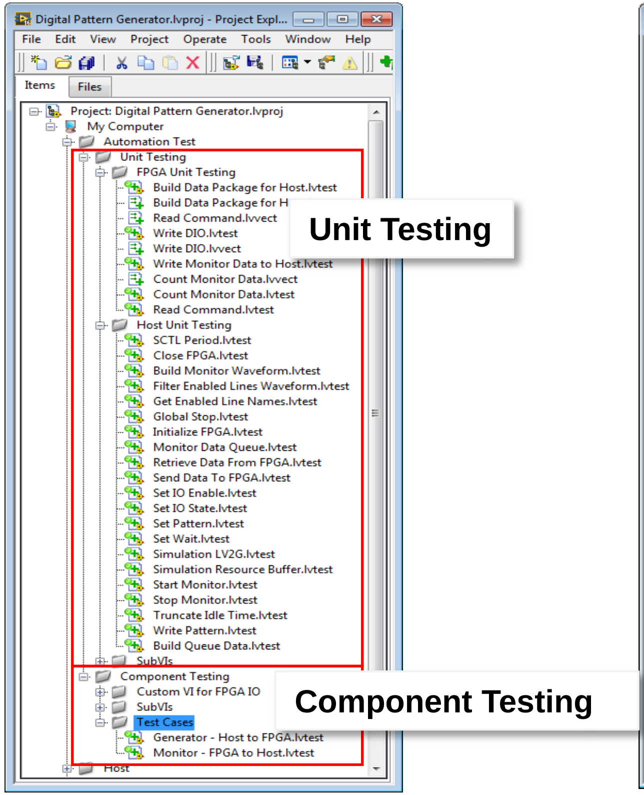
VI Path: C:\TEST_DEPOT\A0220459\Digital Pattern Generator\Automation Test\Integration Testing\Custom VI for FPGA IO\Simulated IO For Digital Pattern Generator.vi

New VI from Template... | Open VI

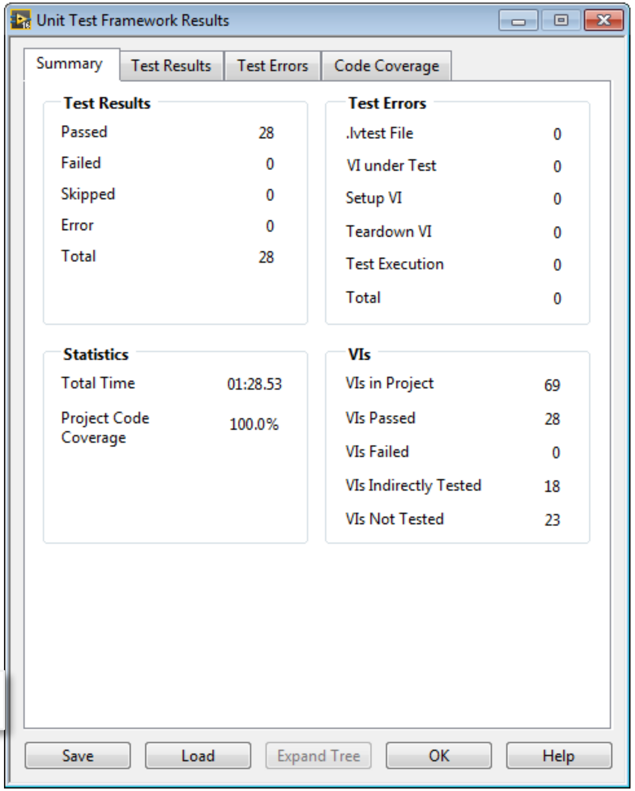
Create Custom VI for FPGA I/O, use Functional Global to get and set defined testing data



NI Unit Test Framework can be used to execute unit and component test cases


Digital Pattern Generator.lvproj - Project Expl...
File Edit View Project Operate Tools Window Help
Items Files
Unit Testing
Component Testing

- Project: Digital Pattern Generator.lvproj
  - My Computer
    - Automation Test
      - Unit Testing
        - FPGA Unit Testing
          - Build Data Package for Host.lvtest
          - Build Data Package for H
          - Read Command.lvvect
          - Write DIO.lvtest
          - Write DIO.lvvect
          - Write Monitor Data to Host.lvtest
          - Count Monitor Data.lvvect
          - Count Monitor Data.lvtest
          - Read Command.lvtest
        - Host Unit Testing
          - SCTL Period.lvtest
          - Close FPGA.lvtest
          - Build Monitor Waveform.lvtest
          - Filter Enabled Lines Waveform.lvtest
          - Get Enabled Line Names.lvtest
          - Global Stop.lvtest
          - Initialize FPGA.lvtest
          - Monitor Data Queue.lvtest
          - Retrieve Data From FPGA.lvtest
          - Send Data To FPGA.lvtest
          - Set IO Enable.lvtest
          - Set IO State.lvtest
          - Set Pattern.lvtest
          - Set Wait.lvtest
          - Simulation LV2G.lvtest
          - Simulation Resource Buffer.lvtest
          - Start Monitor.lvtest
          - Stop Monitor.lvtest
          - Truncate Idle Time.lvtest
          - Write Pattern.lvtest
          - Build Queue Data.lvtest
        - SubVIs
      - Component Testing
        - Custom VI for FPGA IO
        - SubVIs
        - Test Cases
          - Generator - Host to FPGA.lvtest
          - Monitor - FPGA to Host.lvtest
    - Host



