VSDOpen Conference
2018

Front-End Symposium Intro

Steve Hoover
Redwood EDA

FRONT-END AGENDA

Steve:

- Open-Source H/W Trends
- TL-Verilog Primer

Ákos:

- Formal Verification of WARP-V, a TL-Verilog RISC-V Core Generator

Ahmed:

- Top-Down Transaction-Level Design with TL-Verilog

OPEN SOURCE H/W - What's keeping us back?

1. Access to tools
2. Access to h/w
3. Complexity/Scale

...AND THE WALLS ARE BREAKING DOWN!!!

1. Access to tools
   - Complete open-source FPGA design flows now exist
2. Access to h/w
   - Cloud FPGAs are the answer!
3. Complexity/Scale
   - TL-Verilog

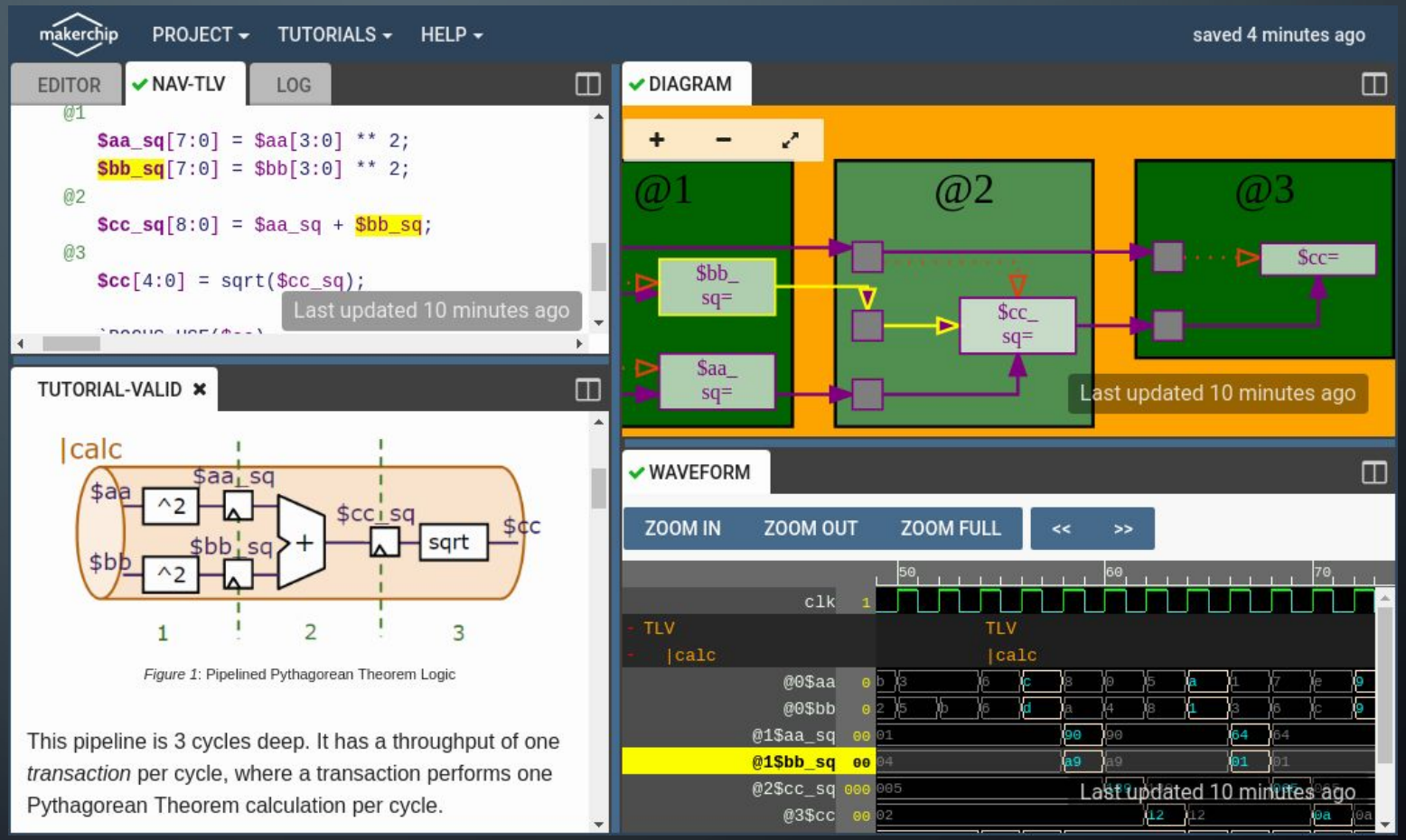
IMPACT

- Greater impact than open-source s/w
- Start-up culture will thrive!
- Just in time for the “golden age of computing”

