Defect Avoidance in Programmable Devices

Steve Trimberger
Xilinx Research
March 2014

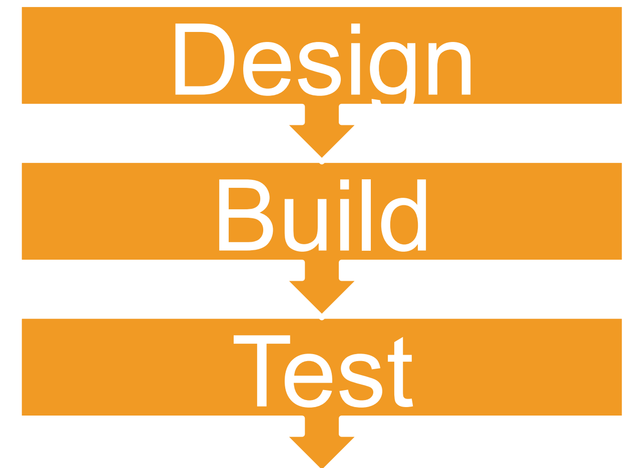
Disclaimer

The topics covered in this presentation are the subject of investigation in Xilinx Research. They are not necessarily being deployed in any Xilinx product present or future.

XILINX ALL PROGRAMMABLE™

What Do We Really Do?

- Qualify process
- Guard-band design for full lifetime and environment
- Design and manufacture
- Test to eliminate manufacturing defects and out-of-spec

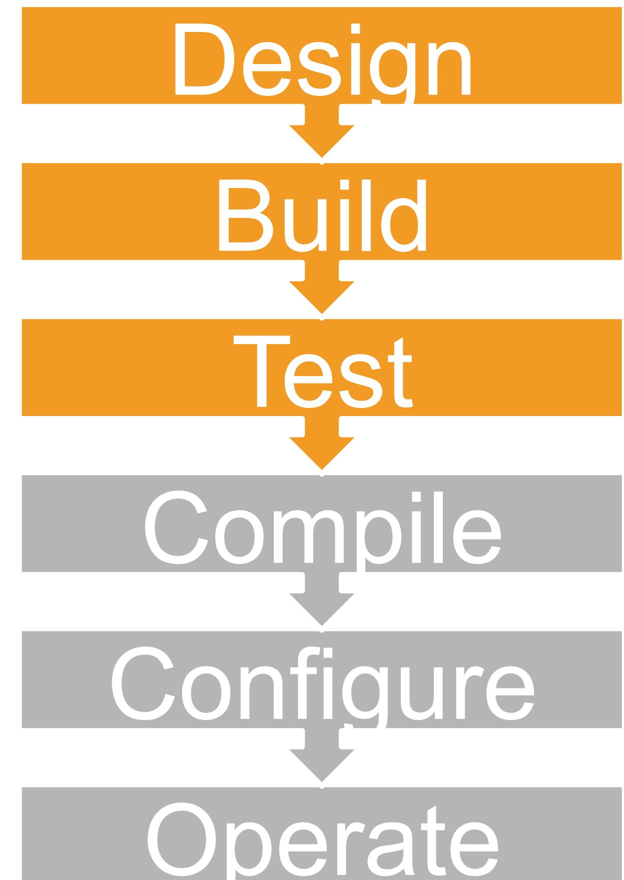


- Compile design to target
- Load design
- Operate over a lifetime

What Do We Really Do?

- **Qualify process**
- **Guard-band design for full lifetime and environment**
- **Design and manufacture** U
- **Test to eliminate manufacturing defects and out-of-spec** U

- **Compile design to target** U
- **Load design**
- **Operate over a lifetime** U



Why Do We Do It?

