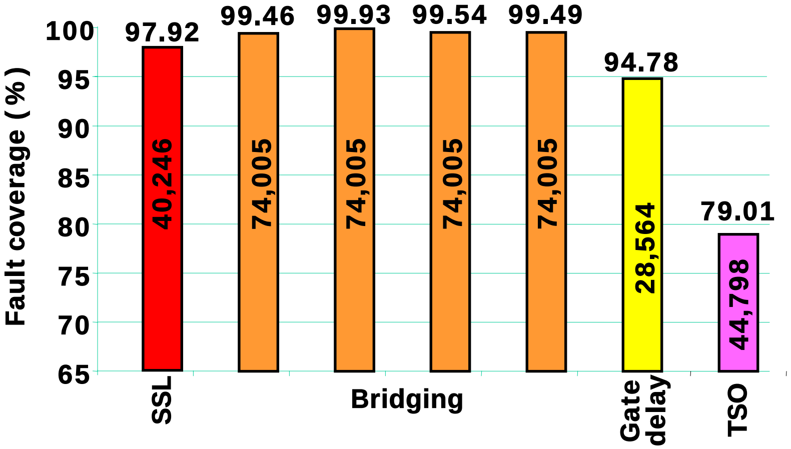
Shawn Blanton

Professor
Department of ECE
Center for Silicon System Implementation
CMU Laboratory for Integrated Systems Test
http://www.ece.cmu.edu/~blanton

Courses:

18-240:	Fundamentals of Computer Engineering
18-340:	Digital Computation
18-441:	Verification of Computer Hardware Systems
18-765:	Digital Systems Testing and Testable Design

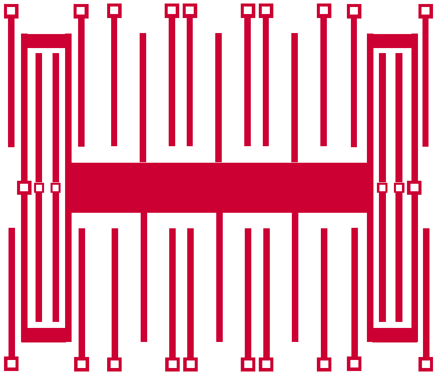
Test and Diagnosis of ICs



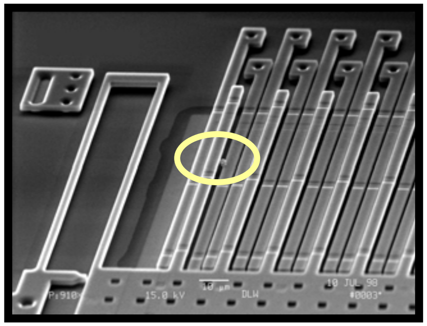
Test grading with **FA**ult **T**uple **SIM**ulator

- Defect modeling
- Test methodology development
- CAD tool development
- Defect diagnosis

MEMS Test



μmechanical structure



μmechanical structure affected by a defect

- Defect modeling
- Built-in self test (BIST) design
- Test methodology development

