ENCS5337: Chip Design Verification

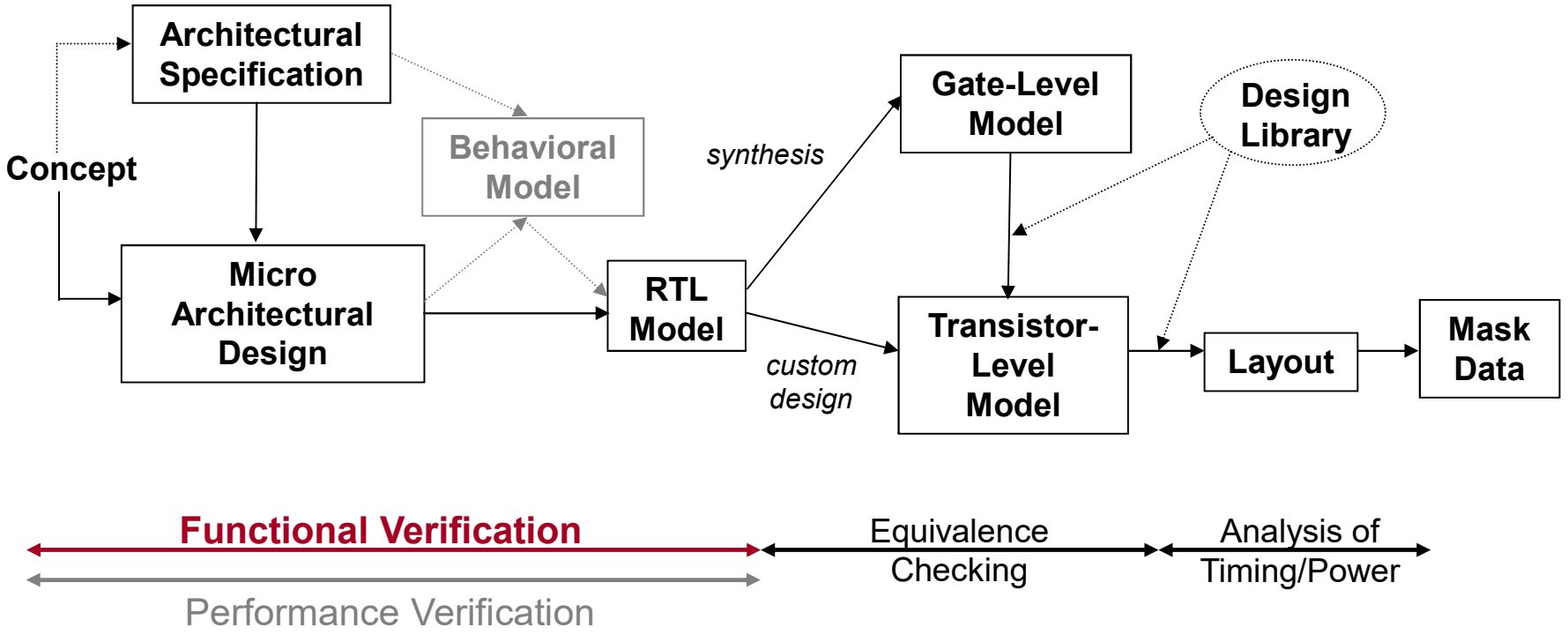
Spring 2023/2024

Introduction and Motivation

Dr. Ayman Hroub

Many thanks to Dr. Kerstin Eder for most of the slides

The IC Design Process



Functional verification aims to demonstrate that the functional intent of a design is preserved in its implementation.

What is Design Verification?

"Design Verification is the process used to gain confidence in the correctness of a design w.r.t. the requirements and specification."

- Confirms that a system has a given input / output behaviour, sometimes called the transfer function of a system.

Why is Verification important?

- Verification is the single biggest lever to effect the triple constraints:

Quality

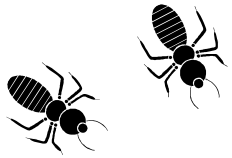
- A high quality track record preserves revenue and reputation.
- Ideally a team can establish a “right-first-time” track record.

Cost

- Fewer revs through the development/fabrication process means lower costs.
- Respinning a chip costs hundreds of thousands of £/$/€
  + the associated lost opportunity costs.

Timing/Schedule

- Fewer revs through the development/fabrication process means faster time-to-market.
- Respinning a chip costs 6-8 weeks at least
  + the associated “lost opportunity” costs.







All about Bugs

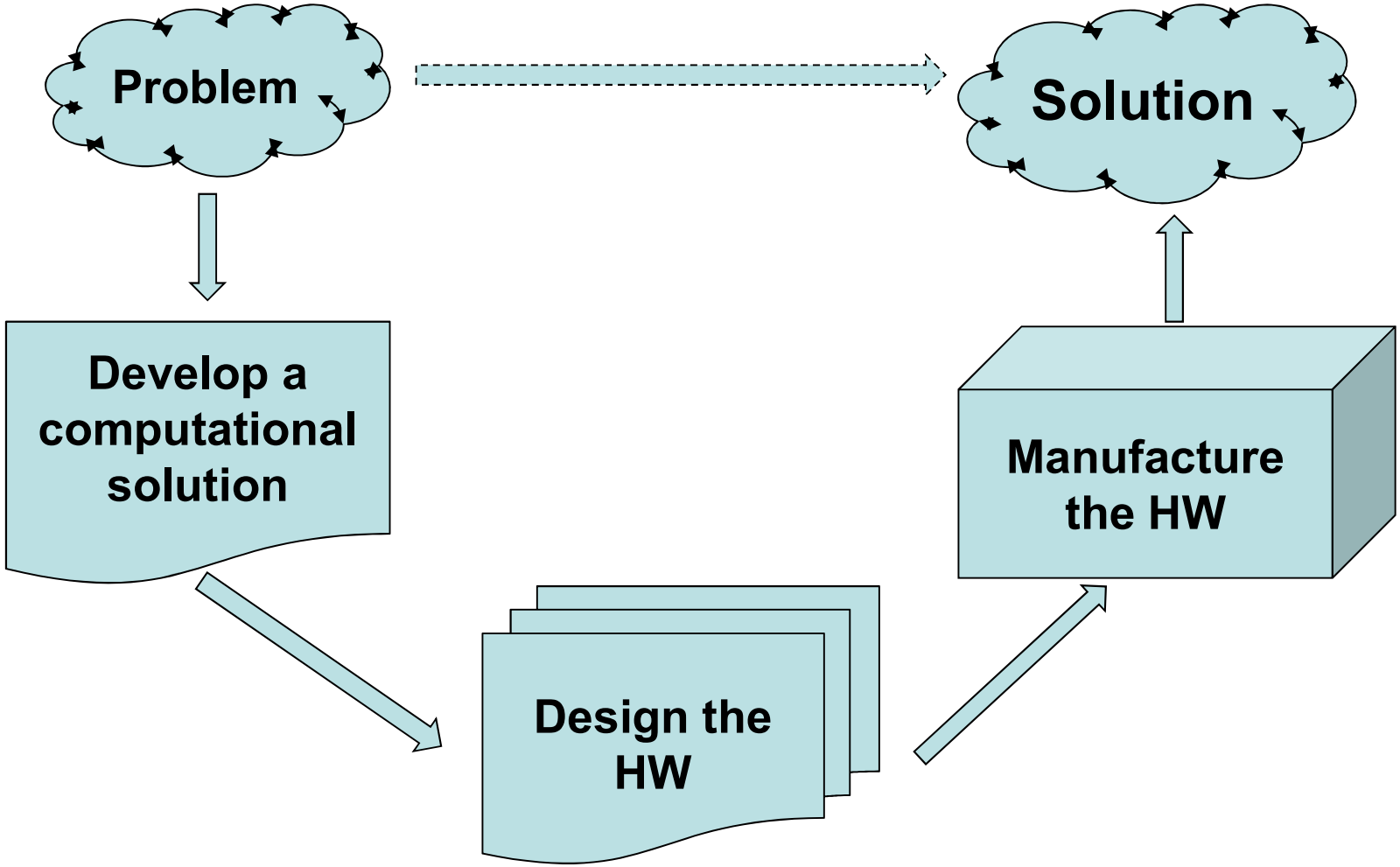
Types of bugs

How are bugs introduced?