4th Barrier - Patents

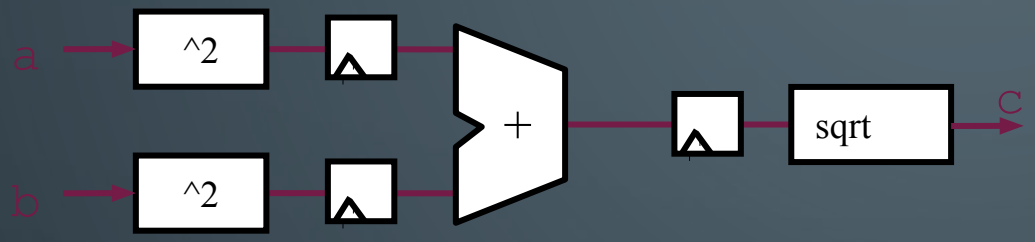
MAKERCHIP

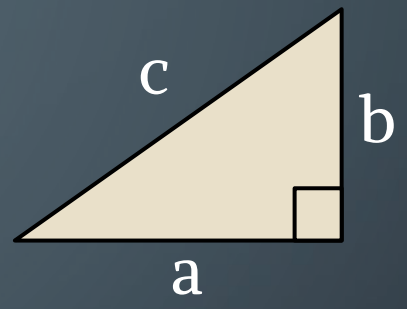
1. Access to tools
   - in browser
2. Access to h/w
   - Cloud FPGAs
3. Complex./Scale
   - TL-Verilog



TIMING ABSTRACTION

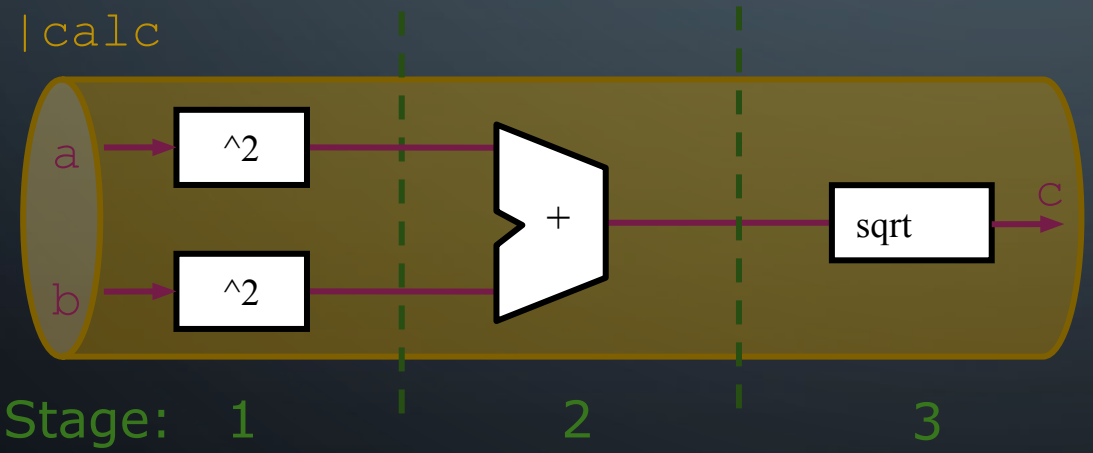
RTL:





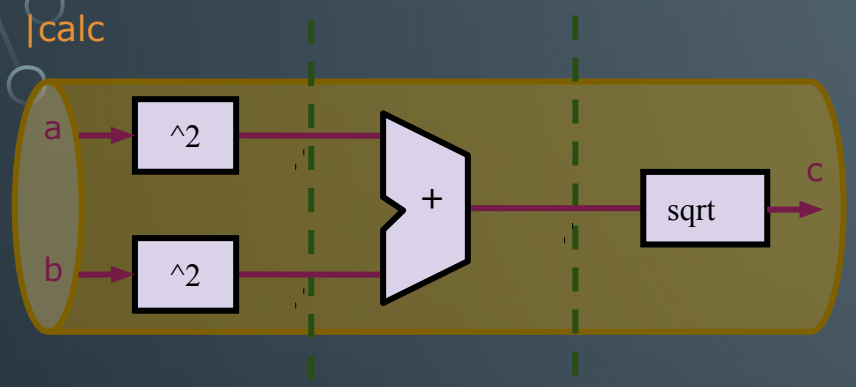
$c = sqrt(a^2 + b^2)$

Timing-abstract:



➔ Flip-flops and staged signals are implied from context.