Electrical & Computer ENGINEERING

CSSI Center for Silicon System Implementation

Digital Circuit Design

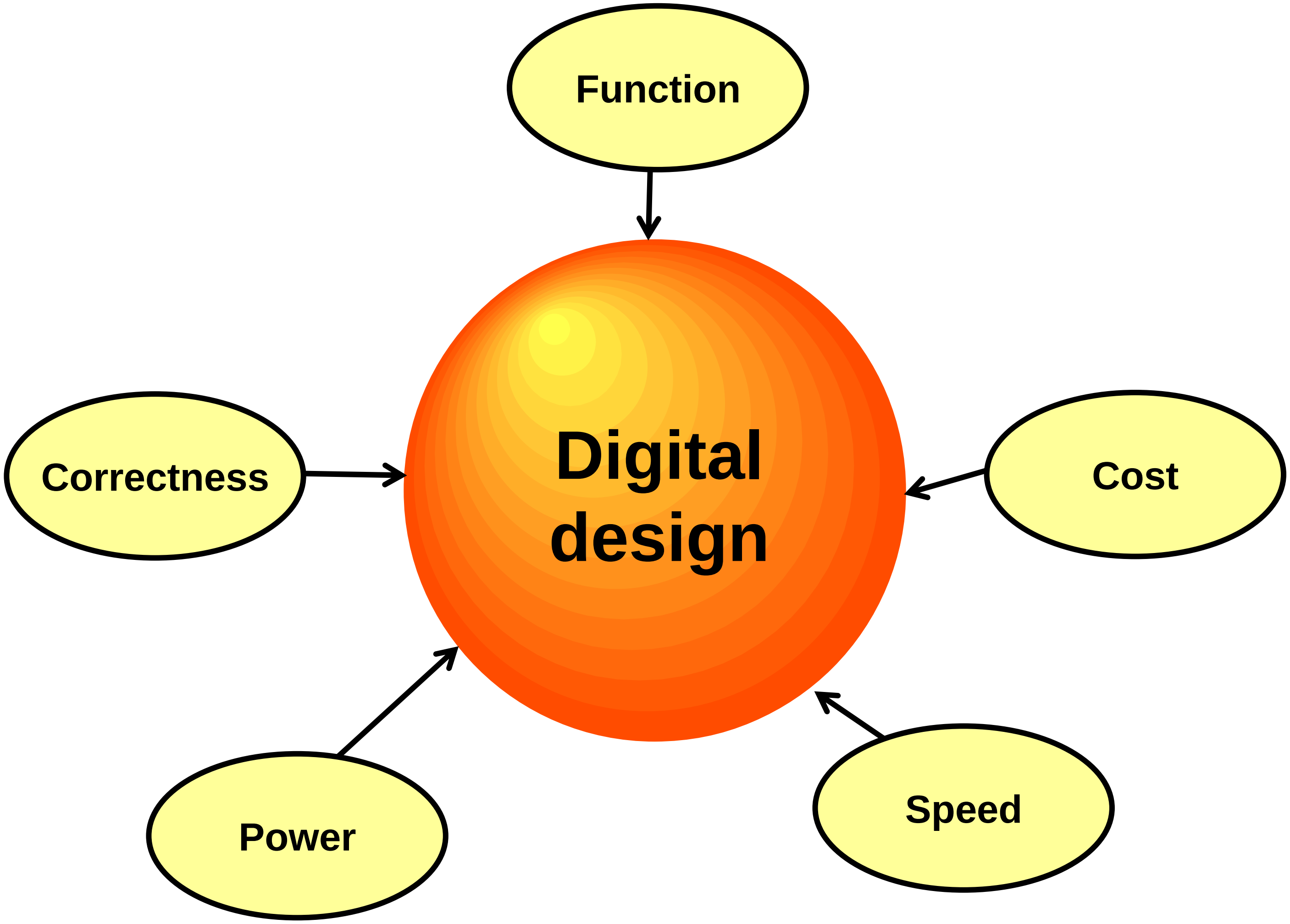
Emerging Trends in Electrical and Computer Engineering

Shawn Blanton

Fall 2004

blanton@ece.cmu.edu

Components of Digital Design



Function

Correctness

Digital design

Cost

Power

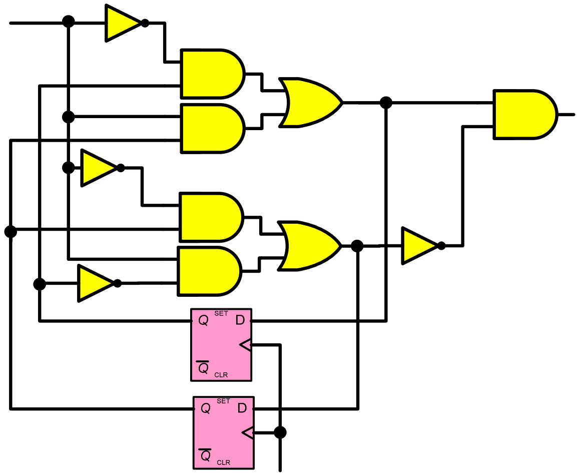
Speed

Function

- When we design a circuit to do a particular function, we assume we have primitive gates like AND, OR, *etc*.
- So we build bigger things from smaller things.
- And we have systematic methods (*i.e.*, CAD tools) for building bigger things.

y1	*y2*		*x* = 0	*x* = 1
0	0	A	A,0	B,0
0	1	B	B,0	C,0
1	1	C	C,0	D,1
1	0	D	D,1	A,0

K-maps
Quine-McCluskey
Boolean Algebra
Synplicity
etc.