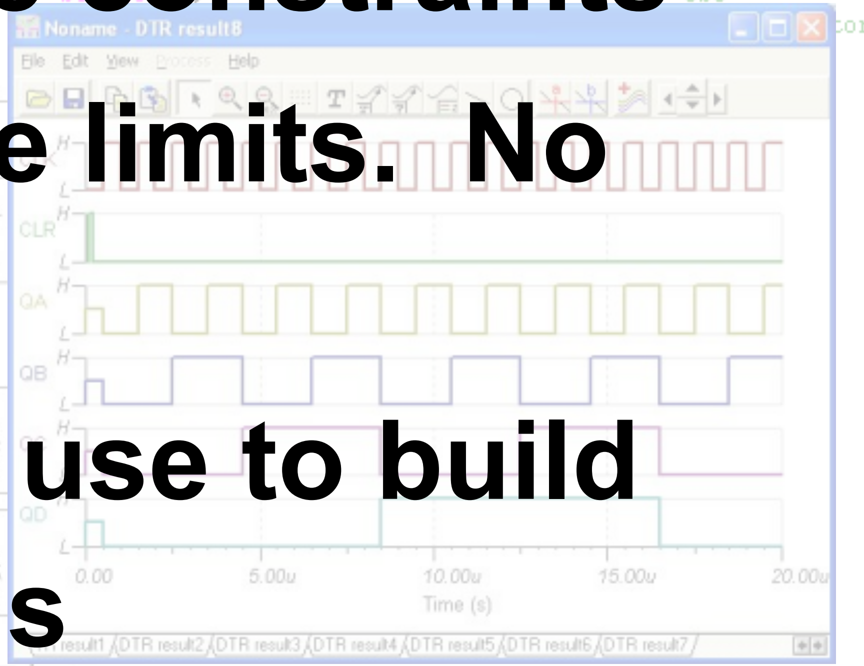
- You don’t get paid for what you do.
- You only get paid for what you guarantee.

Trimberger’s Business Rule

XILINX ALL PROGRAMMABLE.

What is the Problem I'm Facing?

- It is really logic: digital and analog
- It has hard real-time constraints
- And I am near those limits. No spare ns
- This is the stuff we use to build higher-level systems



Question

Why do we throw away bad chips?

Ummm … because they don't work?

Yield

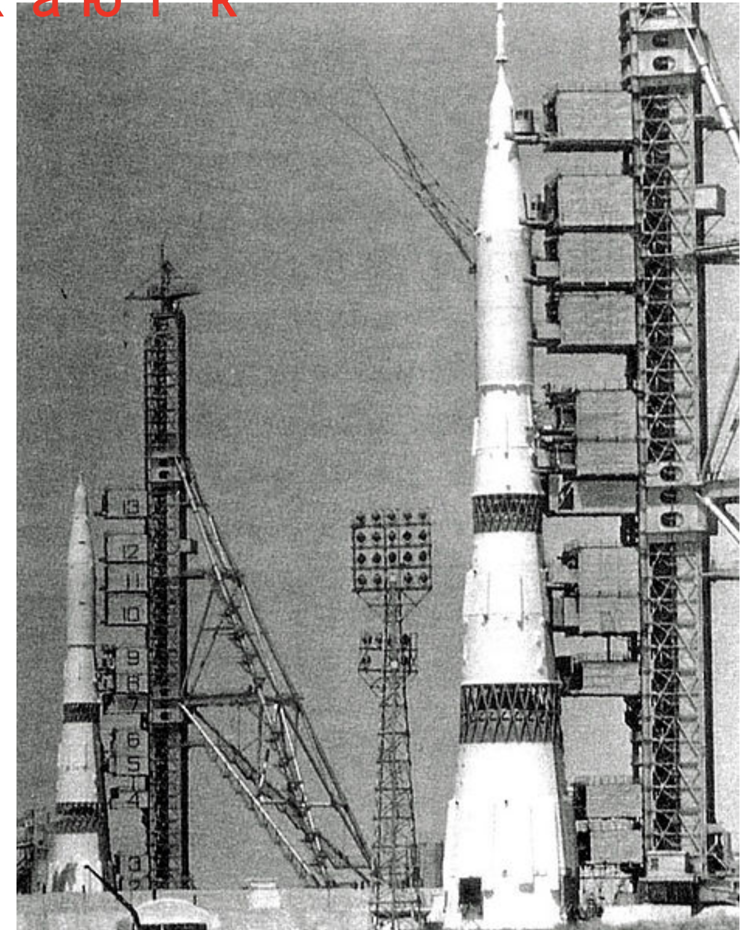
XILINX ALL PROGRAMMABLE.

Consider the Soviet Union Space Program

“If it doesn’t work, we just don’t admit to it.”

Если оно не работает, мы просто не допускают к нему

- **In the semiconductor industry, we call this "Yield"**
- **We extend it to “Binning”**
- **Sell partial devices as a new catalog line item**
  - 3K RAM
  - Processor without FPU
  - 3 core processor
  - Slow speed grade



XILINX ALL PROGRAMMABLE.