TL-VERILOG VS. SYSTEMVERILOG



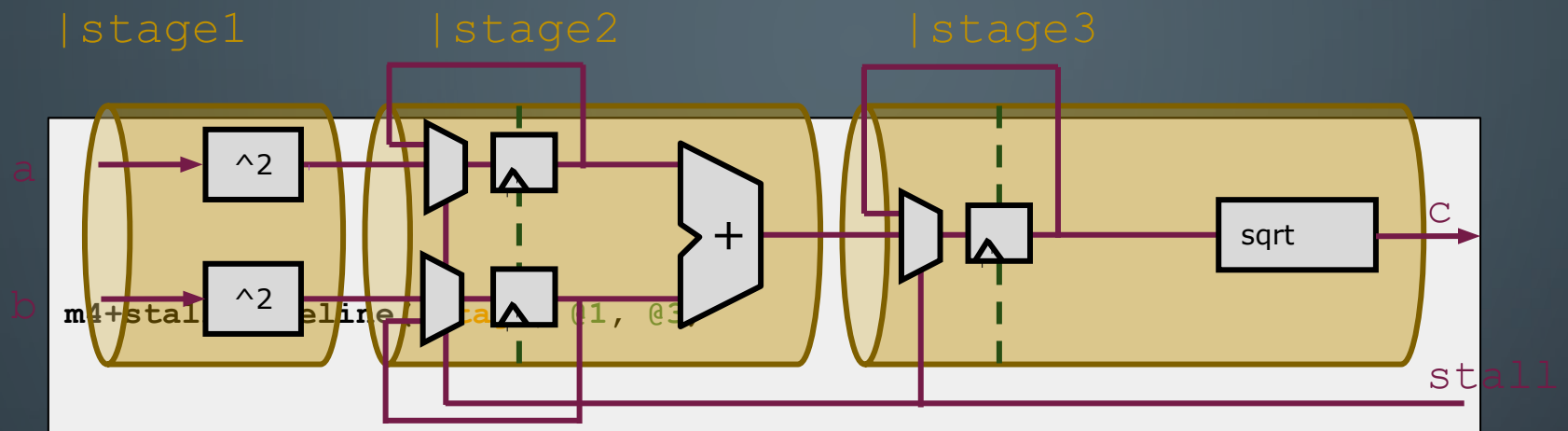
TL-Verilog

```
|calc
   @1
      $aa_sq[31:0] = $aa * $aa;
      $bb_sq[31:0] = $bb * $bb;
   @2
      $cc_sq[31:0] = $aa_sq + $bb_sq;
   @3
      $cc[31:0] = sqrt($cc_sq);
```

System Verilog

```
// Calc Pipeline
logic [31:0] a_C1;
logic [31:0] b_C1;
logic [31:0] a_sq_C1,
             a_sq_C2;
logic [31:0] b_sq_C1,
             b_sq_C2;
logic [31:0] c_sq_C2,
             c_sq_C3;
logic [31:0] c_C3;
always_ff @(posedge clk) a_sq_C2 <= a_sq_C1;
always_ff @(posedge clk) b_sq_C2 <= b_sq_C1;
always_ff @(posedge clk) c_sq_C3 <= c_sq_C2;
// Stage 1
assign a_sq_C1 = a_C1 * a_C1;
assign b_sq_C1 = b_C1 * b_C1;
// Stage 2
assign c_sq_C2 = a_sq_C2 + b_sq_C2;
// Stage 3
assign c_C3 = sqrt(c_sq_C3);
```

ADDING BACKPRESSURE




```
|stage1
   @1
      $aa_sq[31:0] = $aa * $aa;
      $bb_sq[31:0] = $bb * $bb;
|stage2



   @1
      $cc_sq[31:0] = $aa_sq + $bb_sq;
|stage3


   @1
      $cc[31:0] = sqrt($cc_sq);
```