Test Results

Passed	28
Failed	0
Skipped	0
Error	0
Total	28

Test Errors

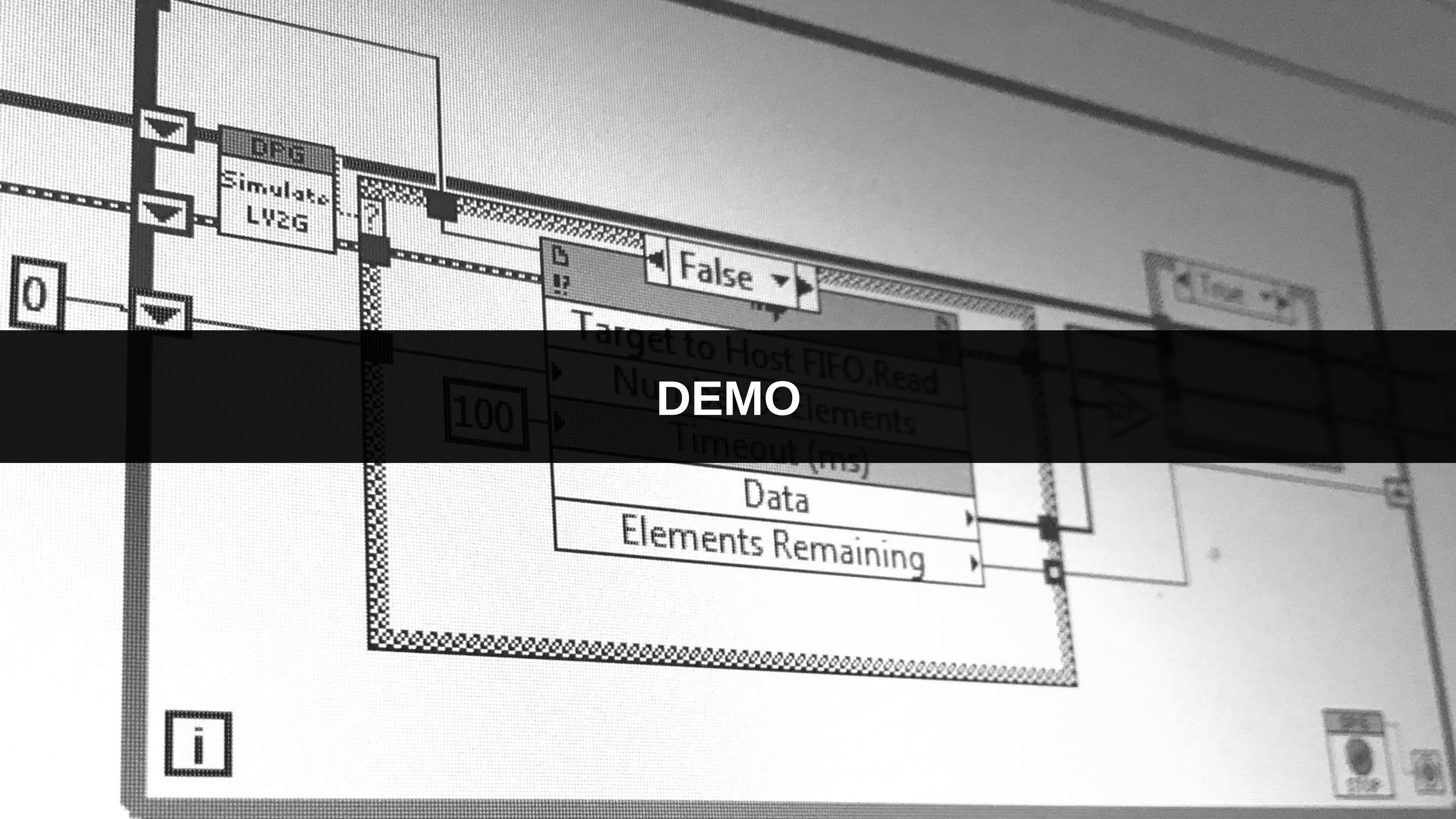
.lvtest File	0
VI under Test	0
Setup VI	0
Teardown VI	0
Test Execution	0
Total	0