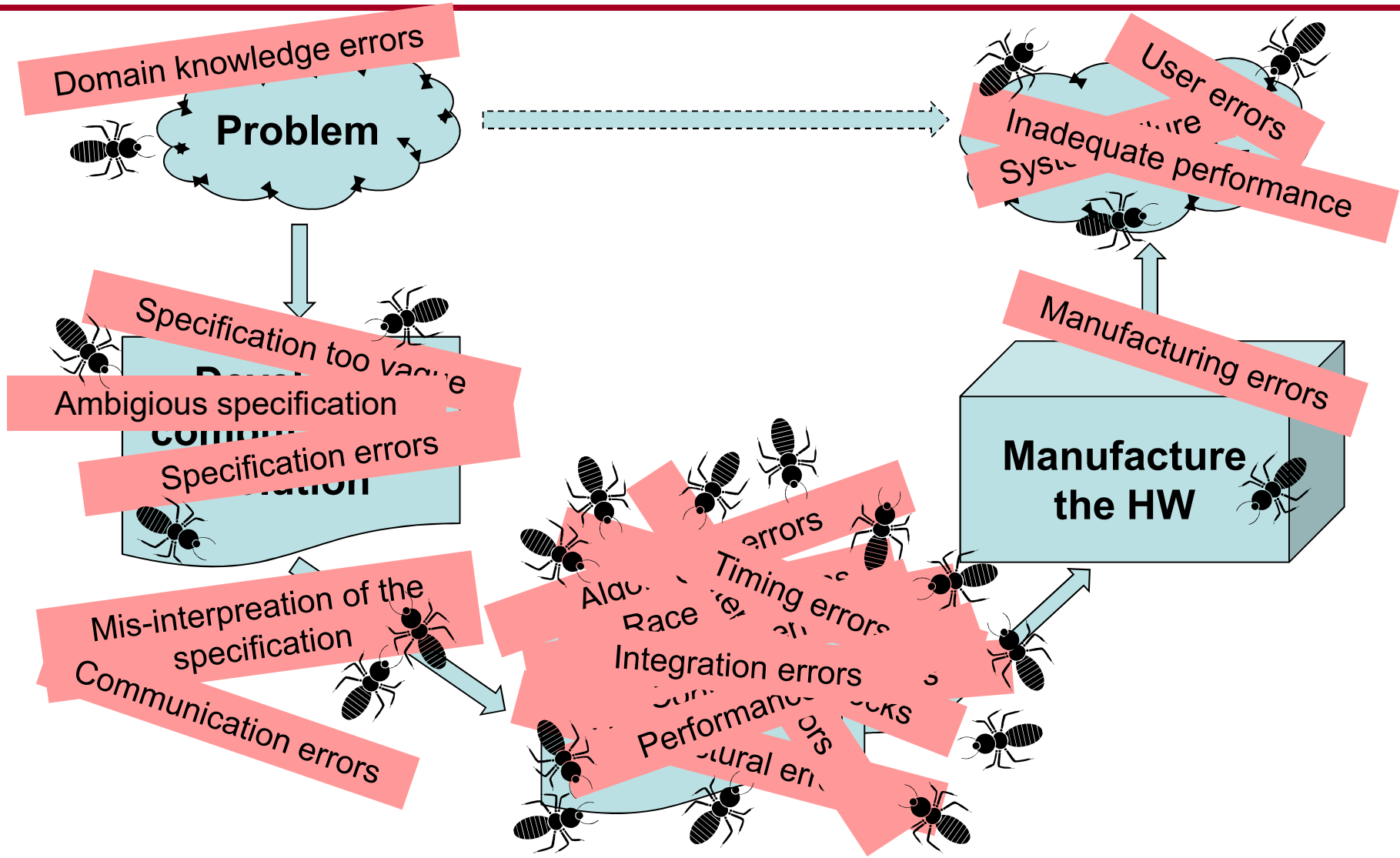
How can bugs be found?

Why do Designs have Bugs?

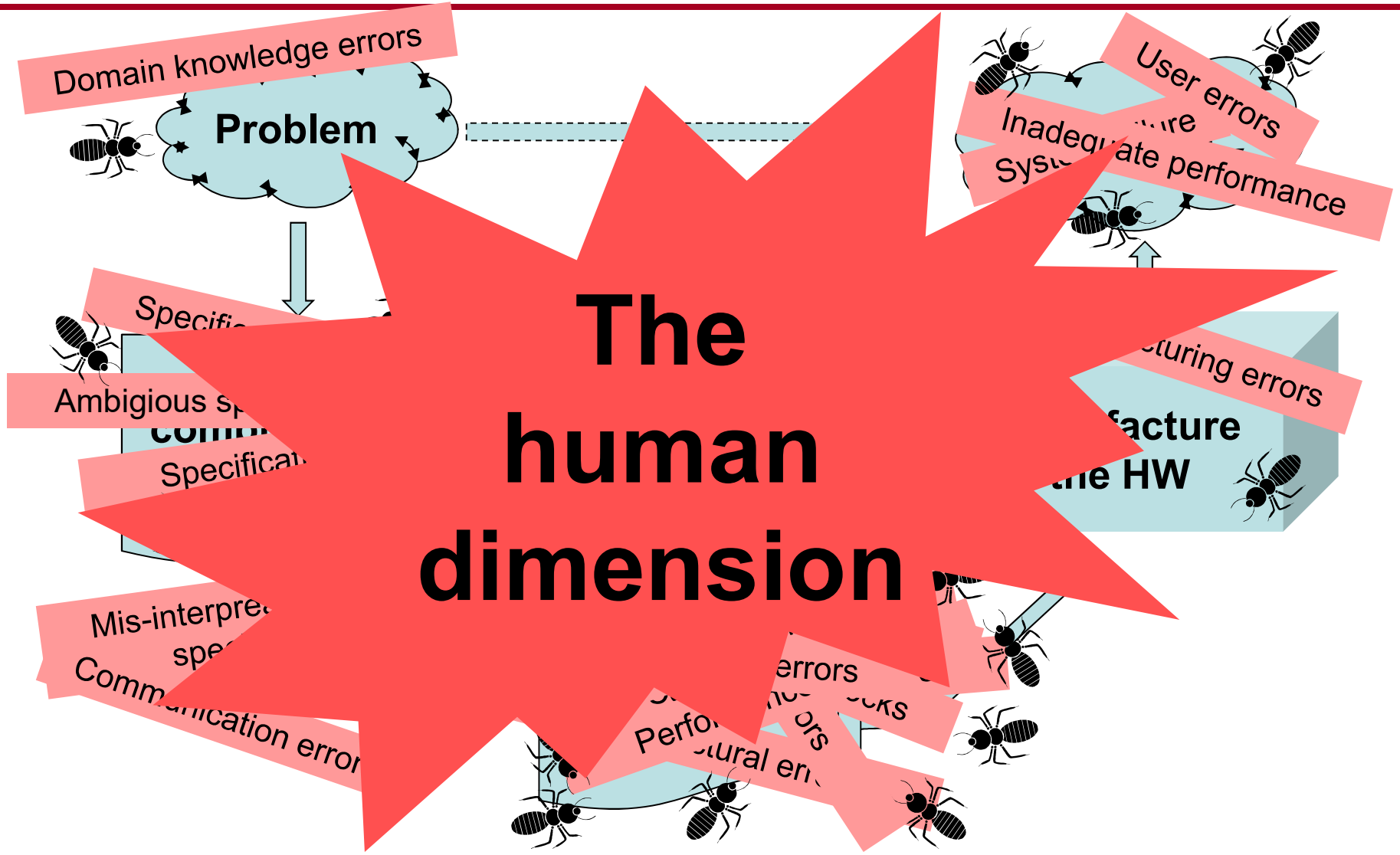
Why do Designs have Bugs?



