Harnessing the Power of Word-Level Formal Equivalency Checking

Theo Drane, Principal Design Engineer
19th November 2012
Company overview

**Leading silicon, software & cloud IP supplier**
- Graphics/video, comms, processor, cloud technologies
- Licensing and royalty business model

**Licensed to many top 20 semis & OEMs**
- Servicing high volume, high growth markets

**Shipped by most major consumer brands**
- Smartphones, tablets, low-power PC’s
- TVs/STBs, games consoles, Connected/broadcast audio devices
- Automotive dashboards, navigation

**Strategic product division: Pure**
- Digital radio, internet connected audio, home automation
- IP business pathfinder, market maker/driver

**Established technology powerhouse**
- Founded 1985; London FTSE250 (IMG.L)
- Employees: 1,200+
- UK HQ; operations world-wide
- Global customer base
Delivering all the world’s standards:

**Comms:**
- TV
- Radio
- Mobile TV
- Wi-Fi
- Bluetooth

**CPU OS:**
- Linux
- Android
- MeOS

**DSP/Real Time:**
- Audio
- SW Stacks
- DSP
- Real Time
- V.VoIP
- VoLTE

**Graphics & GPU:**
- OpenGL ES
- OpenGL EP
- OpenCL
- OpenVG
- DirectX 9/10/11
- OpenRL

**Video:**
- H264
- MPEG4
- MPEG2
- VC1
- AVS
- VP6/8

**Display:**
- De-interlace
- Frame Rate Conversion
- Noise reduction
Traditional Datapath Creation

- **Specification .doc**
  - Hand Coding
  - Hand Coding
  - Hand Coding

- **Gate Level Netlist**
- **Golden RTL Model**
- **RTL Model**
- **RTL Model**

- **Constrained Random Simulation**
- **Bit Level Equivalence Checker**

- **System Model C++/SystemC**
- **System Model C++/SystemC**
1. RTL to RTL Verifications

RTL Changes –

improve Timing
improve Area
improve Power
improve Clock Gating
...
invariably late
performed at a word level

Need to check that (old) functionality is not broken.
Using Bit Level Equivalence Tools

Bit level tools are incapable of proving simple arithmetic properties, e.g.

\[ a \times a - b \times b = (a - b) \times (a + b) \]

Unsatisfactory solutions:
Reduce bit width? Exhaustive Simulation? Waterfalling?

Word Level Equivalence Tools - negligible runtime, appropriate abstraction level
2. System Model to RTL Verifications

- $ew = \text{exponent width} \ [4,11]$
- $mwa/b/y = \text{mantissa width} \ a/b/y \ [4,52]$

Exhaustive simulation for one instance - infeasible ($2^{64}$).

Note – parameter space for such designs very non-linear, no induction possible.

Word Level Formal Equivalence makes covering the entire parameter space feasible.
Formal Verification Setup

2^{128} possible input vectors

82 sec

© Imagination Technologies

IMG Presentation – Confidential
3. System Model to System Model

System Model Changes –
- improve runtimes
- improve memory usage
- refinement for High Level Synthesis?

Formal Verification of system model changes?
- rerun regression tests

Good Fit for Word Level Equivalence Tools
4. Validation – Finding Trust

Ambiguous? Specification .doc

Hand Coding +bug1

RTL Model

Formal Proof

Hand Coding +bug1

System Model C++/SystemC

When ‘Diversity’ Verification Fails

Formally Verify with all pre-existing code & third party reference

SoftFloat

RTL Model

Synopsys’ DesignWare

RTL Model
5. Validation – Finding Trust, Accuracy Freedom

Specification .doc
Accuracy Freedom

Hand Coding, Exploiting Freedom

RTL Model

Auto-generated

System Model
C++/SystemC

Prove Error is Bounded?
### Example: Single Precision Floating Point Multiplier

<table>
<thead>
<tr>
<th></th>
<th>Delay (ns)</th>
<th>Area (μm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round Towards Zero</td>
<td>1.024</td>
<td>8208</td>
</tr>
<tr>
<td>1 unit in last place (ulp)</td>
<td>0.979</td>
<td>6057</td>
</tr>
</tbody>
</table>

#### Diagram:
- **Representable Numbers**
- **Round Towards Zero (RTZ)**
- **Correct Result**
- **Acceptable 1ulp Results**

E.g., in Single Precision Floating Point Multiplier, the representable numbers and their processing characteristics are shown. The table lists the delay and area for two different rounding modes: Round Towards Zero and 1 unit in last place (ulp). The diagram illustrates the placement of these numbers on a scale, with the correct result highlighted, and acceptable 1ulp results indicating the precision level.
Phrasing the Bounded Error Property

If the design is truly 1ulp – output should be ‘1’ for all inputs.
Formal Verification Setup

2^{128} possible input vectors

Constraint $y=1'b1$

Equivalence Proofs

Runtime 6213 sec

Word Level Equivalence Checker

$FPMult$ 1 ulp

$FPMult$ RTZ

Next Float

Comparator

Comparator

OR

$a \quad b$
6. Property Checking

Other proven floating point properties of interest:

A. Commutativity: \[ ab = ba \quad a + b + c = b + c + a \]

B. Monotonicity: \[ 0 < a < b \quad \Leftrightarrow \quad \frac{1}{a} > \frac{1}{b} > 0 \]

C. Iterative Algorithm Termination: Floating Point Range Reduction
Current Challenge – Limits of Trust

An answer?

Theorem Proving e.g. HOL4/HOL Light

Very high level of investment

Productisation Level = Zero

Can be phrased, but inconclusively.

Can be phrased, but exhaustive simulation infeasible
Imagination and Formal Verification

- Formal Verification has proven its value to the core IP business of Imagination, improving quality and saving significant resources
- As Imagination continues to grow, we are always looking for great people to join us as we develop this key capability across all our core IP engineering groups

Further details:

**SNUG 2012 UK:** Property Checking of Datapath using Word-Level Formal Equivalency Tools
Theo Drane *Imagination Technologies*, Himanshu Jain *Synopsys*

**SNUG 2011 UK:** Formal Verification and Validation of High-Level Optimizations of Arithmetic Datapath Blocks
Theo Drane *Imagination Technologies*, Himanshu Jain *Synopsys*

**DAC Knowledge Center 2011:** Leap in the Formal Verification of Datapath
Theo Drane *Imagination Technologies*, George Constantinides *Imperial College London*
Harnessing the Power of Word-Level Formal Equivalency Checking

Theo Drane, Principal Design Engineer
19th November 2012