Statistics

Total Time	01:28.53
Project Code Coverage	100.0%

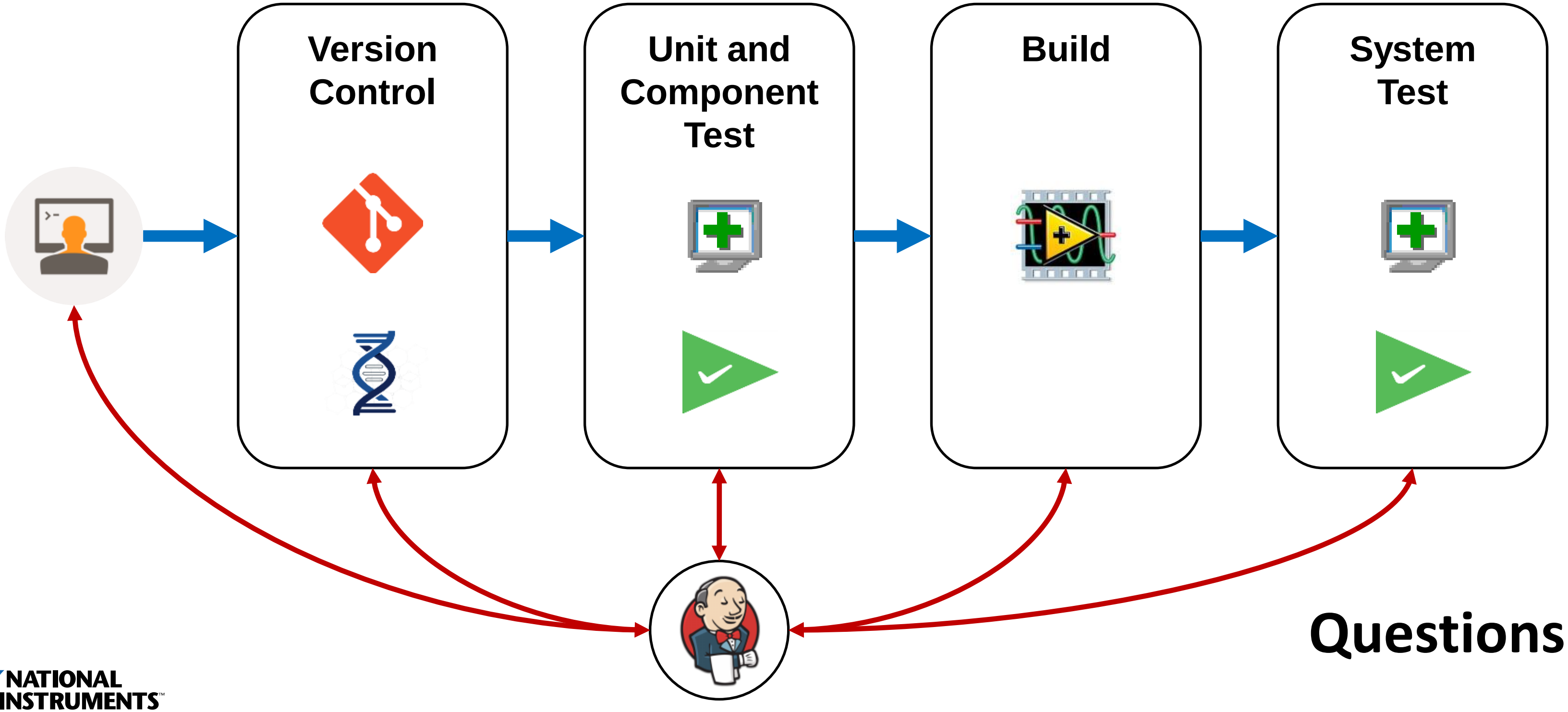
VIs

VIs in Project	69
VIs Passed	28
VIs Failed	0
VIs Indirectly Tested	18
VIs Not Tested	23


DFG
Simulate LV2G
0
False
Target to Host FIFO.Read
Timeout (ms)
100
Data
Elements Remaining
i

DEMO

In summary, combining test automation with the right test methodology can significantly accelerate the development and increase code quality



Questions?

Resources to help build your CI system

- Continuous Integration by Paul Duvall, Steve Matyas, and Andrew Glover http://www.amazon.com/dp/0321336380
- Command Line Tools - https://github.com/chinghwayu/Command-Line-Tools
- My Blog at http://chinghwayu.com
- Continuous Integration User Group https://decibel.ni.com/content/groups/continuous-integration-in-labview
- LabVIEW-CLI – https://github.com/JamesMc86/LabVIEW-CLI
- JUnit Results API – https://github.com/NISystemsEngineering/LV-JUnit
- NI UTF to JUnit – https://github.com/LabVIEW-DCAF/UTF-Test
- JKI Software – https://github.com/JKISoftware