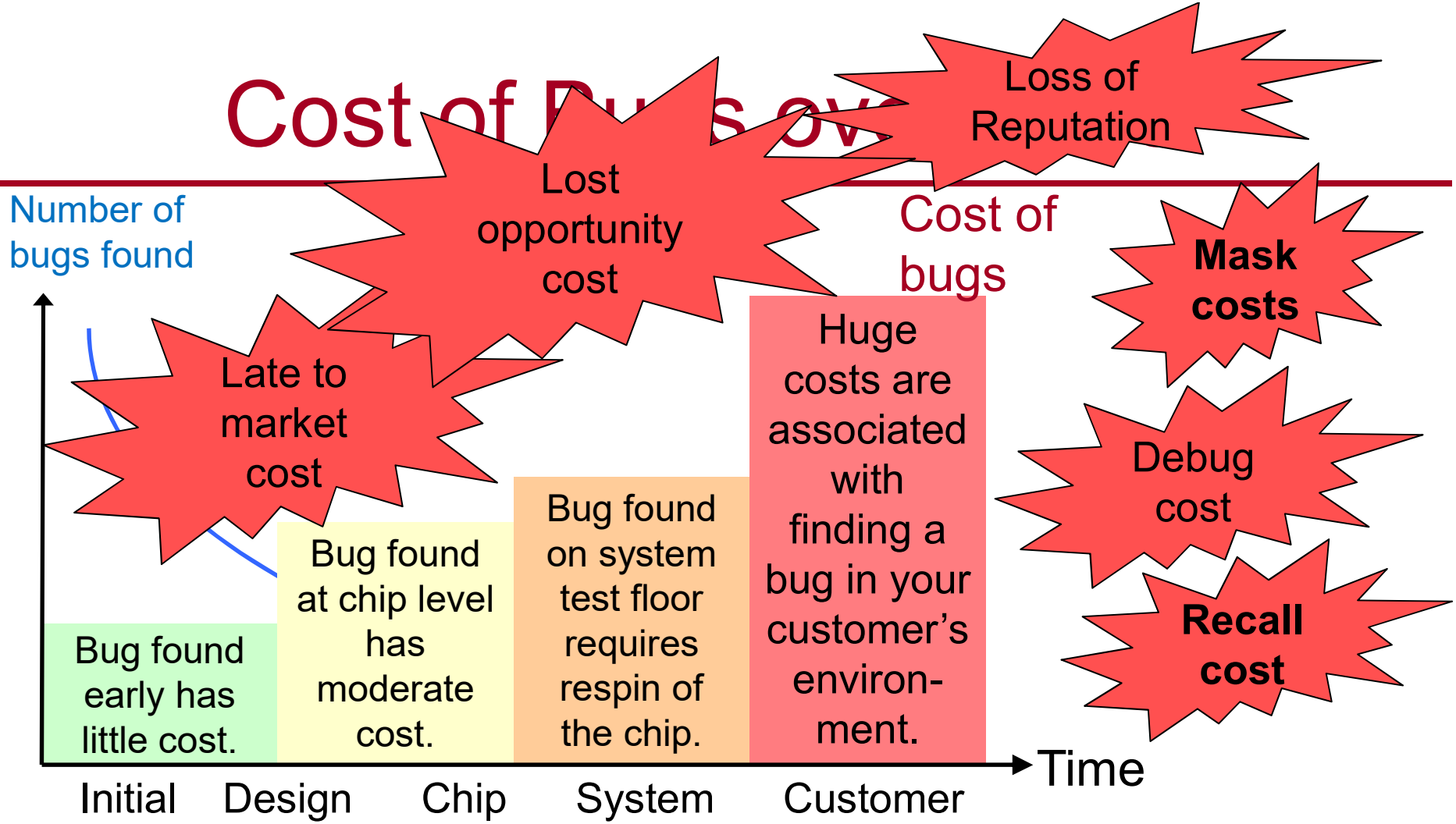
Why do Designs have Bugs?



Why do Designs have Bugs?



Cost of Bugs over



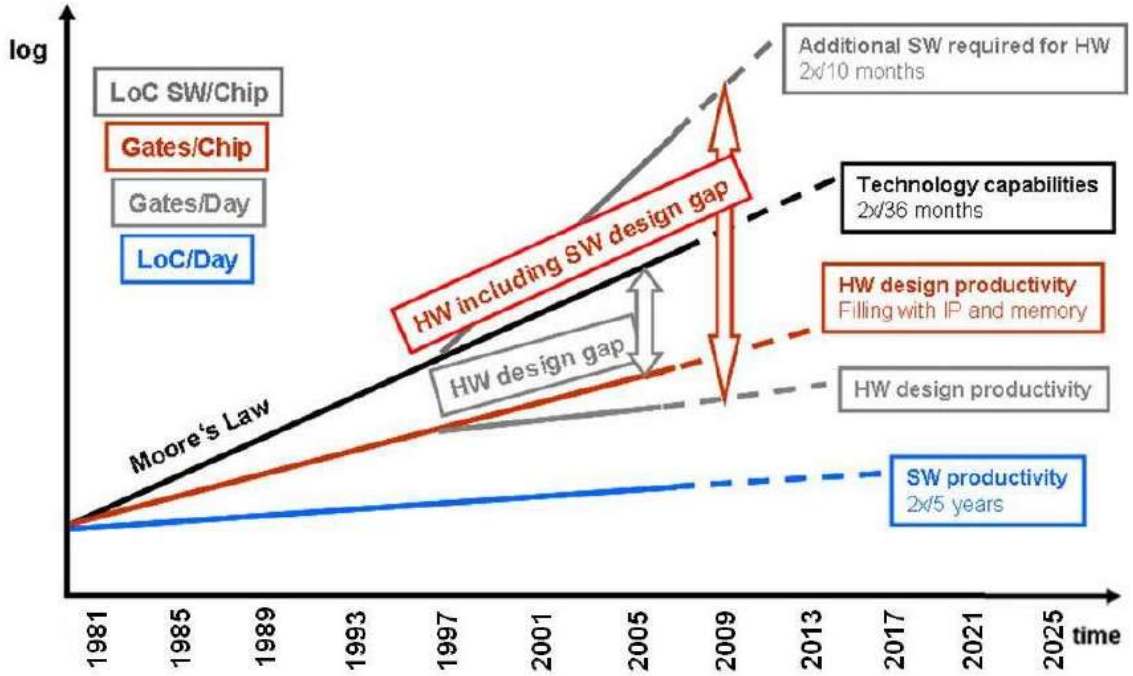
The longer a bug goes undetected, the more expensive it is!

