

Management *Sans* Frontiers

Removing the Roadblocks in FPGA Projects

Chris Higgs

POTENTIAL
VENTURES

What is a Roadblock?

*Something that **hinders** your **organisation**
specifically in terms of
development*

Simplistic Reduction

*The problems and roadblocks faced are either **easy** or **hard**.*

EASY Problems

already solved

*just add [**time** | **money**]*

Easy Problems: *examples*

- Revision control
- Regression testing and **automation**
- SW/HW backwards **compatibility**
- Doc / SW / HW / Verif **consistency**
- Self-describing hardware
- Release process and **version** management
- Issue **tracking** and support

Easy Problems: *impact*

- Development speed
- Reluctance / inability to change
- Complexity glass ceiling
- Release and maintenance overhead
- Engineer happiness

***HARD* Problems**

difficult to address

Require ***willingness to change***

*not just [**time** | **money**]*

Hard Problems: *examples*

- Accurate picture of:
 - Wasted effort
 - Technical debt
 - Efficiency
 - Progress vs plan / expectations

Very difficult to measure!

Hard Problems: *examples*

- Organisational structure:
 - Decision making hierarchy
 - Appropriate design input
 - Cross-team collaboration
 - Minimising duplication of effort
 - Flexibility and agility

Hard Problems: *examples*

- Responding effectively to:
 - Changing requirements
 - Market evolution
 - New technologies
 - Increased / more accurate information

Problem Space: Dependencies

Methodology



Organisation

Tools

Problem Space: Solution

- Data-driven optimisation
- Engineer the process
- Inspect and adapt
- Experiment and customise

Efficient Comms: Cache Theory

- Maximise value of *informal* comms
- Exploit human *temporal locality*
- Reduce overhead of communication

Dream Team

- Architecture and Design
- Hardware
- Verification and test
- Software
- Scripting and automation

Team Communication

- Implicit and Informal channels
 - High bandwidth
- Rapid feedback paths
- Self-organising

Challenges: Tools

- Source Control
 - Feature branches, code review
 - Documents
- Co-simulation
- Automation
 - Shared definitions
 - Regressions

Challenges: Conflicts

- “Get it done” vs. “not my problem”
- Need a *lingua franca*
 - Python and C?
- Willingness to learn new skills
- Training

Data Driven: Metrics

- Code edits
- Build times
- Block level fmax, registers, LUTs
- Coverage of hardware and software
- Licence usage
- Tool warnings
- Bug rates

Example: From Scratch

```
git init
```

```
cookiecutter $TEMPLATE
```

```
git add $COMPONENT
```

```
git commit -m "Initial commit"
```

Example: Output

- documentation
 - component_datasheet.rst
- rtl
 - component.sv
- simulation
 - testbench.py
 - bringup_test.py
- software
 - component_hal.cpp
 - component_hal.h

Automation in Action

The screenshot shows the Jenkins Job Generator configuration interface. The top navigation bar includes the Jenkins logo, a search bar, and the breadcrumb path: Jenkins > job_template_rtl > configuration. On the left sidebar, there are navigation links: Back to Dashboard, Status, Changes, Workspace, Build with Parameters, Delete Job Generator, Configure, Plots, and Git Polling Log. The main content area is titled 'Job Generator job_template_rtl' and contains the following sections:

- Global Generation Parameters:**
 - repository:** A text input field containing 'file:///demos/aldec/.git'. Below it, a tooltip shows the path format: 'Path to the git repository. file:///path/to/some/local/repo/.git/ git@github.com/some/repo.git http://github.com/some/repo.git'.
 - component:** A text input field containing 'example'. Below it, a tooltip shows: 'Name of the component'.
- Options:**
 - Process this job only
 - Disable jobs
 - Delete jobs

At the bottom of the configuration area is a blue 'Generate' button. To the right of the options are three help icons (question marks). On the left sidebar, the 'Build History' section shows a list of recent builds:

Build Number	Timestamp
#32	Jul 16, 2015 5:00 PM
#31	Jul 16, 2015 2:15 PM
#30	Jul 16, 2015 12:29 PM
#29	Jul 16, 2015 10:41 AM
#28	Jul 15, 2015 6:52 PM
#27	Jul 15, 2015 6:46 PM
#26	Jul 15, 2015 6:44 PM
#25	Jun 11, 2015 12:19 AM
#24	Jun 10, 2015 9:06 PM
#23	Jun 10, 2015 1:16 AM
#22	Jun 10, 2015 12:07 AM
#21	Jun 9, 2015 11:18 PM
#20	Jun 8, 2015 11:44 PM

