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

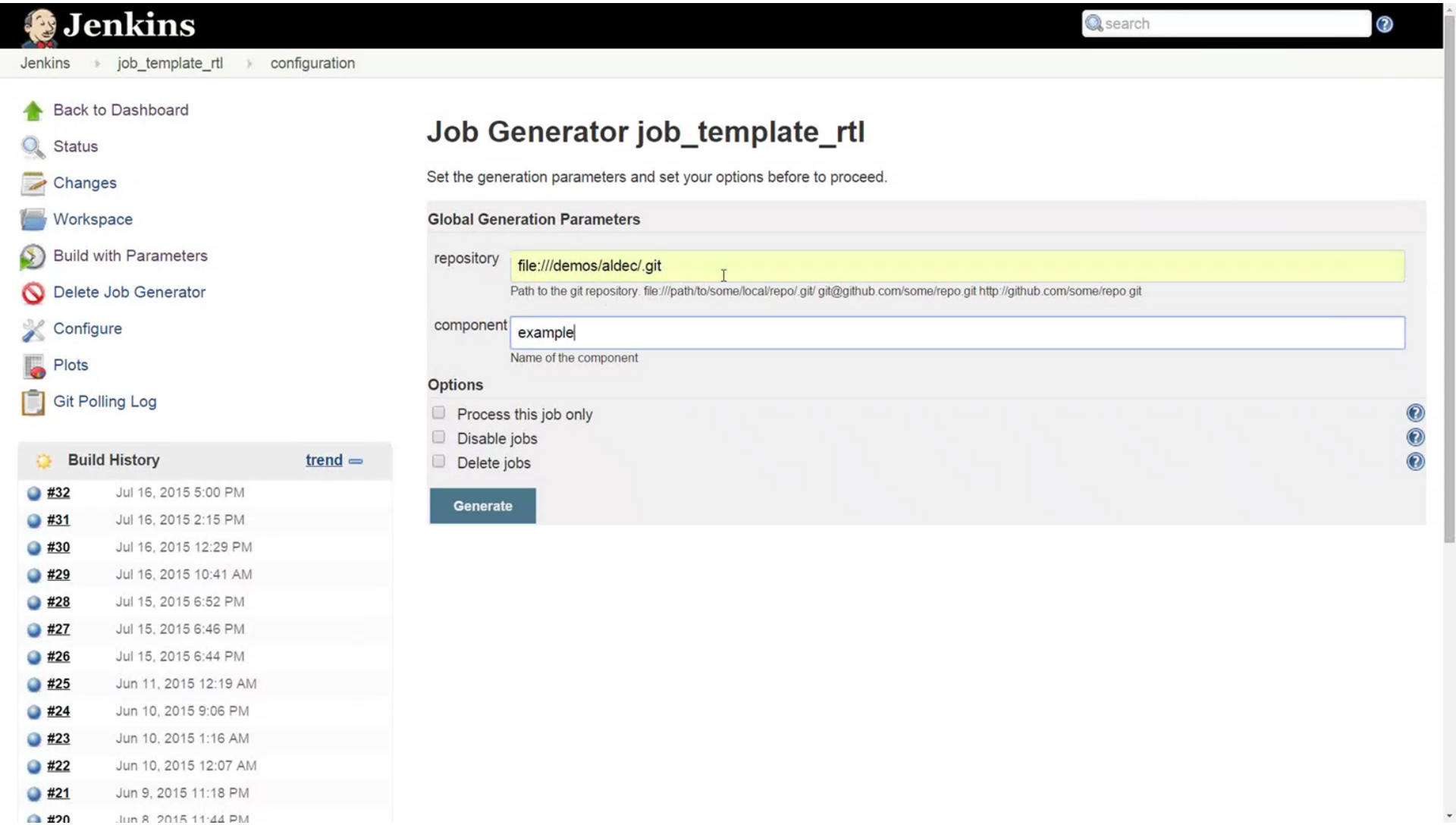
Example: From Scratch

```
git init

cookiecutter $TEMPLATE

git add $COMPONENT

git commit -m “Initial commit”
```