Remember the Intel Pentium FDIV bug!
http://en.wikipedia.org/wiki/Pentium_FDIV_bug

Increasing Design Complexity vs tight TTM Constraints

ITRS Edition 2009, Design Chapter (http://www.itrs.net/ and http://www.itrs2.net/)

– Hardware and Software Design Gaps versus Time

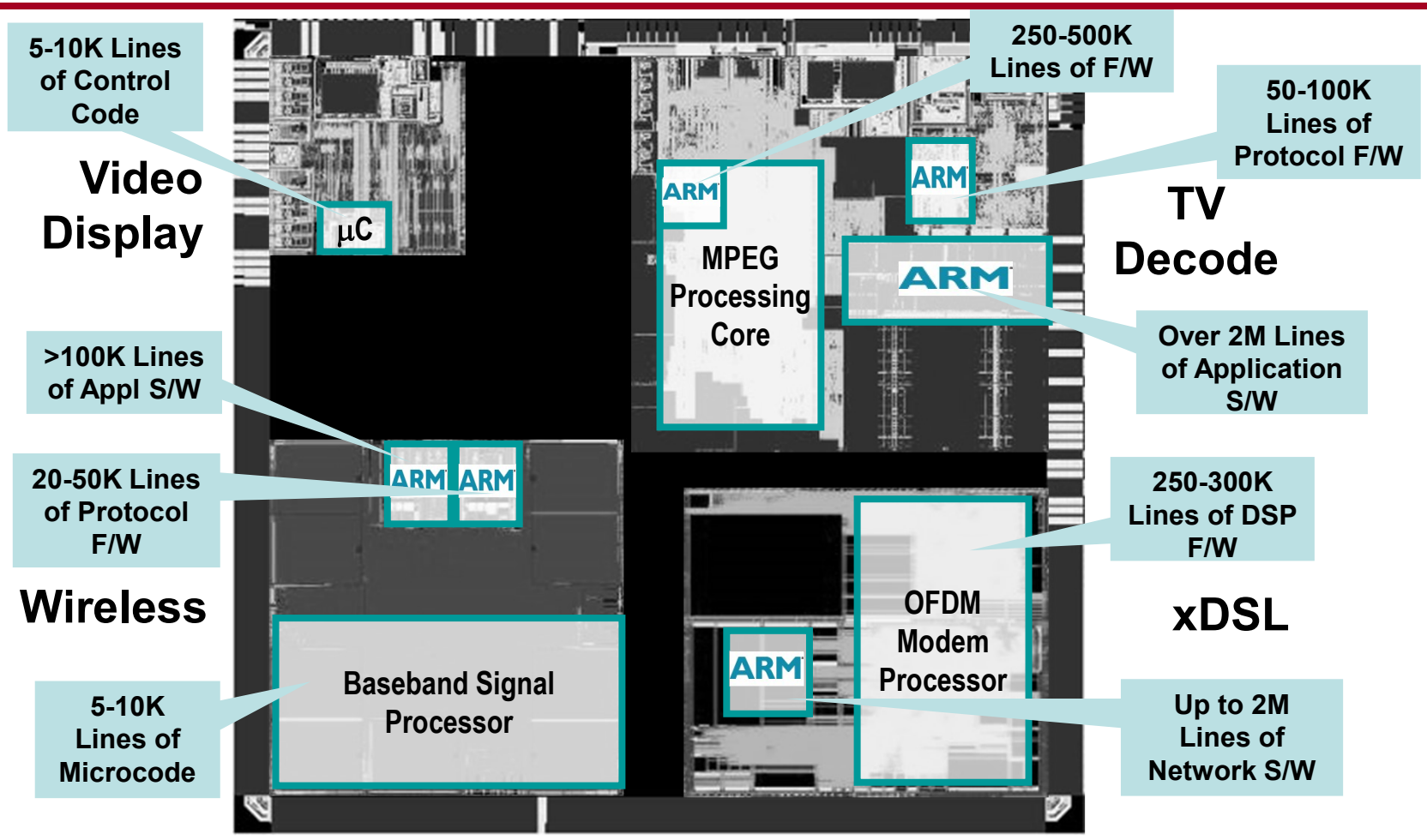


Getting it right (first time) is more and more difficult:

– rapidly increasing design complexity

– tight "time-to-market" constraints

Increasing Design Complexity



Multiple Power Domains, Security, Virtualisation
Nearly five million lines of code to enable Media gateway