Question

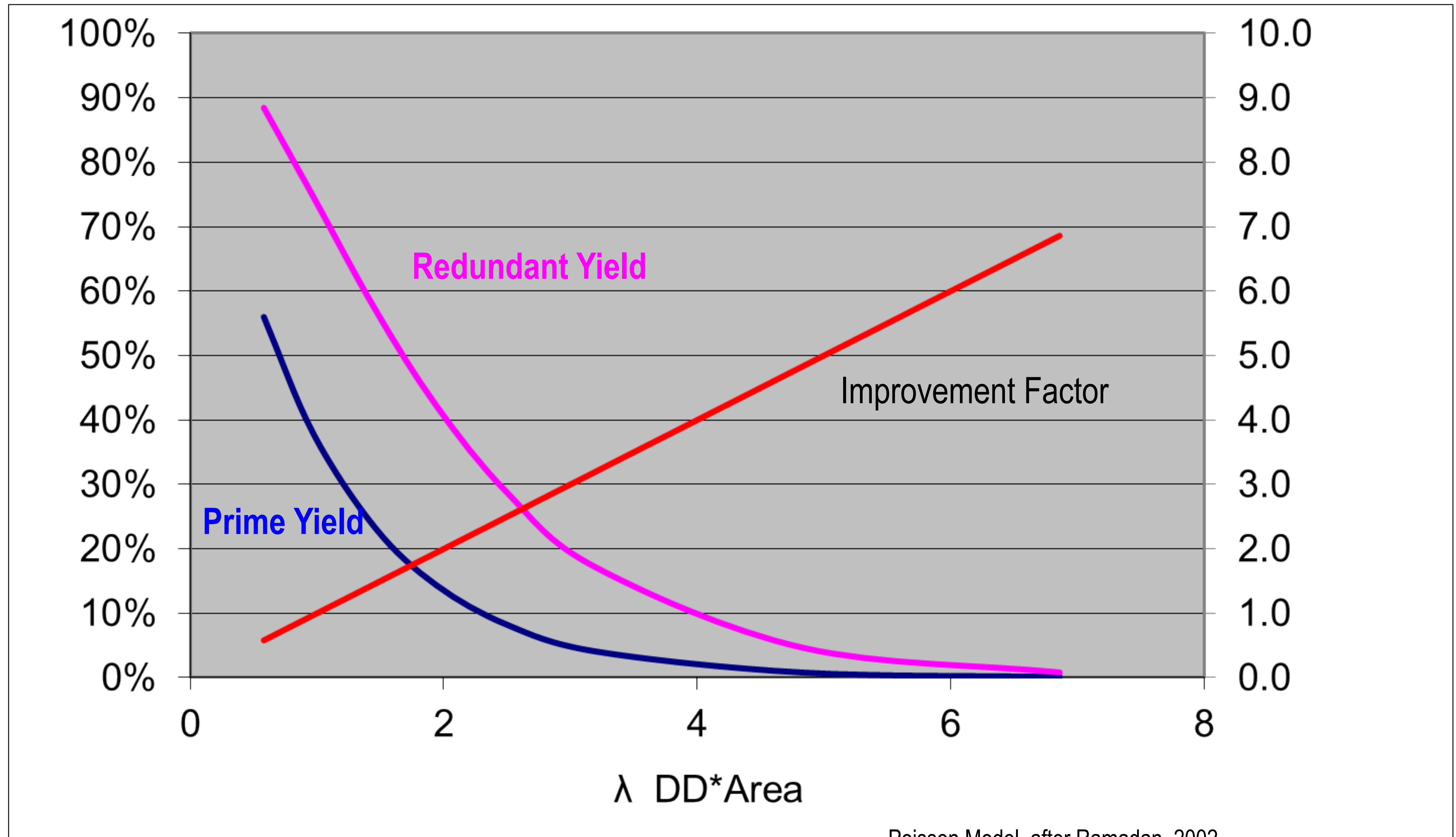
Yield

Why do we throw away bad chips?

What if it doesn't make a difference?

XILINX ALL PROGRAMMABLE.