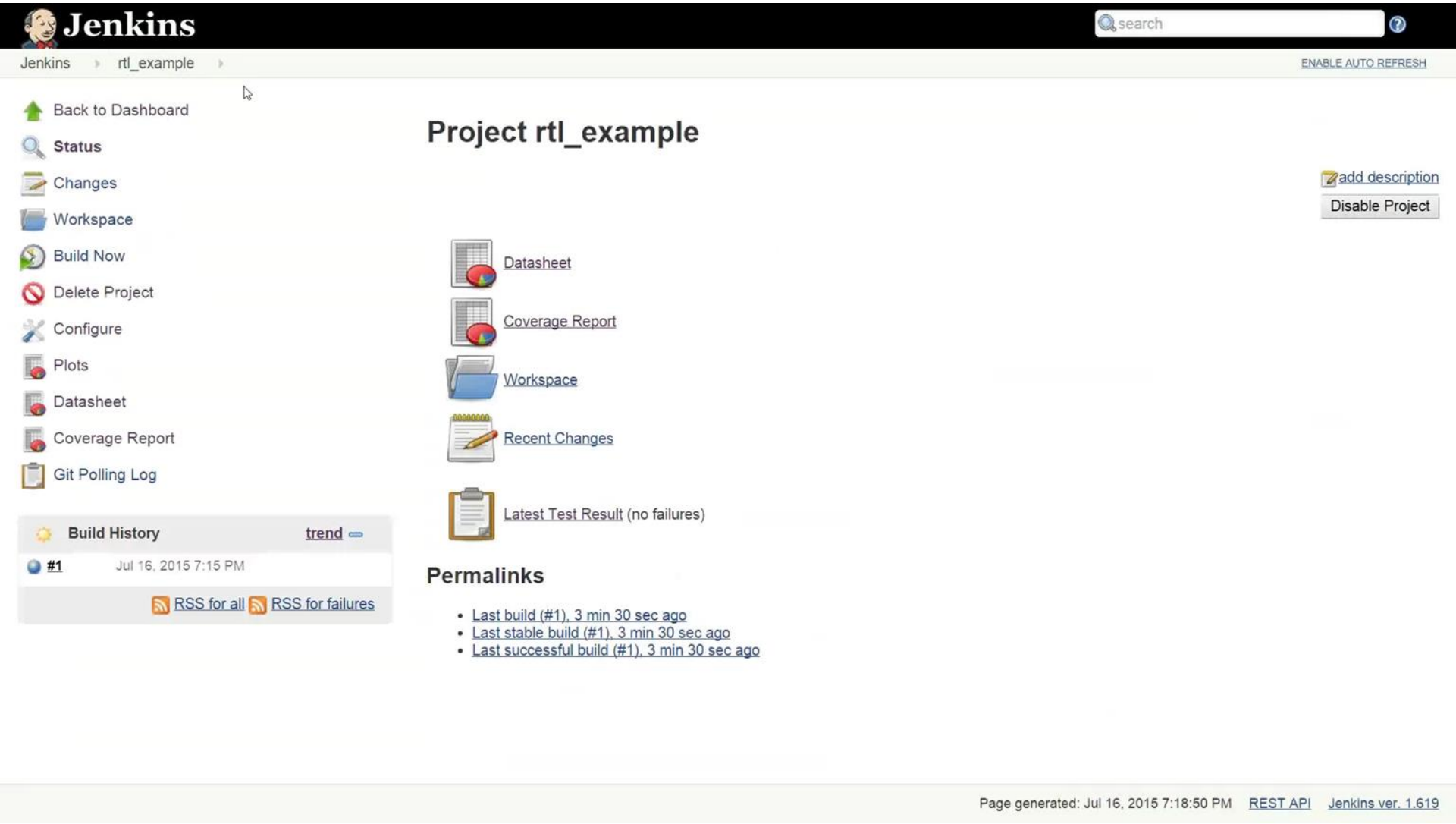
Example: Output

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