Function cont…

- But what if you wanted to build a fast, 64-bit adder?
- What about using a ripple-carry adder?
- Systematic techniques cannot handle some things we want to build.

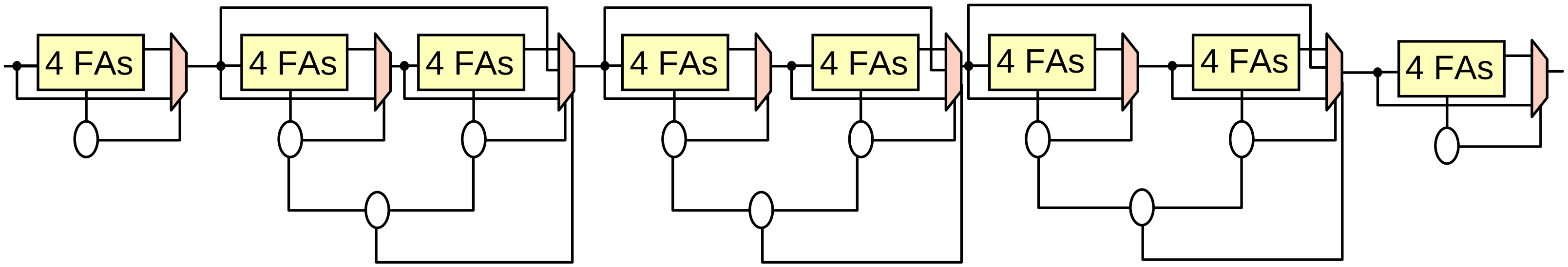
How many rows are in the truth table for a 64-bit adder?

0000000000000000000...000000000000000000000000000000000000000000000	0
0000000000000000000...000000000000000000000000000000000000000000001	1
0000000000000000000...000000000000000000000000000000000000000000010	1
0000000000000000000...000000000000000000000000000000000000000000011	0
0000000000000000000...000000000000000000000000000000000000000000100	1
0000000000000000000...000000000000000000000000000000000000000000101	1
0000000000000000000...000000000000000000000000000000000000000000110	0
: : : :	:
: : : :	:
1111111111111111111...111111111111111111111111111111111111111111111	1

Function cont…

- There are many known adder architectures.
- To learn how to design an adder for your given application, you must "see" these adders and understand how they work.