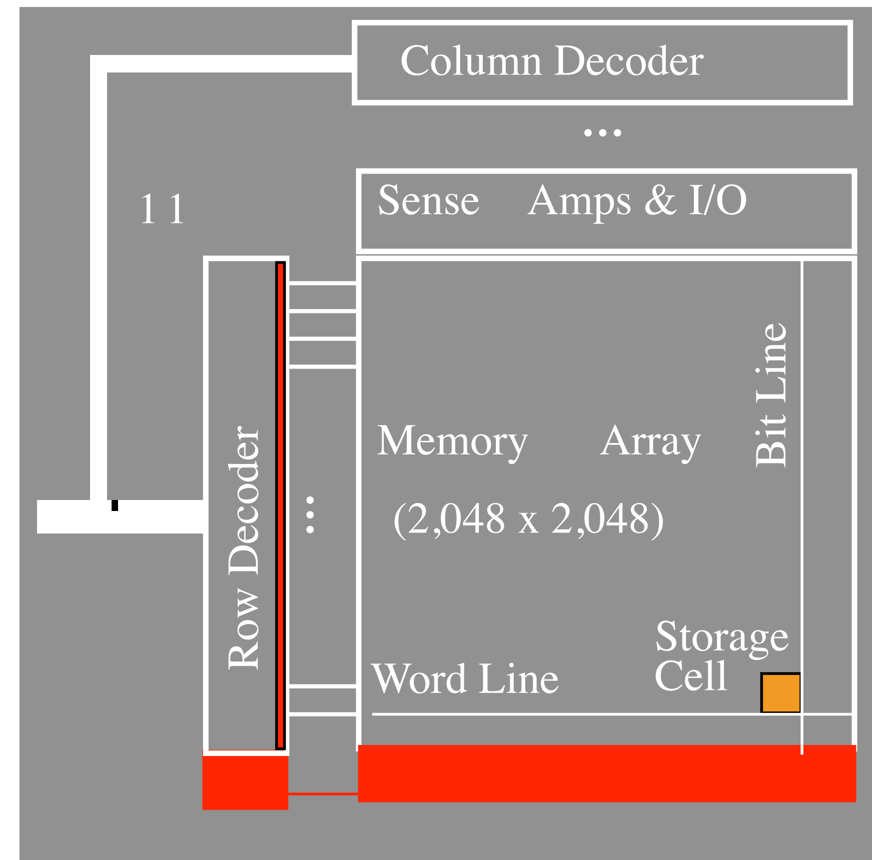
Consider Redundancy



Poisson Model, after Ramadan, 2002

Consider Classical Redundancy

- **Additional area for spares**
  - Fewer gross die per wafer
- **Additional delay for spares**
- **Defect localization**
  - Finding the actual location of a defect is much more difficult than merely identifying the presence of a fault
- **Manufacturing complexity**
  - Non-volatile memory for storing the fault location
- **Algorithm to actually avoid the defect**



XILINX ALL PROGRAMMABLE.

Xilinx Confidential

Consider HP Teramac: Route Around Defects

- **Teramac: Logic Emulator or Supercomputer**
  - Rack full of boards full of partially-defective MCMs full of partially-defective (custom) FPGAs
- **Map defects (one time)**
- **Compile around defects**

- **That works OK for a computer, but a unique bitstream for each chip requires huge SW runtimes and it is a logistical nightmare.**