Output: Regression

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Plots

Datasheet

Coverage Report

Git Polling Log

Build History trend

#1 Jul 16, 2015 7:15 PM

RSS for all RSS for failures

Project rtl_example

add description

Disable Project

Datasheet

Coverage Report

Workspace

Recent Changes

Latest Test Result (no failures)

Permalinks

- [Last build \(#1\), 3 min 30 sec ago](#)
- [Last stable build \(#1\), 3 min 30 sec ago](#)
- [Last successful build \(#1\), 3 min 30 sec ago](#)

Output: Datasheet

[Back to rtl_example](#) [index](#)

[Zip](#)

Example 0.0.1 documentation »

[index](#)



Datasheet index

Example Datasheet

RTL Interfaces and Ports

Clock: clk

Reset: reset

Streaming Sink: stream_in

Avalon-ST Properties

Avalon-ST Signals

Waveform diagrams

Memory Mapped Slave: mm_interface

Avalon-MM Signals

Waveform diagrams

Register Definitions

Software API

This Page

[Show Source](#)

Quick search

[Go](#)

Enter search terms or a module, class or function name.

Example Datasheet

This is a component created during a live demo

RTL Interfaces and Ports

Clock: clk

Clock input. All interfaces are synchronous to this clock.

Reset: reset

For active low reset, define the macro `ARCH_ACTIVE_LOW_RESET`.

For asynchronous reset (synchronously de-asserted), define the macro `ARCH_RESET_IS_ASYNCRONOUS`.

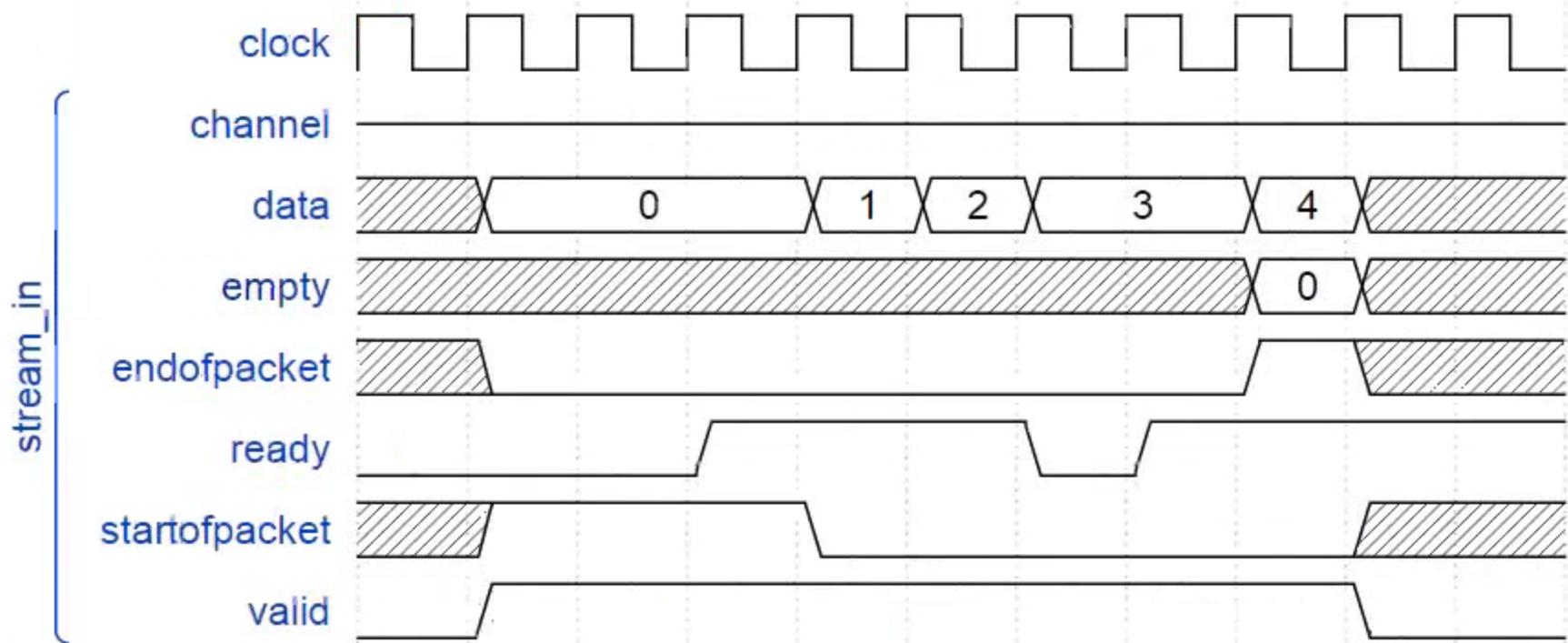
Streaming Sink: stream_in

This is a packetised streaming point-to-point data interface.