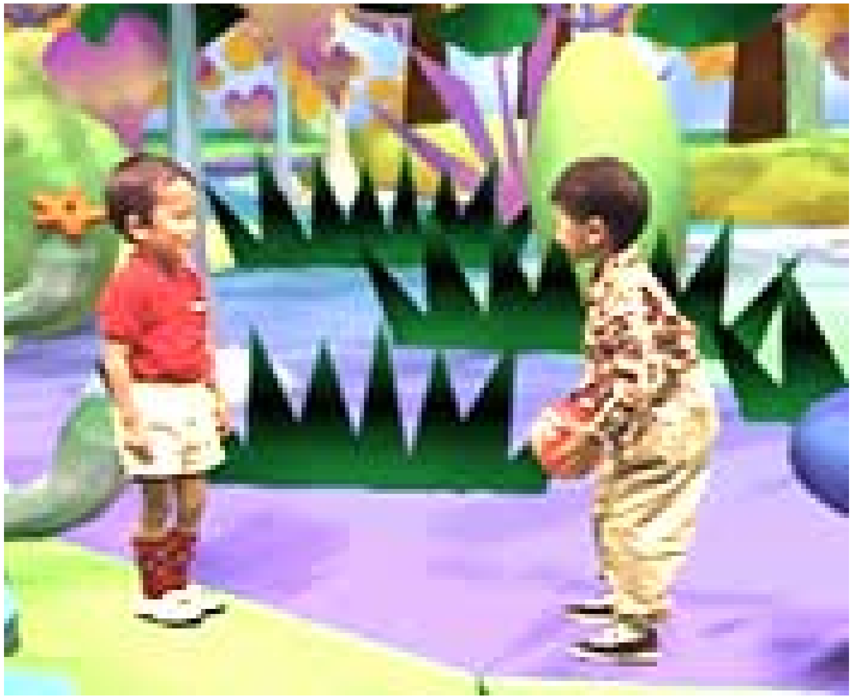
32-bit carry-skip adder



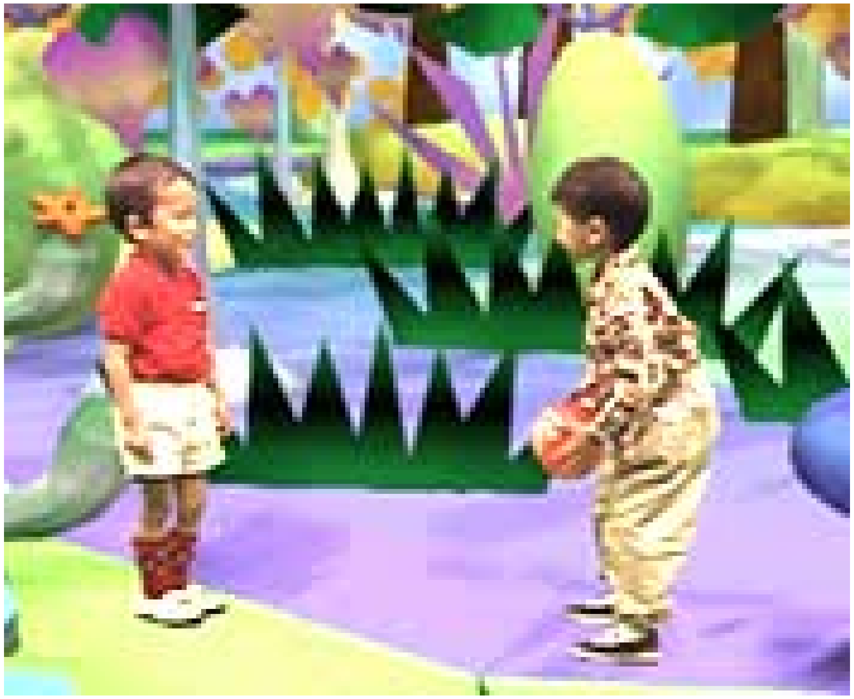
- The same is true for other *data-processing* circuits: multipliers, dividers, filters, *etc*.