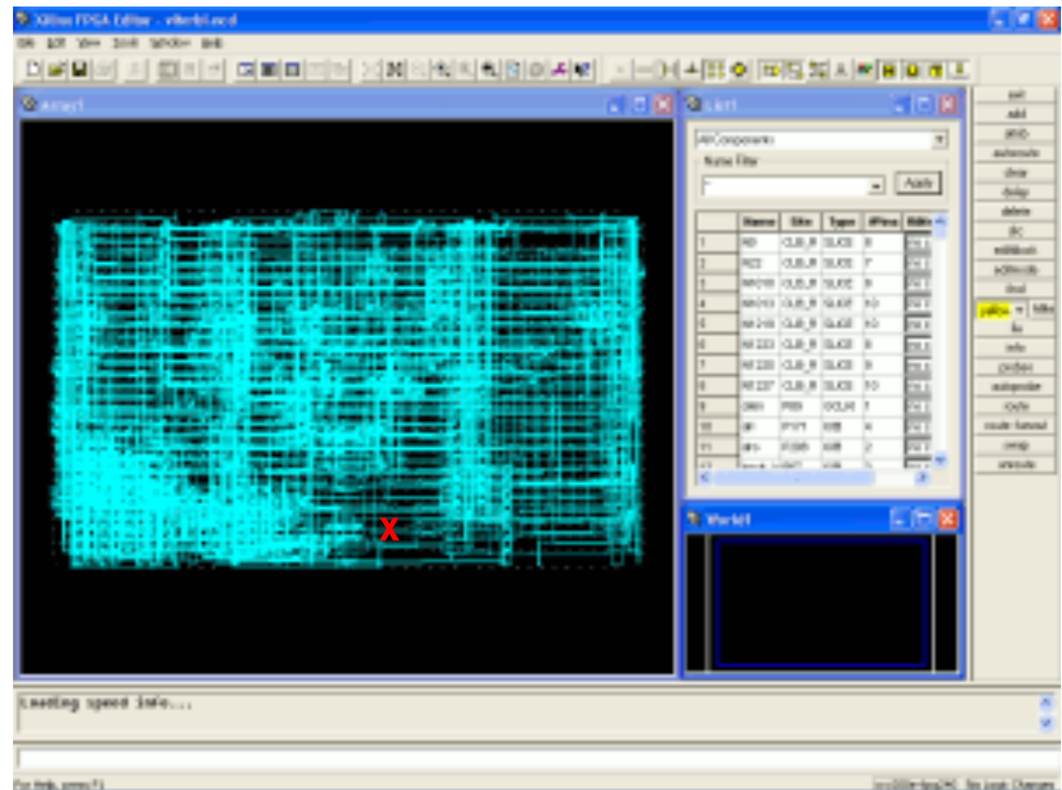
Keukes, 1997

XILINX ALL PROGRAMMABLE.

Consider Xilinx EasyPath™

Cost reduction with no new masks, no new layout, no re-qualification.

- **After the design is done**
- **Generate a custom test program that matches the design**
- **Select FPGA devices that pass the test program**



- **The chips may have defects in unused areas.**
  - That's true for all devices.

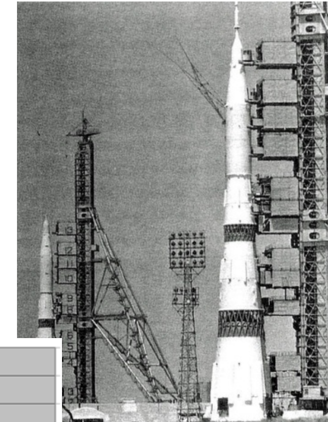
- ***Make a bin for each design***
  - *Instead of pre-defining them*

FPGA

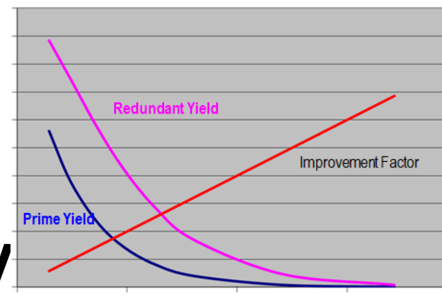
Customer A

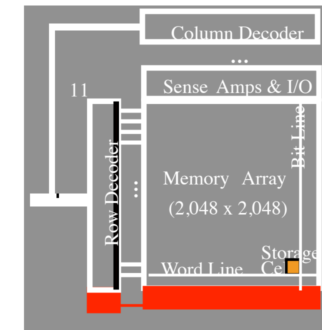
Customer B

What We've Considered…



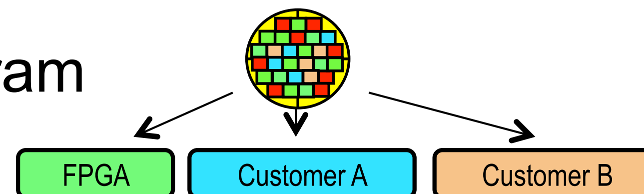
- **The Soviet Union space program**
  - Binning to a pre-defined catalog
- **Redundancy**
  - Essential for large λ