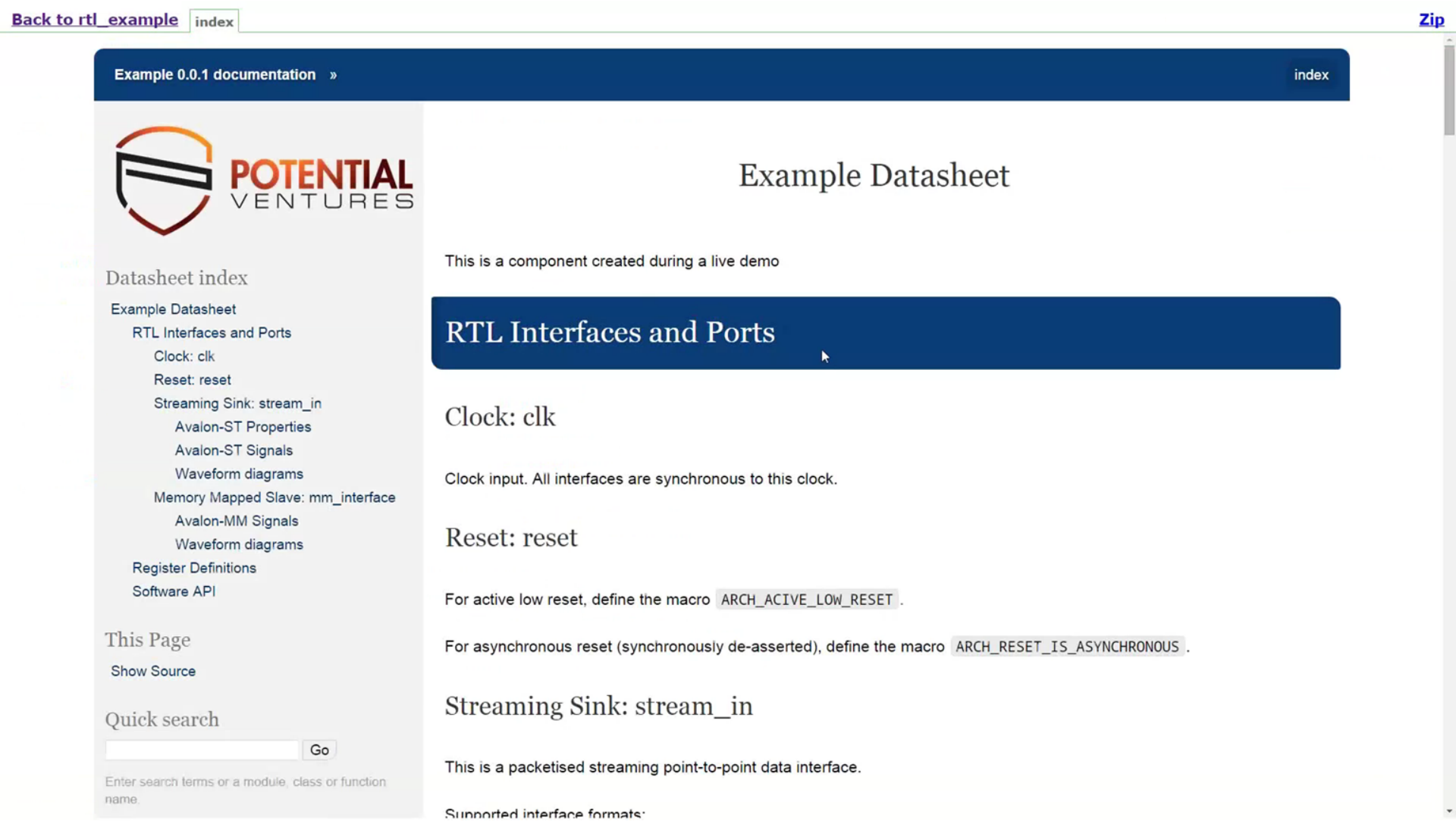
Automation in Action



Output: Regression