Customized Floating Point Unit

64-bit floating point

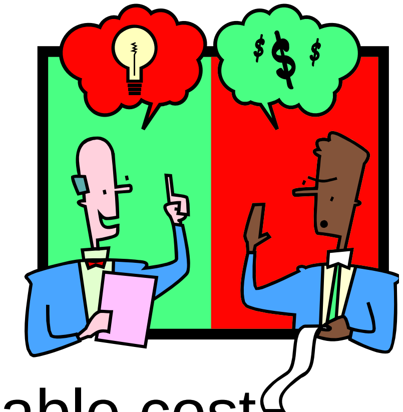


13-bit custom floating



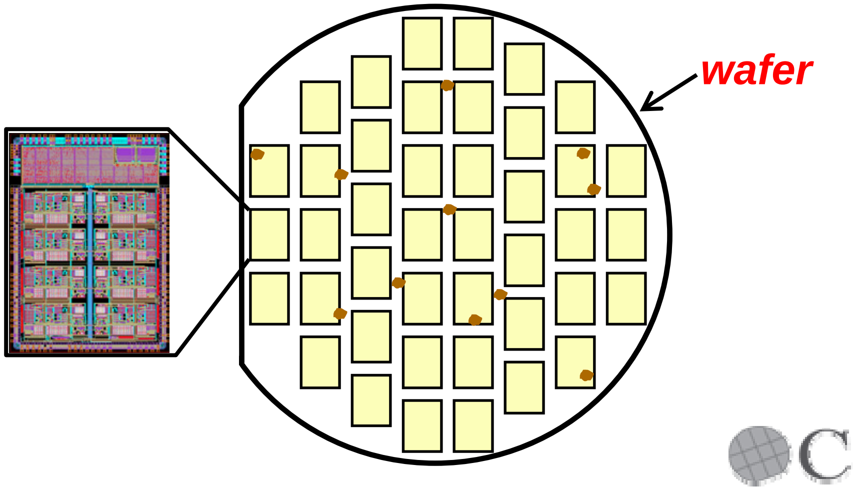
Courtesy of Prof. Rob Rutebnar

Cost $$$

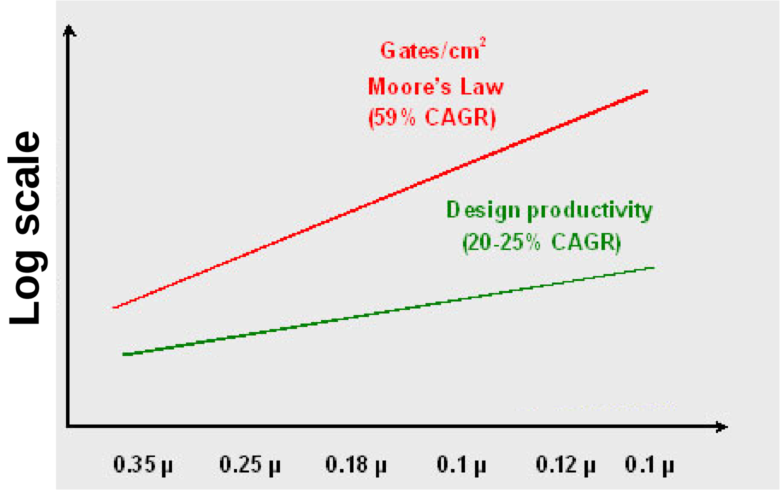


- Many ideas you may have are likely possible.
- But as engineers, it must be possible at a reasonable cost.
- One type of cost has to do with design size.
- Ideally, you want a circuit implementation to take very little “area” for a number of reasons:
  - Small circuits means more can fit onto a “wafer”.
  - Small circuits have a higher “yield”.

$$\text{Yield} = \frac{\text{good chips}}{\text{Total chips}}$$



Cost $$$ cont…



- The number of gates available in the same area is doubling every 18 months (“Moore’s Law”).

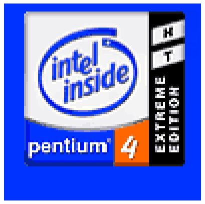
- Designer productivity is not increasing at the same rate. This leads to the “design gap”.

- Solution: Better CAD tools for design and analysis!

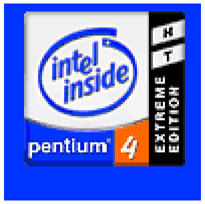
Speed

- Your new adder design may “add”, but does it add fast enough?
- Digital design requires you to understand how to accurately predict the speed of your circuit.
- Lots of factors affect speed:
  - Design structure
  - Fabrication technology
  - Environment
- Actually, you cannot really predict speed that accurately.