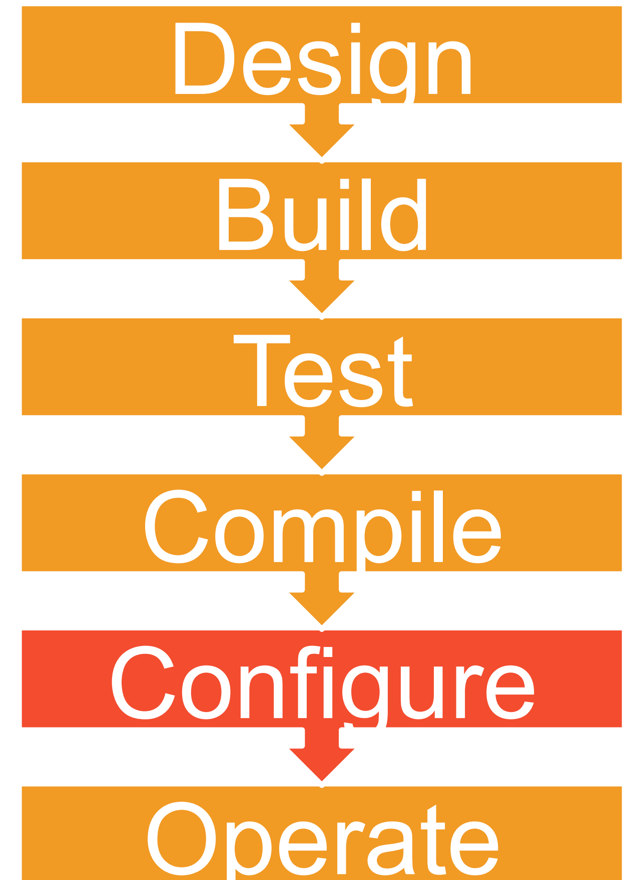
- **Classical redundancy**
  - Overhead: area, performance, NVM, fault localization
- **HP Teramac: route around defects**
  - No classical redundancy overhead, but need to recompile for each device

- **Xilinx EasyPath**
  - Design-specific test program
  - Bin to the specific design



Now Here's What I Want to Talk About…

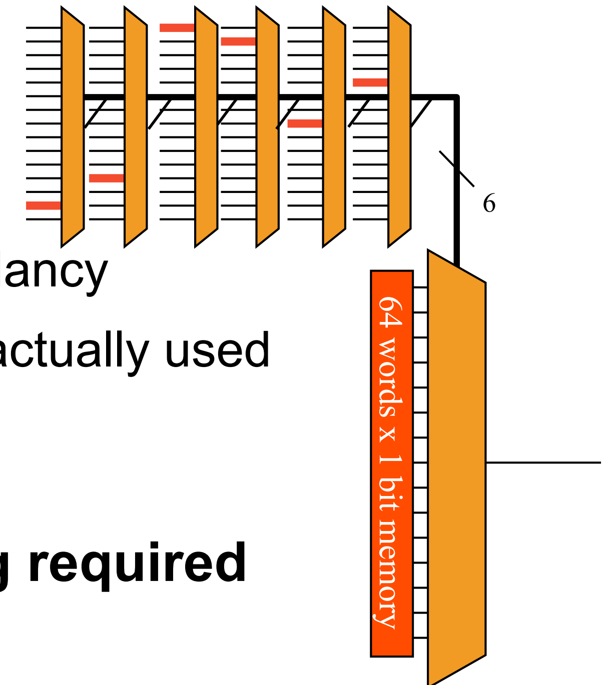
- **Classical Redundancy, Binning and EasyPath are Defect Avoidance at *Test* Time**
- **Teramac is Defect Avoidance at *Compile* Time**
- **What about Defect Avoidance at *Configuration* Time?**



XILINX ALL PROGRAMMABLE.