CONEXANT

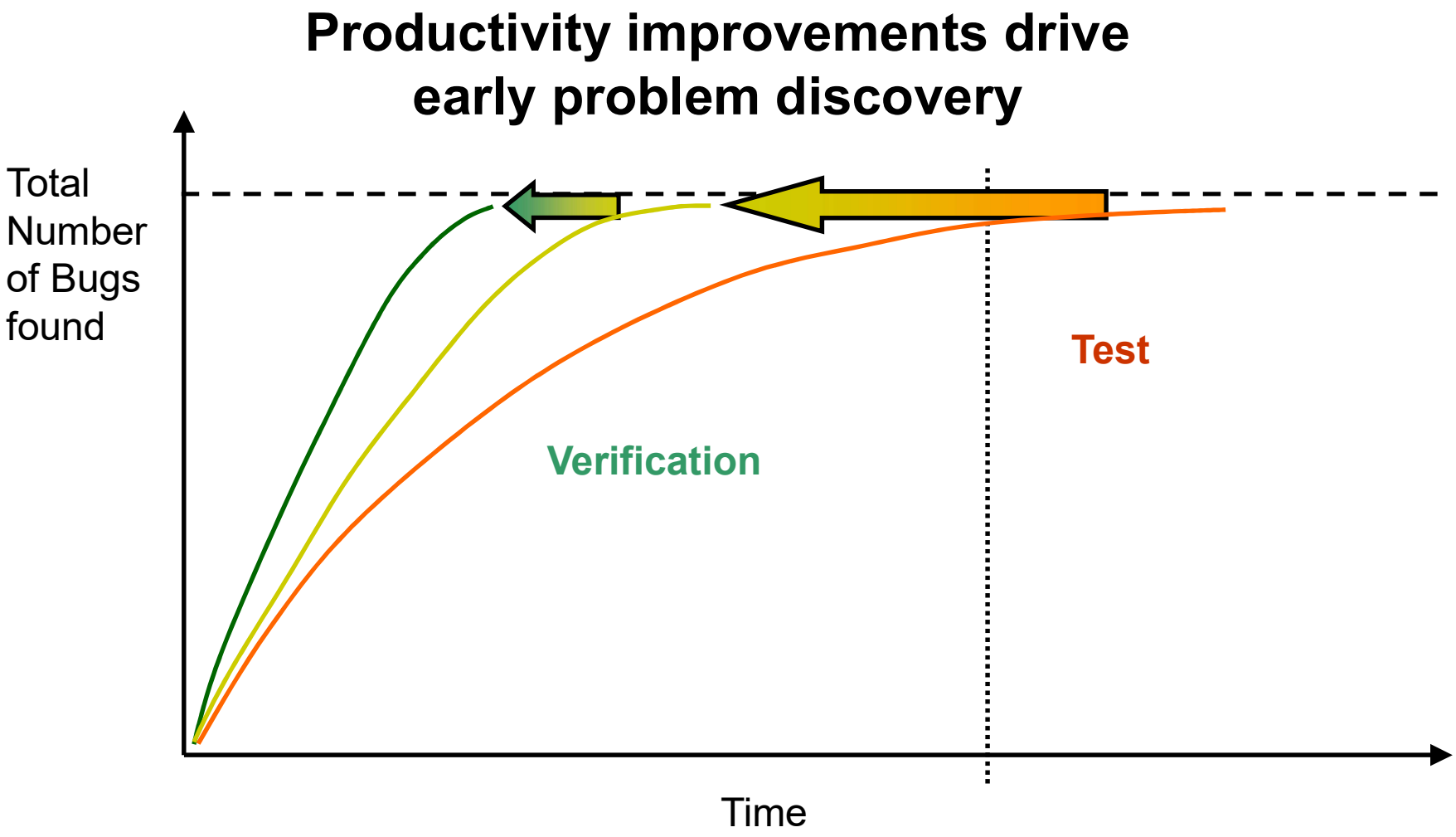
Increasing Verification Productivity

Need to minimise verification time e.g. by using:

- **Parallelism:** Add more resources
- **Abstraction:**
  - Higher level of abstraction (i.e. C vs Assembly)
  - This often means a reduction of control!
- **Automation:**
  - Tools to automate standard processes
  - Requires standard processes/methodology.
  - Usually a variety of functions, interfaces, protocols, and transformations must be verified.
  - Not all (verification) processes can be automated.

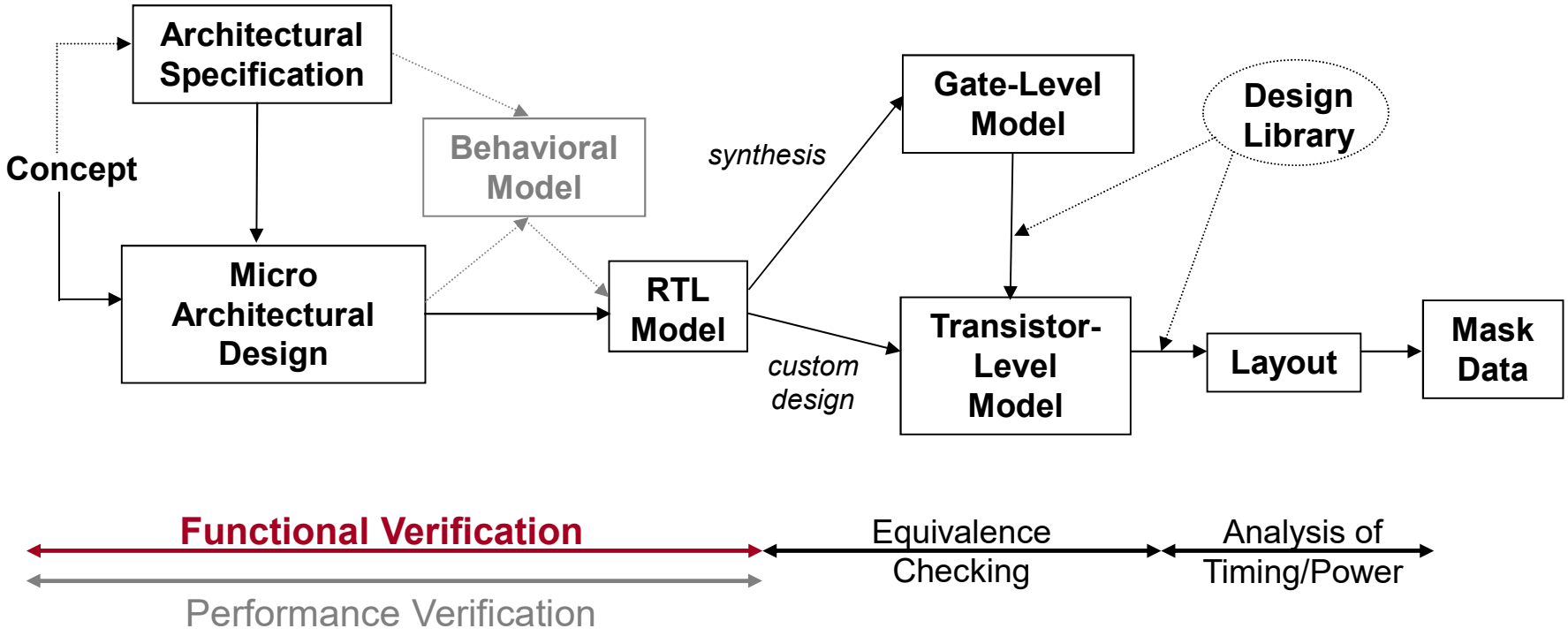
Productivity improvements drive early problem discovery!