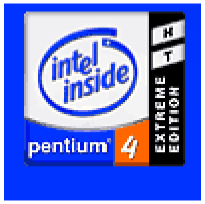
4GHz



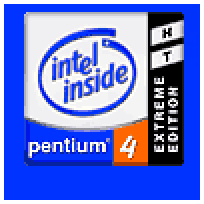
3.66GHz



3.1GHz

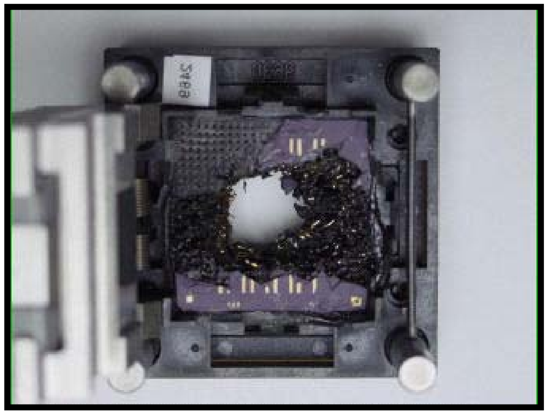


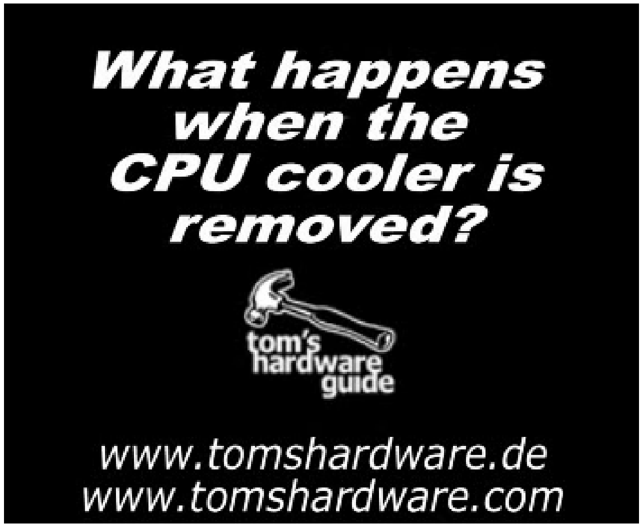
2.6GHz



Power

- Power is now the number one challenge for digital circuit designers!
- Why? Many electronic systems are portable.
- Low power means longer battery life.
- High power generates lots of heat so designs must be cooled.





What happens when the CPU cooler is removed?

tom's hardware guide

www.tomshardware.de
www.tomshardware.com

Correctness

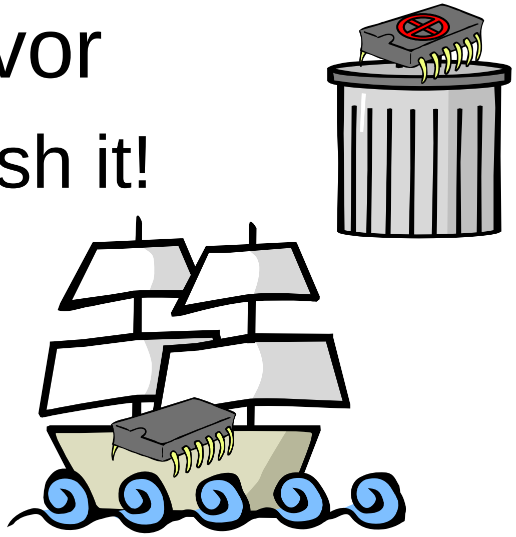
- Designers make mistakes.
- How do you find errors in a design?
- In 18-240, your method for finding mistakes is either *ad hoc* or impractical.
- You need systematic approaches that utilize CAD tools that include simulators, equivalence checkers, model checkers, *etc*.
- Big companies still make mistakes. Intel made an error designing a divider circuit. It cost Intel **$470M**.

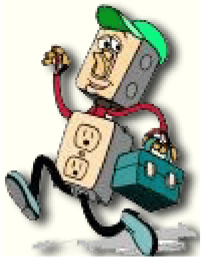
Correctness cont…

- Manufacturing is not perfect.
- How do you find imperfections in the fabricated design?
- You must test each and every chip to ensure it works.

Test and Diagnosis

- Test is a binary endeavor
  - – Chip doesn't work – trash it!
  - – Chip works – ship it!





- Diagnosis
  - – Why doesn't the chip work???
  - – Required for silicon debug, process tuning, *etc*.

What do you test for?

- Test the chip function?
  - Portions of the chip are tested this way.
  - Too costly however for the entire chip.

- It is more cost-effective to test for the things that can go wrong.

What can go wrong?

One or more defects can occur!

- A defect can be extra or missing material or changes in material properties
  - Defects can affect wires and vias
  - Defects can affect transistors
- Defect-based test is impractical
  - Too many possible locations (billions)
  - Too many different types (N × billions)
- To keep test manageable, abstraction is necessary