Supported interface formats:

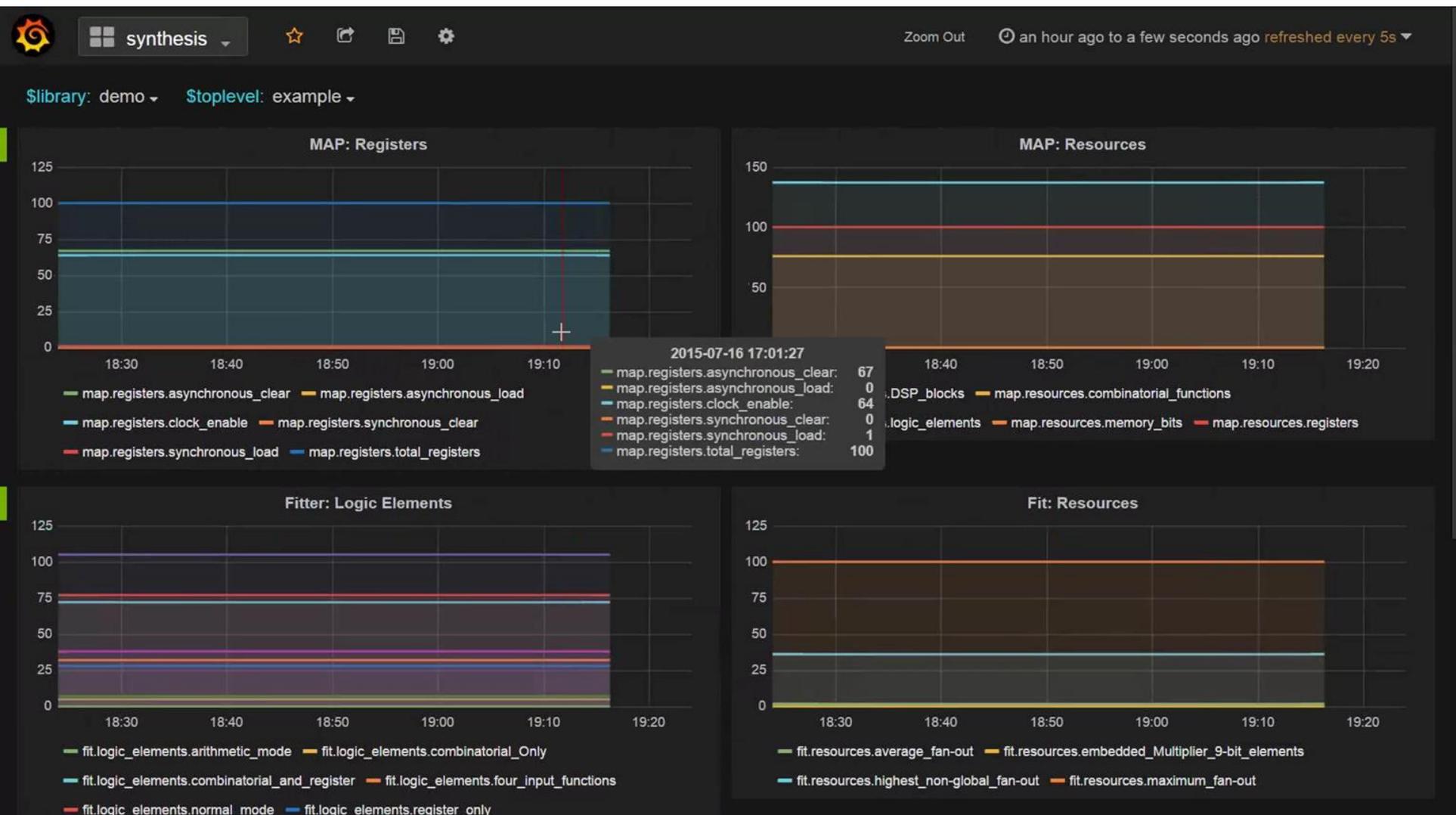
Output: Waveforms

Example of transaction with backpressure:



Avalon-ST with backpressure

Output: Metrics



Summary: 30 seconds

- New component skeleton
- Regression
 - Simulations
 - Synthesis
 - Datasheet
- Metrics, coverage, co-simulation
- Versioned, self-describing, compatible

Rapid FPGA Development

- Eminently possible
- Requires tight SW/HW integration
 - Organisational changes
 - Improved tooling and methodology
 - Wider skills base, blurrier lines
- Evolution isn't optional



Thank You

Questions?

POTENTIAL
VENTURES