Increasing Verification Productivity



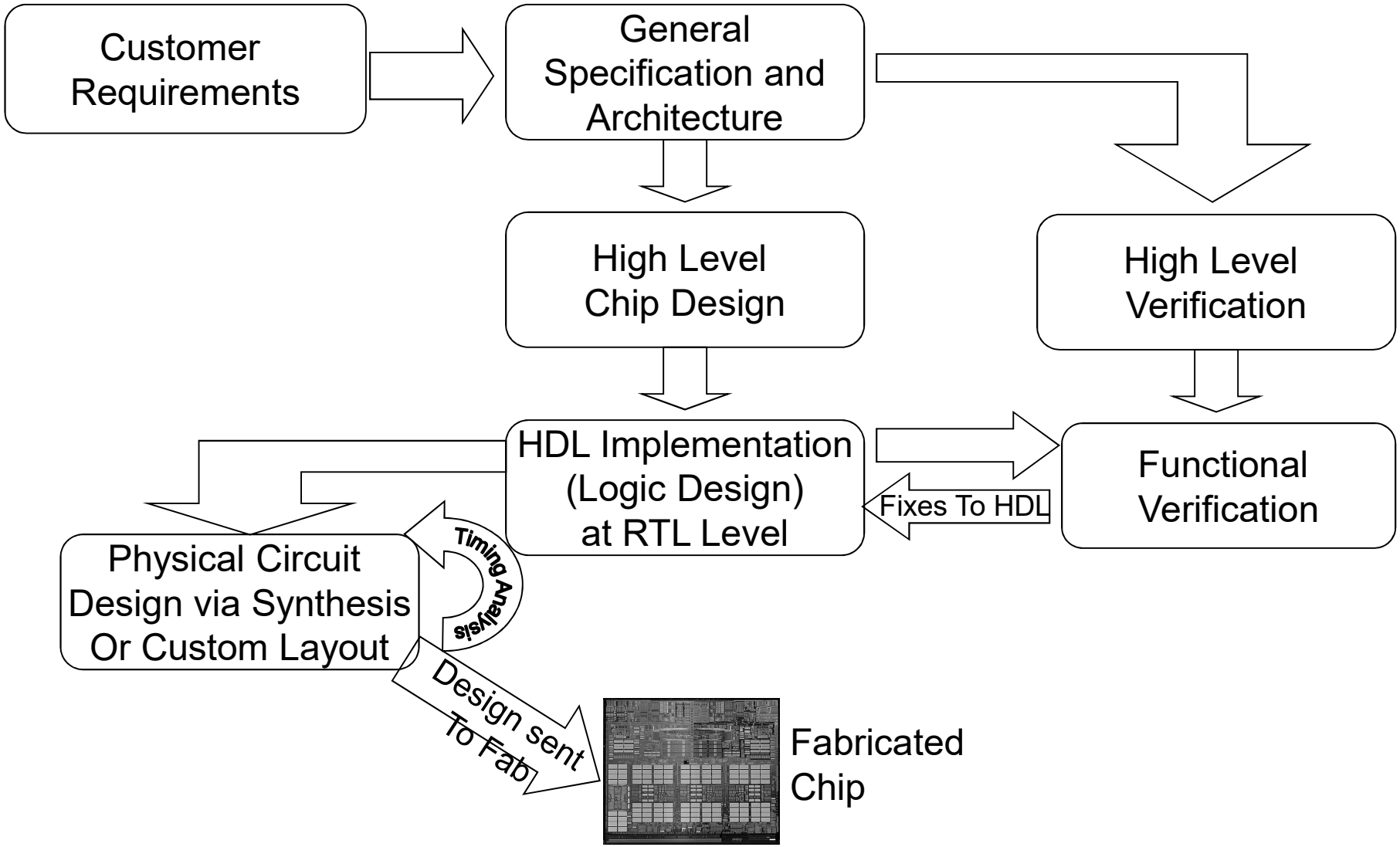
Verification in the IC Design Process

Recall: The IC Design Process



Functional verification aims to demonstrate that the functional intent of a design is preserved in its implementation.

Chip Design Process

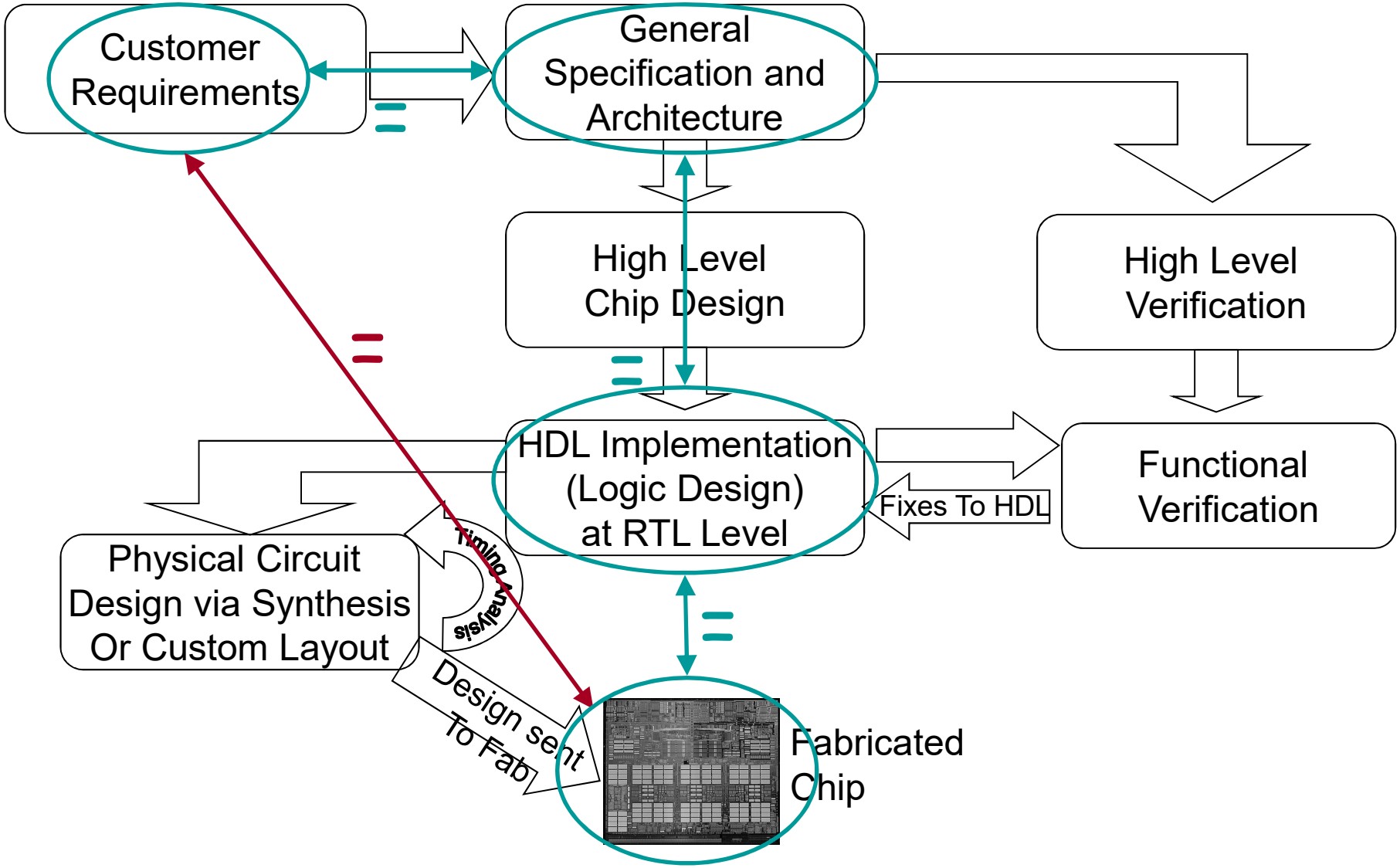


Role of Verification in IC Design

IC design process is complex:

