

#### The Expanding Use of Formal Techniques in Electronic Design

Raul Camposano Chief Technology Officer Synopsys, Inc.

## Why Bother with Formal Techniques?





#### Traditional simulation provides diminishing returns



## Without Formal Verification, A Good Idea...





## ....Can Disappoint





#### Another Great Design...





#### **But the Bugs Weren't Found**



#### January 6, 2000 Lockheed Martin Announces Mars Polar Lander Loss Remains a Mystery



#### **The Verification Challenge**

#### **Every 18 months:**

•Gate count doubles •Vector set grows 10x •Frequency increases 50%

#### **Benchmark Design**

- Today:
  - 0.18µ
  - >500 MHz
  - 10 million gates



Moore's Law



#### **Time to Simulate**

#### For 200M cycles, today's verification choices:

•	500M cps:	0.4 sec	Actual system HW
•	5M cps:	40 sec	Logic emulator <sup>1</sup> (QT Mercury)
•	500K cps:	7 min	Cycle-based gate accelerator <sup>1</sup> (QT CoBALT)
•	50K cps:	1 h	Hybrid emulator/simulator (Axis)
•	5K cps:	11 h	Event-driven gate accelerator (Ikos NSIM)
•	50 cps:	46 days	CPU and logic in HDL simulator (VCS)
•	5 cps:	1.3 years	Gate-level simulation (VCS)

#### How much simulation is enough?



#### New Static Verification Methodology

RTL Functional Verification
/ Regression

imput [8:0] a; output zz; wire b = a[8]; wire [2:0] c = a[8:6]; wire [5:0] d = a[8:3]; wire zz = ~ ( b | c=3'b0\_01

#### RTL Simulation Regression

 Ensures correct functionality before synthesis



 Guarantees specified properties



#### <u>Equivalence</u> <u>Checking</u>

- Usually very fast
- Exhaustive
- Focus on the design, not the vectors



#### <u>STA</u>

- Very fast, large capacity
- Exhaustive
- Signoff quality

#### SYNOPSYS\*



## • Static Timing Analysis ~\$100M, 20%

Equivalence checking

Property checking



#### What is STA?

 Determines if a circuit meets timing constraints by calculating and timing the critical paths

- Exhaustive
- Not dependent on vector stimulus



#### What is STA?

• Up to <u>100x</u> faster than dynamic simulation

- Comprehensive timing checks
  - -Set-up / hold / recovery
  - -Minimum period / clock pulse width / skew
  - -Clock gating / glitch detection
  - -Bus contention / float
  - -Unconstrained logic paths
  - -More...



## **STA in the ASIC Design Flow**





STA breaks designs into sets of signal paths
Each path has a start point and an end point

SYNOPSYS\*

#### **Signal Flow Direction**

Eliminates false paths
Used for logic functions





#### **Transistor-Level Delay Calculation**

- Group channel-connected transistors
- Consider additional elements that may affect the delay
  - Pass transistors
  - Detailed RC at output nodes



#### Slack Histogram Example



#### Timing slack histogram shows slack vs. number of endpoints synopsyst

## **Timing Waveform Example**

W <u>a</u> veform <u>V</u> iew <u>W</u> indow <u>H</u> elp	
Pin (U5/U273/B)	21.4855 slew = 0.121991
Pin (U5/U304/B)	] 22.1465 slew = 0.140608
Pin (U5/U306/B)	22.3748 slew = 0.121991
Pin (U5/U307/B)	 22.9335 slew = 0.0703218
 Pin (U5/U269/B)	23.1618 slew = 0.121991
Pin (U1/core/OPERATION[1])	
	setup required 11.9
CLOCK (U1/core/OPERATION[1])	Latenby 5.5
Ready	

SYNOPSYS'

#### **Technical Challenges**

Cannot analyze:

- Asynchronous logic
- Analog logic
- Combinational feedback loops
- Must specify detailed timing information of
  - Internally derived clocks
  - False paths (some)
  - Multi-cycle / zero-cycle paths
  - Functional / test mode info
- Process, voltage, temp variation