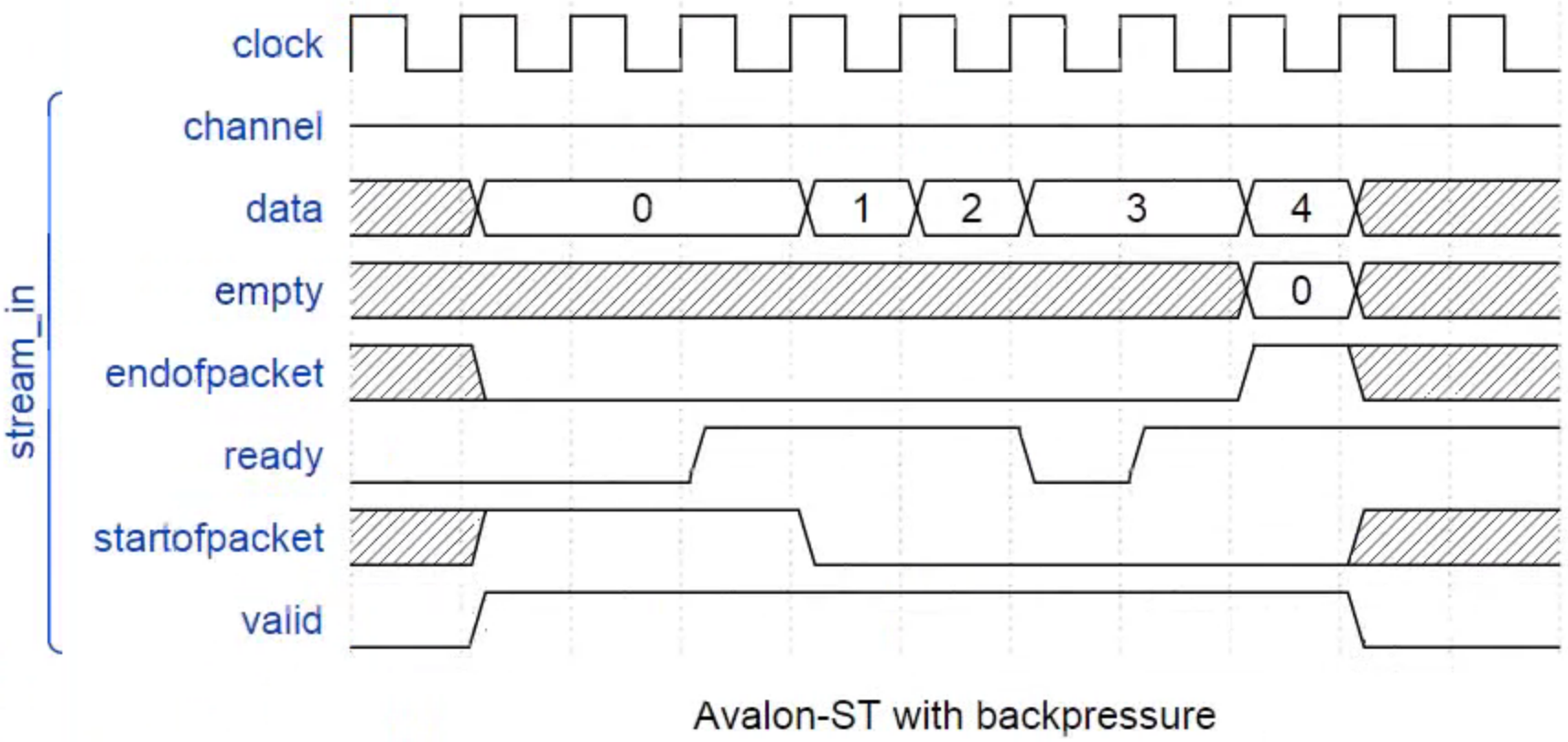
Output: Datasheet



POTENTIAL VENTURES