Defect Abstraction

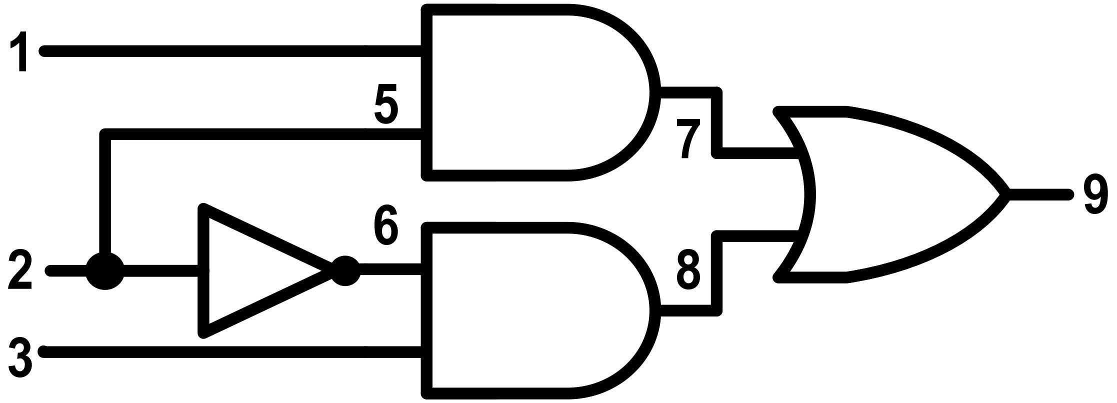
- Abstraction involves
  - – Moving from physical level to higher level
  - – Reducing number of defects
  - – Simplifying defect behavior

- Defect abstraction ≡ fault model

SSL Fault Model

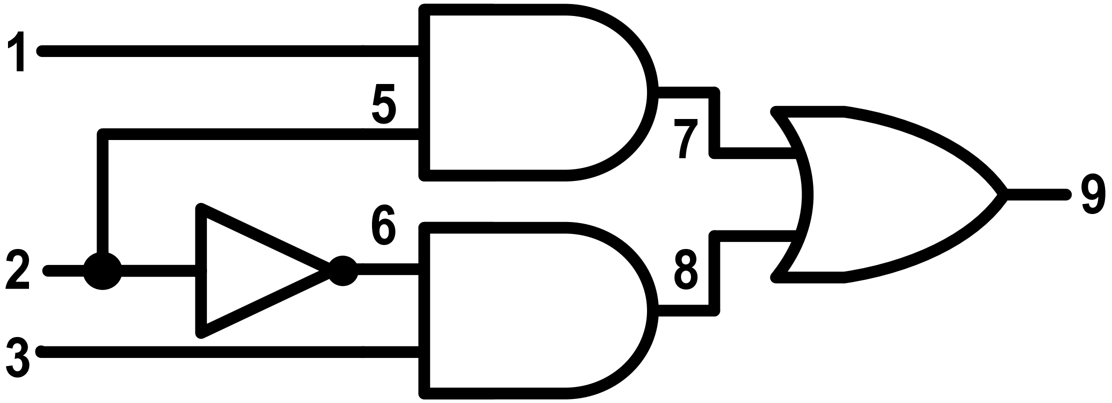
- The most widely-use abstraction is single stuck-line (SSL) fault model.
- First published use was for vacuum tube test in 1957 by R. Eldred.
- SSL fault model properties:
  - a gate-level model
  - assumes gates are unaffected
  - only a single logic line can be faulty
  - faulty line is permanently stuck-at 0 or 1

Stuck-at Testing



- The circuit above has 9 logical lines.
- Each line is susceptible to either a stuck-at-0 or stuck-at-1 fault, for a total of 18 faults.
- Must find tests that distinguish the 18 faulty circuits from the "good" circuit.