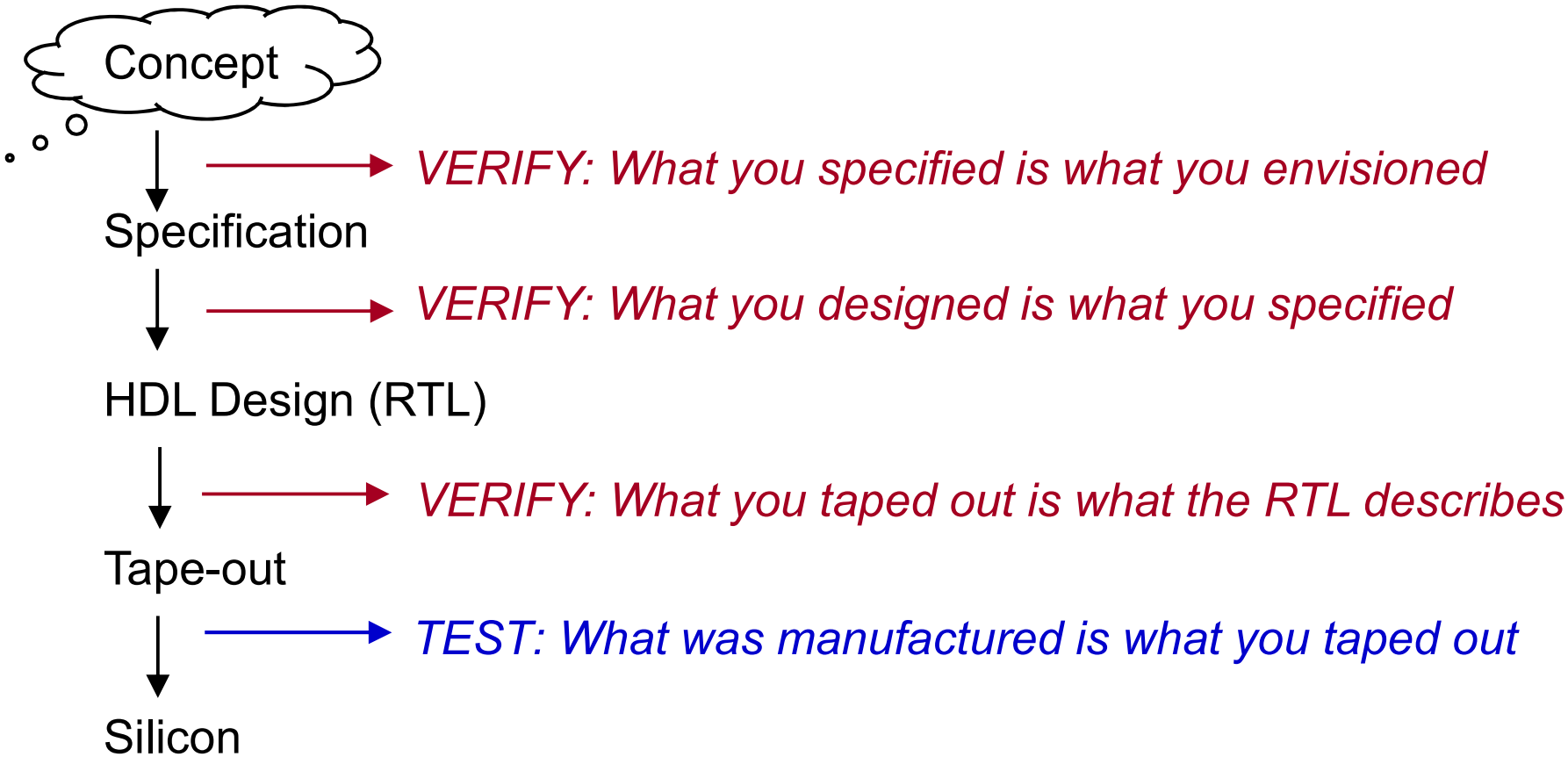
- **Engineers need to balance conflict of interest:**
  - Tight time-to-market constraints vs. increasing design complexity
- **Aim:** "Right-first-time" design, "correct-by-construction"
- More and more time-consuming to obtain acceptable level of confidence in correctness of design!
- **design time << verification time**
  - Remember: Verification does not create value!
    - But it preserves revenue and reputation!
  - Up to 70% of design effort can go into verification.
  - 80% of all written code is often in the verification environment.
  - Properly staffed design teams have dedicated verification engineers.
  - In some cases verification engineers outnumber designers 2:1.

Chip Design Process



What are you going to verify?

How do Designers know whether a circuit is correct?



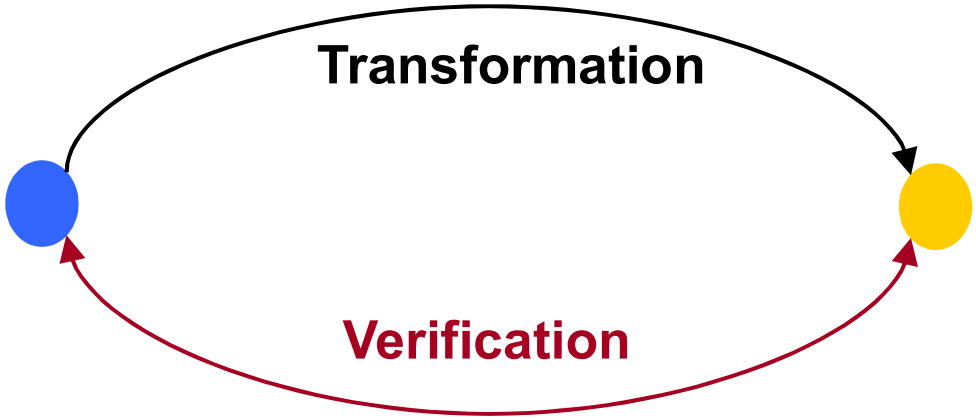
There is skill, science and methodology behind verification.

Reconvergence Models [Bergeron]

Conceptual representation of the verification process

- Most important question:

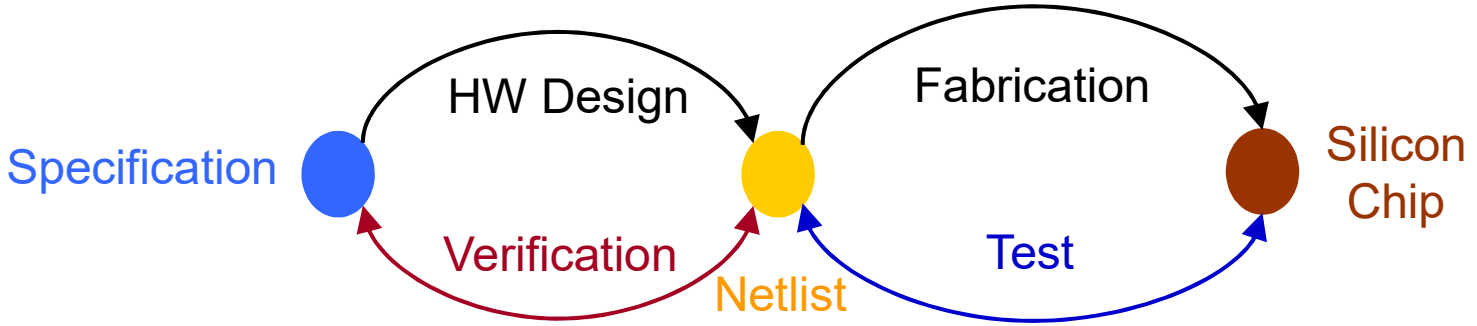
What are you verifying?



- Purpose of verification is to ensure that the result of some transformation is as intended or as expected.