Defect Avoidance at Configuration Time

***During configuration of the FPGA in the field*, the design is checked and corrections applied to avoid failures.**

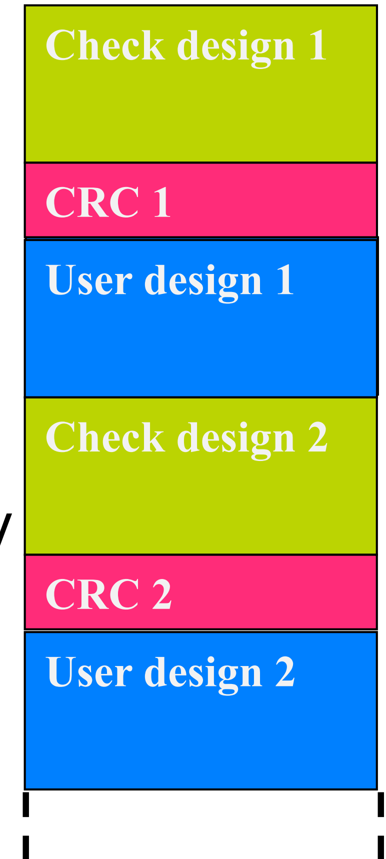


- **VERY fine-grain redundancy**
  - Individual wires, MUX inputs, LUT bits
  - Programmable logic has significant inherent redundancy
  - At the small scale, a small fraction of the device is actually used
- **"No" silicon overhead – no (explicit) spares**
- **No non-volatile memory or laser programming required**
- **No fault localization required**
  - The test is performed on the design as it is loaded into the FPGA
- **But we must identify and fix failures on the chip in the field**
  - Whole test program inside the chip?
  - Compile inside the chip?

XILINX ALL PROGRAMMABLE.

An Implementation

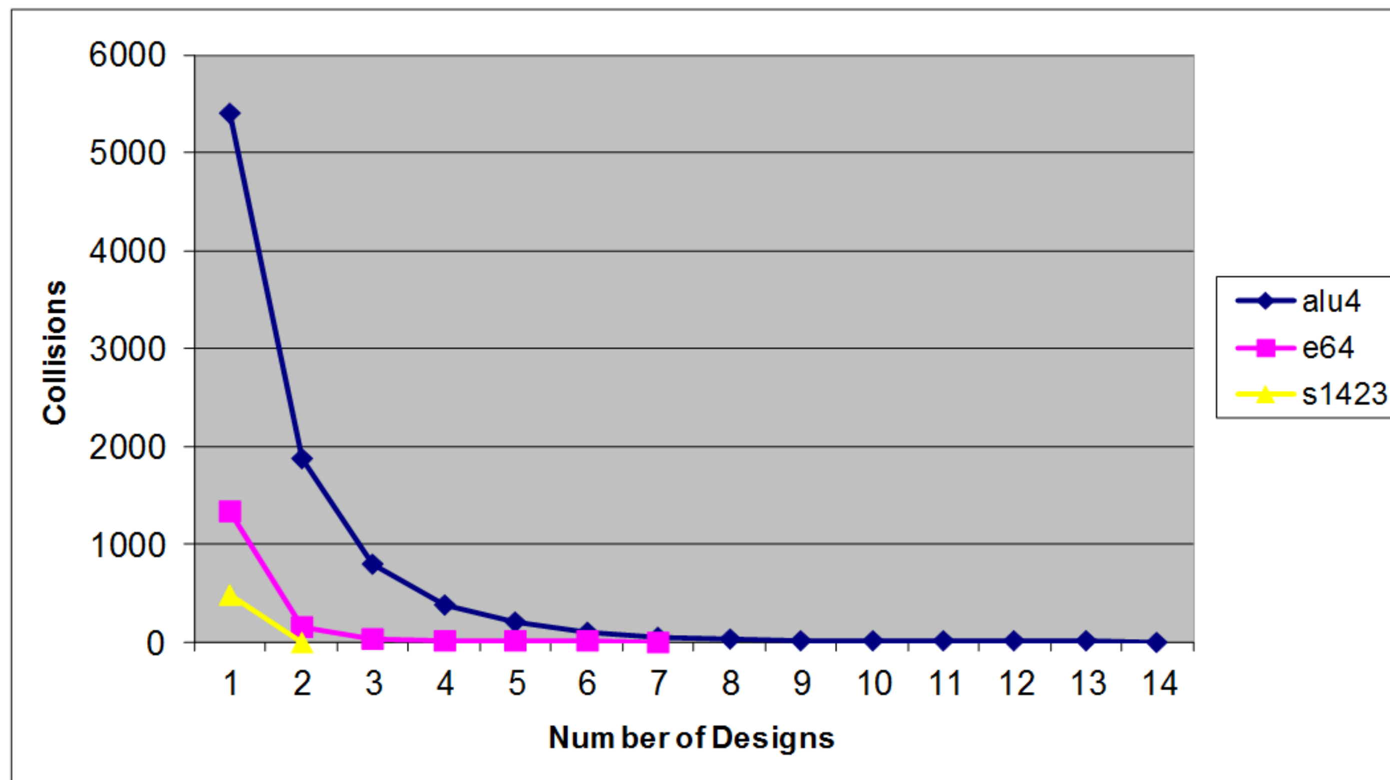
- **Prepare *multiple* bitstreams that all meet the same specification for a function**
  - Each bitstream uses a different subset of the FPGA
  - Easypath, but with multiple attempts
- **At configuration time**
  - Chip applies tests for each *bitstream*
  - Chip loads the first bitstream that meets functionality and performance
- **10-12 bitstreams are required to meet AQL**
  - Assume each bitstream uses 25% of the available configurable resources (mux inputs, LUT bits, …)
  - $P(fail_{all}) = P(fail_1)^n = 0.25^{10} < 1$ ppm
  - We verified this experimentally



XILINX ALL PROGRAMMABLE.

