

ASIC-Strength Verification in a Fast-Moving FPGA World

Bryan Murdock Fusion-io

FPGA vs. ASIC

FPGA World:

Short release cycles

Features are mixed and matched

Very small team

ASIC-strength verification in the FPGA world requires addressing these head on.

We have had great success doing so with the below tools and techniques.



Automation with Python

Challenge:

- Don't do things manually that can be scripted
- Scripts aren't your
- primary job
- Scripts need to be *easy* to read and maintain

Saved time with Python

- Very readable: like
- pseudo-code
- ■No \$%&*(#@) line-noise
- "Batteries Included." There are lots of helpful Python libraries

See How Pretty:

```
get the length of a list,
 and a string
a_list = ['a', 'b', 'c']
list_length = len(a_list)
```

a_string = 'foobarbaz' string_length = len(a_string)

Distributed Version Control (DVC)

Challenge:

Many concurrent lines of development:

- Multiple big features
- Support for old bitstream releases
- Stabilizing and testing multiple releases CVS, SVN don't handle branches and merges well

DVC Met the Challenge:

- Branching and merging are handled beautifully
- Commits are local: edit, re-arrange, or combine them before sharing
- Free, Open Source, and high quality DVC tools: Mercurial, Git







A Configuration-Driven Build System

Challenge:

With a single codebase:

- Create bitstreams for multiple FPGA's
- Create multiple bitstreams with different feature sets

Collected compile-time options and file lists into *Builds*

<i>build: functional</i> <i>lefine FUNCTIONAL_BUILD 1</i> <i>lefine CLOCK_FREQUENCEY_MHz 100</i> <i>lefine BUS_WIDTH 32</i> <i>lefine BUFFER_DEPTH 16</i> <i>hardware ecc</i>	<pre>// file list for build: functional top.v fifo.v bus.v ecc/ecc.v</pre>
<i>build: small</i> lefine SMALL_BUILD 1 lefine CLOCK_FREQUENCEY_MHz 700 lefine BUS_WIDTH 16 lefine BUFFER_DEPTH 4 lefine SOFT ECC 1	<pre>// file list for build: small top.v fifo.v bus.v software_ecc/ecc.v</pre>

Web-Based Code Reviews

Challenge:

In-person Code Reviews are:

- Inconvenient
- Hard to concentrate on the code

Review Board to the rescue

You ha	we a pending review. Edit Review Publish Discard		
19	<pre>5 count_from cf1;</pre>	15	<pre>count_from cf1;</pre>
10	6 count_from cf2;	16	<pre>count_from cf2;</pre>
		17	<pre>count_from cfs[\$];</pre>
17	<pre>7 cf1 = new(5);</pre>	18	cf1 = new(5);
18	<pre>g cf2 = new(100);</pre>	19	cf2 = new(100);
		20	cfs = '{cf1, cf2};
19	<pre>9 for(int i = 0; i < 5; i++) begin</pre>	21	<pre>for(int i = 0; i < 5; i++) begin</pre>
1 20	<pre>\$display("cf 1: %0d", cf1.body());</pre>	22	<pre>foreach(cfs[i]) begin</pre>

No written record of the review

21 \$display("cf 2: %0d	1", cf2.body() Your comment	\$display("cf %0d: %0d", 1, cfs[1].body());	
	excellent use of foreach!	nd	
22 end			
23 end	🔲 Open an <u>i</u> ssue 🤐	dit	
24 endmodule	Save Cancel Delete		
1. <u>sim/functor.sv</u> : 4 changes [<u>1 2 3 4</u>]			∇

Future Work: Registers

Challenge: Registers are tedious to maintain manually

Commercial register tools automate the process One definitive source turns into code and documentation.



ASIC-Strength Verification in a Fast-Moving FPGA World

http://www.fusionio.com

bmurdock@fusionio.com