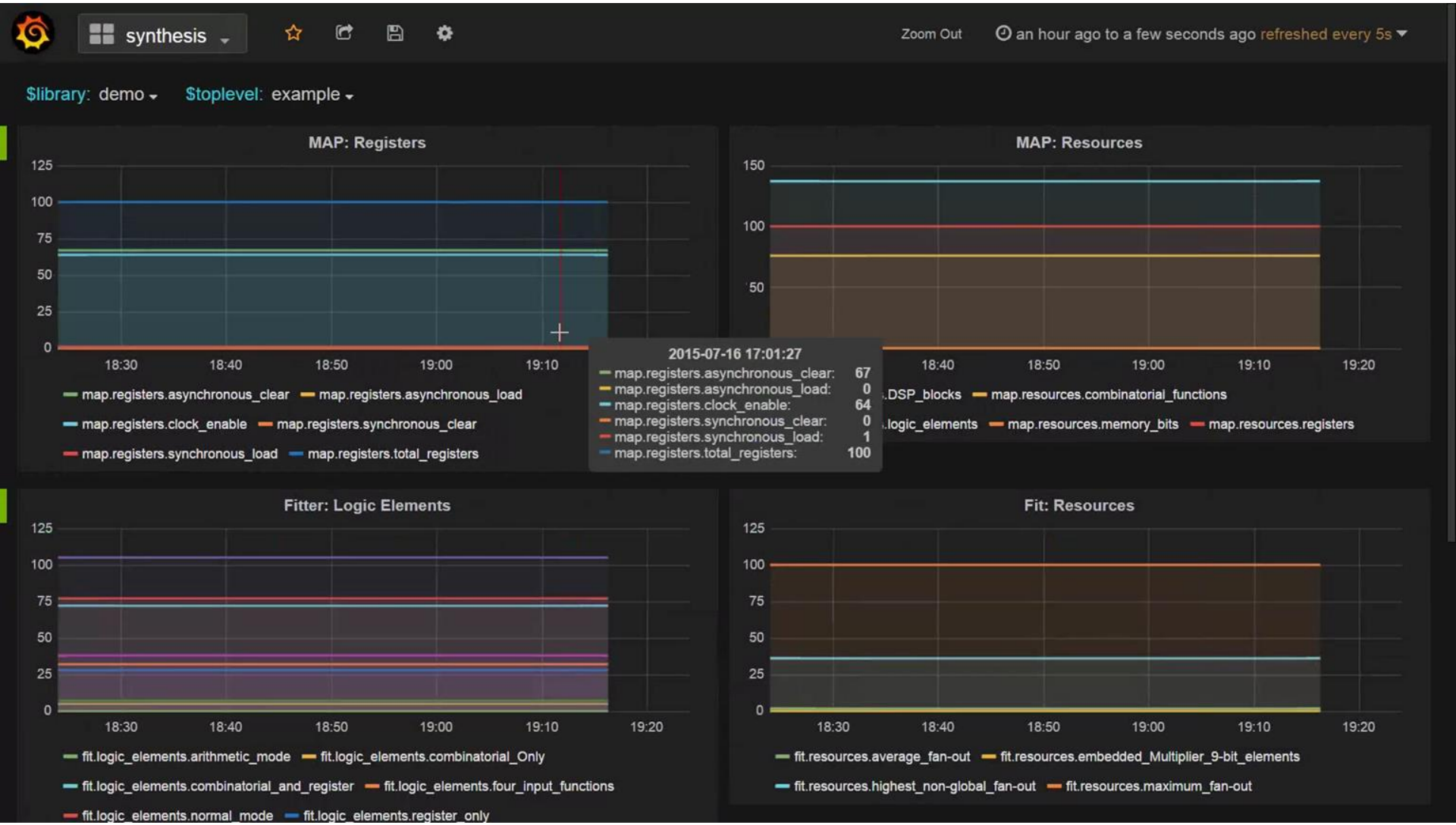
Output: Waveforms

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