Preparation of Multiple Bitstreams

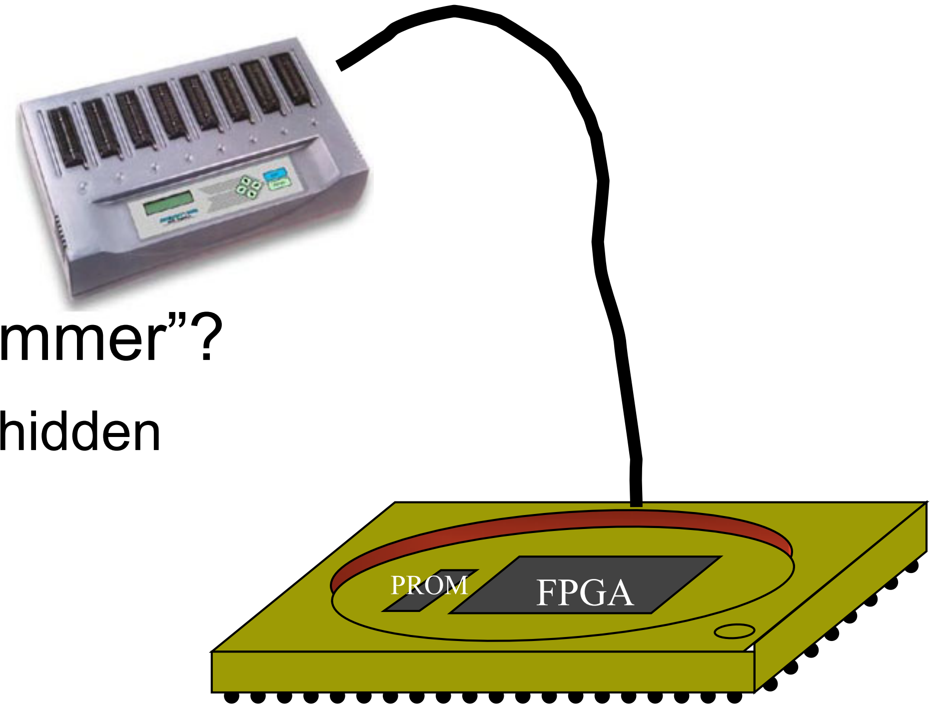
- Minimize the number of “avoidable” features that are used in *every* implementation (“collisions”).



Xilinx Confidential

XILINX ALL PROGRAMMABLE.

Some Options and Open Questions



- **Route around defects**
  - Put the router on-chip?
  - Put the router in the "programmer"?
    - Configuration selection time is hidden
- **Efficient test needed**
  - Can it be done with few self-test configurations?
  - Can it be done with BIST?

XILINX ALL PROGRAMMABLE™

Defect Avoidance Summary

- Generate multiple bitstreams for a customer design
- When configuring, check the device, select the bitstream that avoids defects on the device
- Defect-avoidance algorithm can be in a device programmer, and therefore hidden from customers

- "An extension of post-manufacture voltage selection to functionality."

Xilinx Confidential

Application to Reliability / Field Failures / Performance Degradation

- **Offline. Re-select periodically to avoid field failures**
- **Addresses parametric degradation as well as outright failures**
  - Possible to tighten test guardband
- **Possible to address environmental conditions**
  - Alternate FPGA programming pre-built for different environments
  - What device triggers are required?
    - Temperature, voltage fluctuation, recent-failure count
- **We can do more than just DVFS**

XILINX ALL PROGRAMMABLE™

Some Lingering Problems

- Compile times
- Configuration times
- System qualification concerns

XILINX ALL PROGRAMMABLE™

Defect Avoidance

- It works with hard real-time constraints
- No need to throw away bad chips
- No need to guard-band against worst-case, longest-term operation
- We can do much more than just DVFS

XILINX ALL PROGRAMMABLE™