Stuck-at Testing cont’d



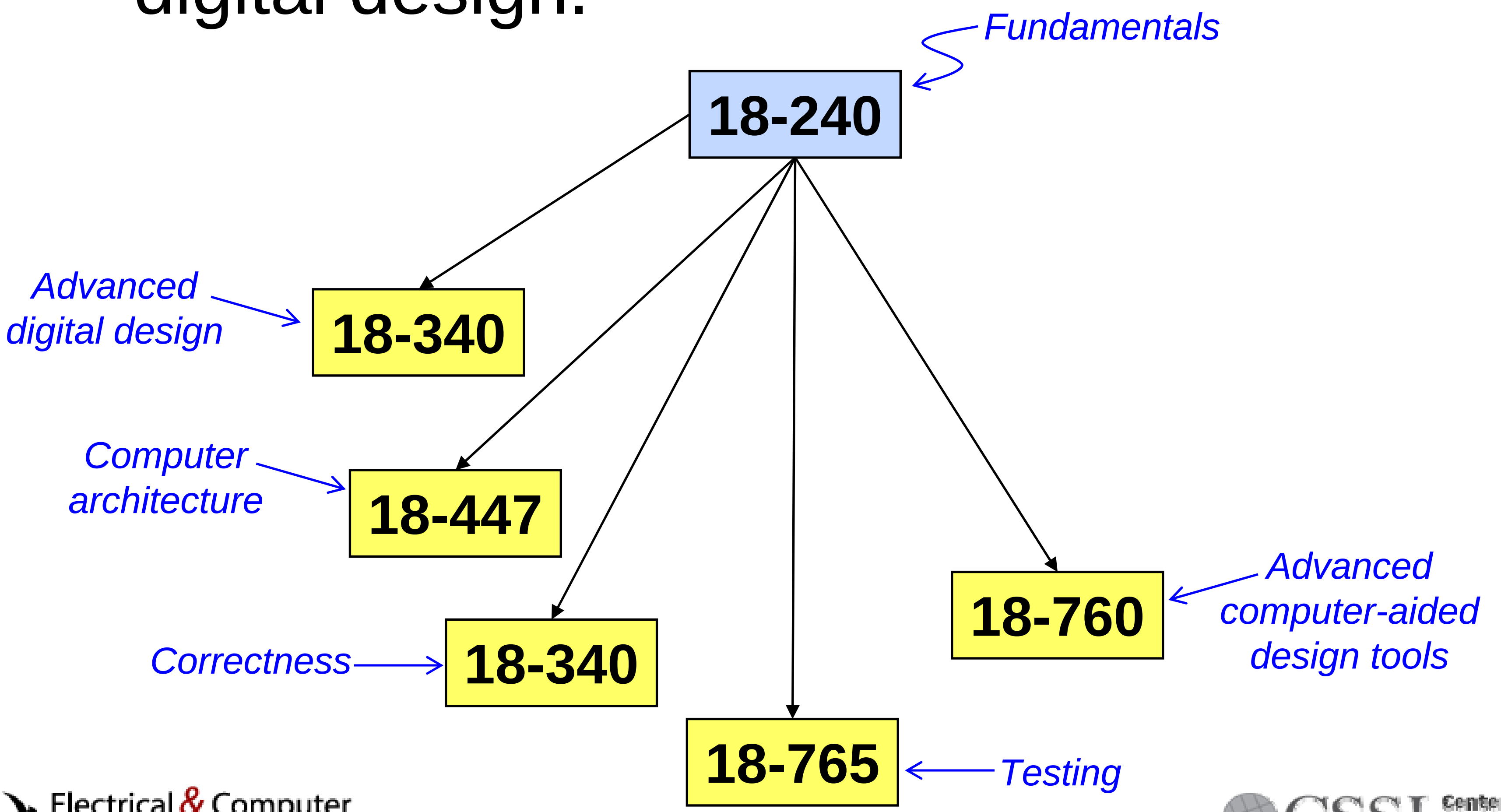
Test set

1. 001
2. 011
3. 100
4. 111

1 2 3	9	9/1	7/1	2/1	9/0	7/0	5/1	8/0	1/1	2/0	3/1	6/1
0 0 0	0	1	1	0	0	0	0	0	0	0	1	0
0 0 1	1	1	1	0	0	1	1	0	1	1	1	1
0 1 0	0	1	1	0	0	0	0	0	1	0	0	0
0 1 1	0	1	1	0	0	0	0	0	1	1	0	1
1 0 0	0	1	1	1	0	0	1	0	0	0	1	0
1 0 1	1	1	1	1	0	1	1	0	1	1	1	1
1 1 0	1	1	1	1	0	0	1	1	1	0	1	1
1 1 1	1	1	1	1	0	0	1	1	1	1	1	1

Digital Design Courses

Learn more by taking various courses in digital design.



Fundamentals

18-240

Advanced digital design

18-340

Computer architecture

18-447

Correctness

18-340

Testing

18-765

Advanced computer-aided design tools

18-760