#### **Future Challenges for STA**

Cross-talk analysis / signal integrity

 Smaller geometries, closer nets
 Can cause signal speed-up or slowdown

 Inductance in delay calculation
 Capacity / performance improvements

 Growing design sizes: >10M gates







### **Equivalence Checking**

Equivalence checking verifies whether two designs are functionally equivalent

#### Can verify:

- RTL to RTL
- RTL to Gates
- Gates to Gates
- Does not validate RTL or Gates



#### **How Does It Work?**

- Two logic circuits broken into logic cones
- Corresponding end points (compare points) are matched between the designs
- Equivalence of combinational logic cones is mathematically proved or disproved



### How Does It Work?



## **Performance & Capacity**

Significantly Faster Verification than Gate-level Simulation:

Design Description	Formal Verif. Run Time*	Simulation Run Time
100k flat gates to hierarchical gates	4.5 minutes	hours
440k hierarchical gates to hierarchical gates	12 minutes	days
1.2M flat gates to hierarchical gates	37 minutes	days
1.2M RTL to hierarchical gates	4 hours	days

\* Synopsys Formality. Run times vary based on design style.



## **Equivalence Checking Challenges**

 Modules that are too complex – (e.g., large multipliers)
 Timing dependent functionality

- (e.g., pulse generators)
- Cannot verify added logic is actually correct – (e.g., scan chains)
- Initialization
  - (i.e., reset sequence)



#### Future of Equivalence Checking

Methodology

 Clock-gated designs
 Pipeline retiming
 Clock trees
 Black boxes
 Asynchronous bypass





Static Timing Analysis

Equivalence Checking

Property Checking





## **Property / Model Checking**

- Verifies that a design does/doesn't conform to a specified property under all possible sets of legal input conditions
- Ideal for verifying a high-level spec with an RTL implementation

 Example: Verify that a bus access will never be granted to two clients at the same time (safety)



#### Formal Property Verification Example



#### **How Model Checking Works**

Model checking algorithm
 Fix point computation
 Can the design over go had

– Can the design ever go bad?





#### **How Model Checking Works**

Model checking algorithm
 Fix point computation

– Can the design ever go bad?





#### **How Model Checking Works**

Model checking algorithm
 Fix point is reached

– The design can never go bad!



SYNOPSYS"



### **Model Checking**

Pro – Exhaustive

- Con
  - -Expensive
    - Symbolic computation and representation of all states that can potentially go bad
    - Naïve BDD-based symbolic model checkers can handle around 150 latches
    - 2<sup>150</sup> states! (10<sup>78</sup> particles in the universe)
    - Not big enough for industrial designs



## **Capacity Problem**

Model-checking capacity depends on the techniques used

- Multiple Analysis Engines
  - SAT
  - BDD
  - ATPG
- Abstraction
- Combining Formal techniques with Simulation



## **Property Verification Results**

	Proven	Violated	Unknown	total
User Defined	2	2	2	6
\$finish	7	0	0	7
// Illegal state	6	8	4	18
full & parallel_case	81	8	2	91
total	96	18	8	122

Replay these traces to find bugs !!!



#### **Stimulus Generation: Composite Results**

Goalset		Regression	States	States	States	Search
Name	Nets	Coverage	Reached	Unreachable	Unknown	Runtime
goal1	4	13	13	3	0	20m
goal2	4	10	10	6	0	3.5H
goal3	13	269	132	7152	908	59H
goal4	16	246	18780	0	46756	70H
goal5	17	579	6294	94208	30570	10 H
goal6	14	531	213	13056	3115	40H
goal7	17	6555	6193	0	59343	25H
goal8	24	2727	18588	13725632	3032996	24H
goal9	12	247	259	3632	205	22H



#### Formal Reachability Analysis Assists Random Simulation





#### Summary

- Formal techniques can eliminate many bugs that traditional simulation doesn't find
- Static timing analysis and equivalence checking are mature formal technologies
- Property checking is an emerging formal verification technology
- Adoption is accelerating





## **Prevent Disastrous Design Problems**





# **If They Had Only Used Formal Techniques...**



SYNOPSYS'

## **SYNOPSYS**<sup>®</sup>

#### **Your Design Partner**