Verification vs. Test

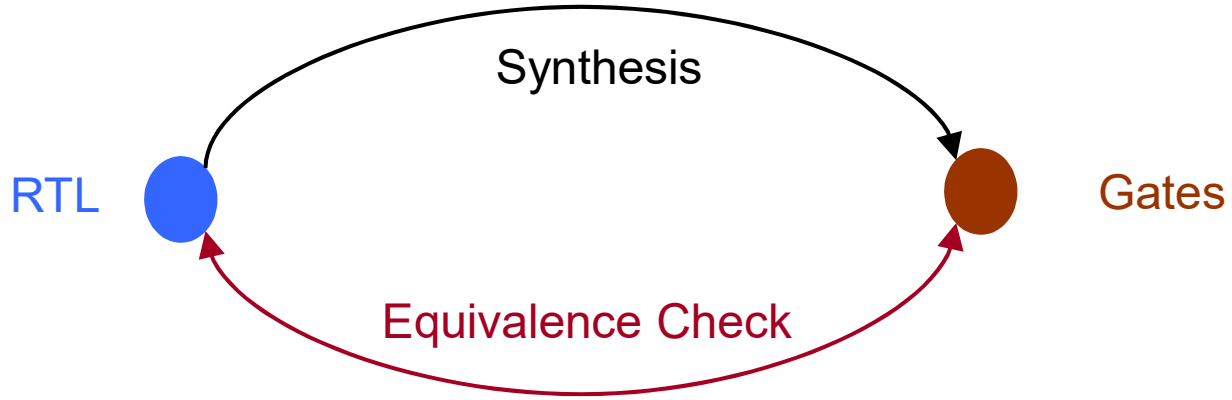
- **Often confused in the context of HW design!**
  - Purpose of test is to show design was **manufactured** properly.
  - Verification is done to ensure that **design meets its functional intent prior to manufacture!**



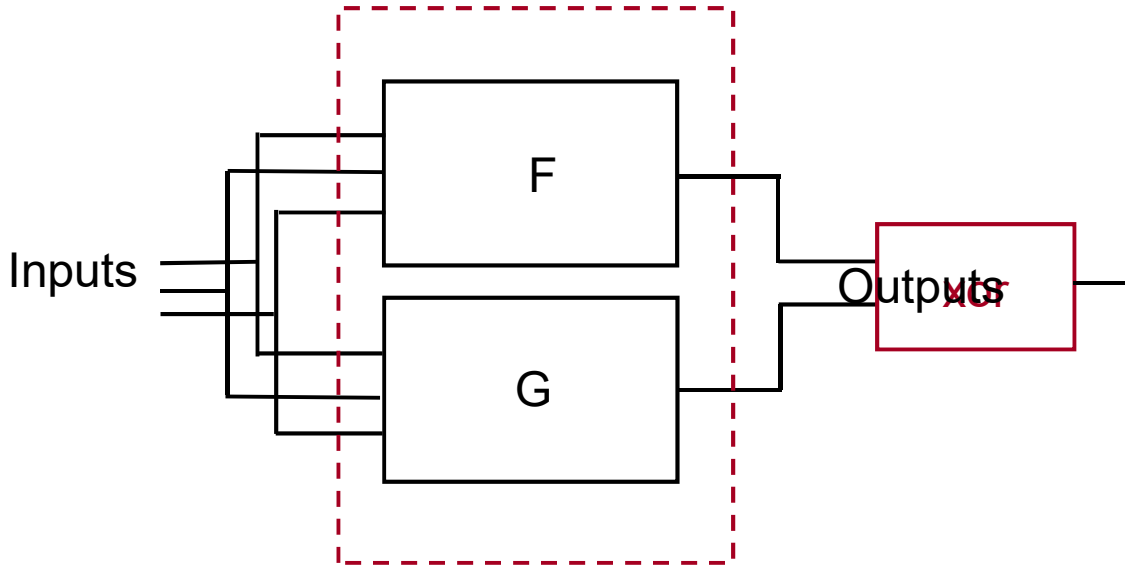
Formal: Equivalence Checking

Compares two models to check for equivalence.

- Proves mathematically that both are *logically equivalent*.
  - Commonly used on **lower levels** of design process.
- Example: RTL to Gates (Post Synthesis)



Equivalence Checking



- Conceptually, we are asking the question:

  "Is there an input vector such that the output of the XOR gate can be "1"?

Cost of Verification

Necessary Evil

- Always takes too long and costs too much.
- As number of bugs found decreases, cost and time of finding remaining ones increases.

So when is verification done?

(Will investigate this later!)

  - Remember: Verification does not generate revenue!

Yet indispensable

- To create revenue, design must be functionally correct and provide benefits to customer.
- Proper functional verification demonstrates **trustworthiness** of the design.
- Right-first-time designs demonstrate **professionalism** and ”increase” reputation of design team.

Verification is similar to statistical hypothesis testing

Hypothesis ”under test” is: **The design is functionally correct.**

	Good Design (no bugs in design)	Bad Design (buggy design)
Bugs found		
No Bugs found		

Verification is similar to statistical hypothesis testing

Hypothesis ”under test” is: **The design is functionally correct.**

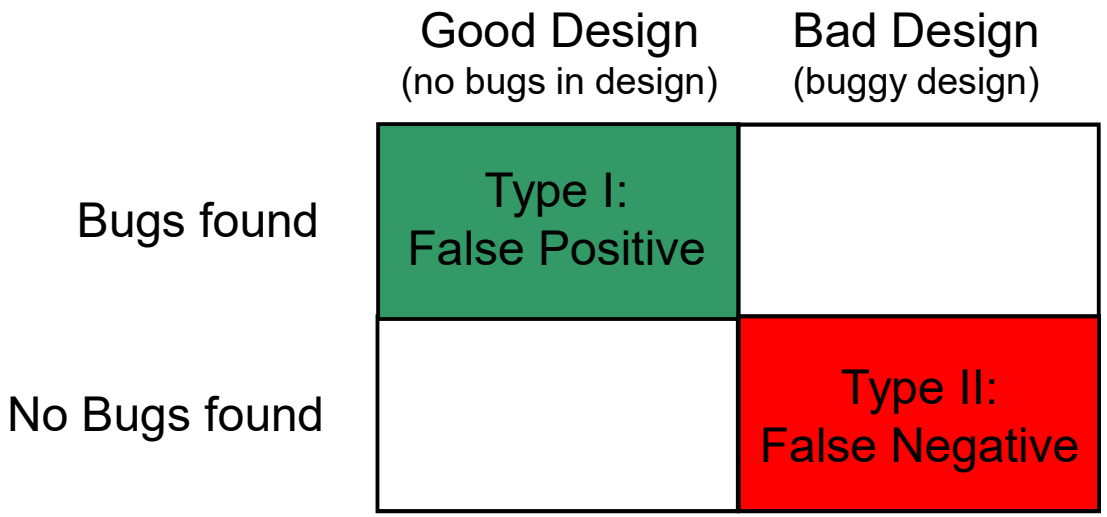
	Good Design (no bugs in design)	Bad Design (buggy design)
Bugs found	Type I: False Positive	
No Bugs found		

Type I mistakes (“convicting the innocent”, a “false alarm”):

- Easy to identify - found error where none exists.

Verification is similar to statistical hypothesis testing

Hypothesis ”under test” is: **The design is functionally correct.**



Type I mistakes (“convicting the innocent”, a “false alarm”):

- Easy to identify - found error where none exists.

Type II mistakes (“letting the criminal walk free”, a “miss”):

- Most serious - verification failed to identify an error!
- Can result in a bad design being shipped unknowingly!

Summary

- **What is Design Verification?**
  - Why do we care?
  - Verification vs validation
- **Bugs**
  - Sources of bugs
  - Cost of bugs
  - Importance of Design Verification
- **Impact of increasing design complexity**
  - Shrinking time to market windows
  - Increasing Productivity
- **The chip design process**
  - Where does Verification “fit”?
- **Reconvergence Models**
  - Help us identify what is being verified