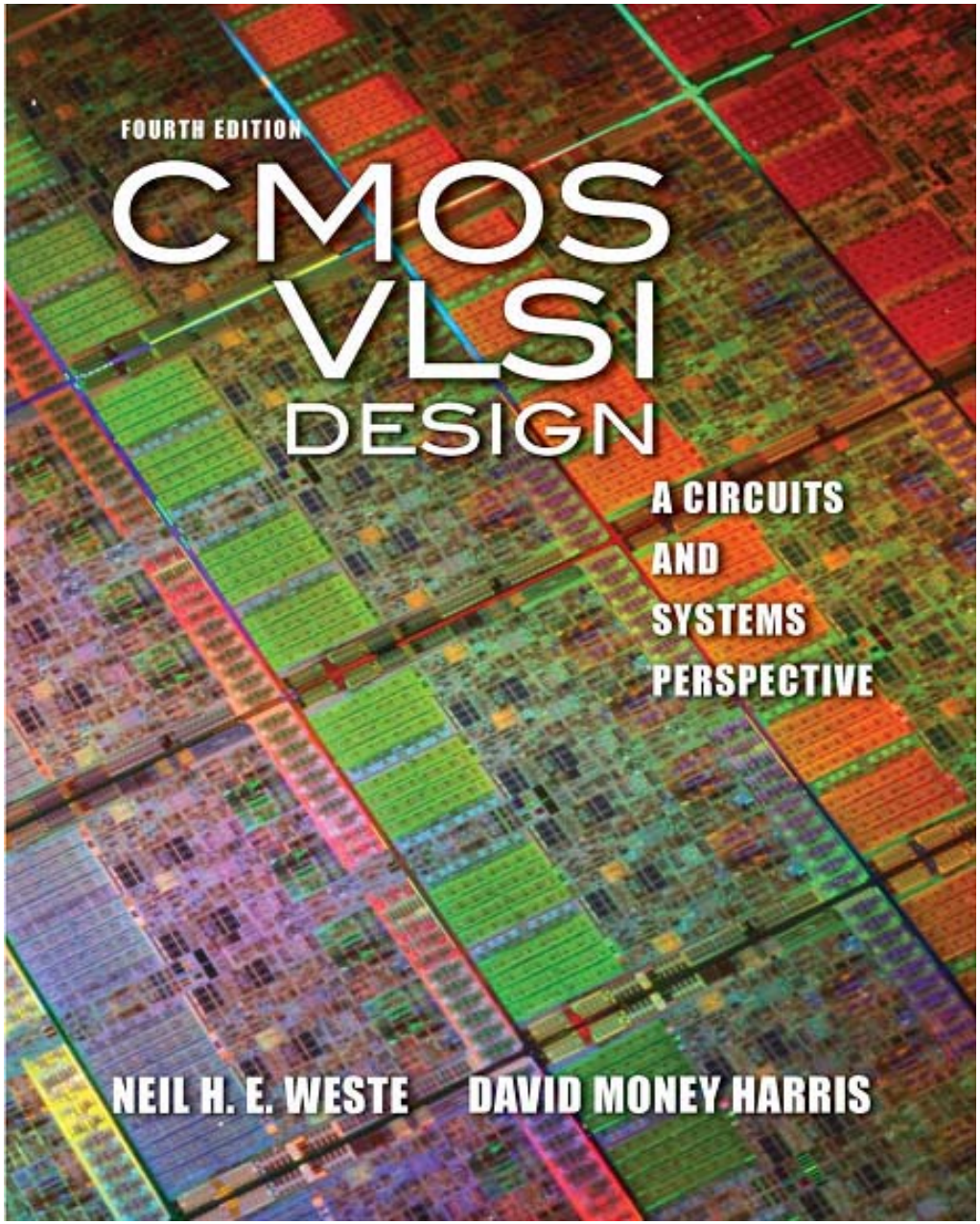


Lab 3 Introduction

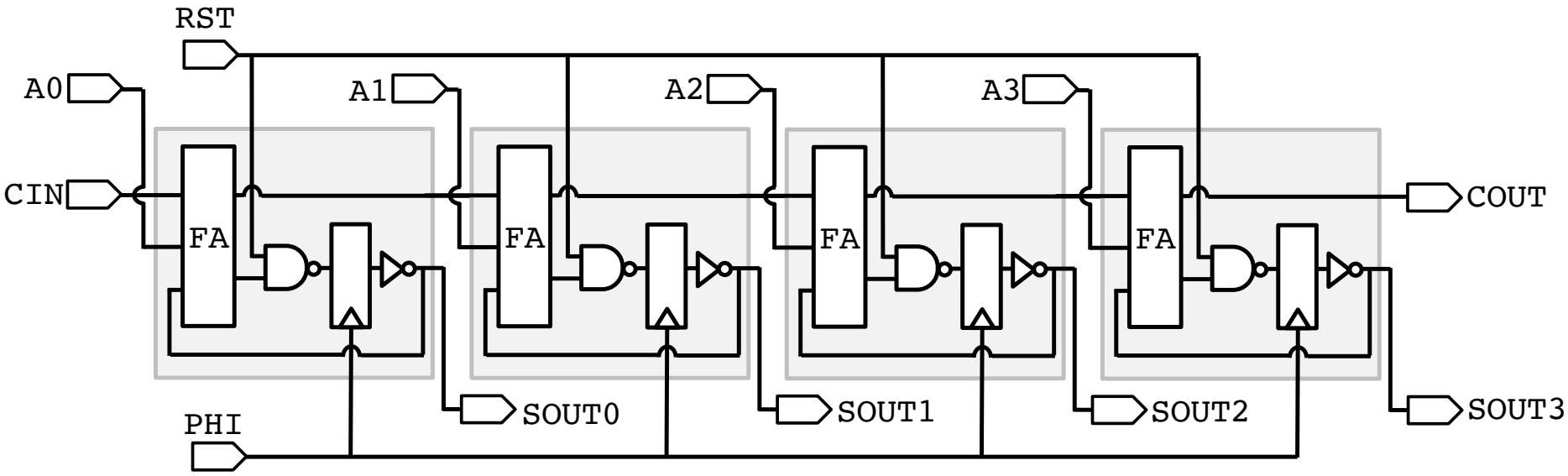
10/20/2020

Design Automation

- Semiconductors are ~$450B/year market
- Approximate Market Capitalization of Major EDA companies:
  - Synopsys ($20B)
  - Cadence ($19B)
  - Mentor ($4B)
- Compare to Semiconductor Manufacturers
  - Intel ($250B)
  - TSMC ($260B)

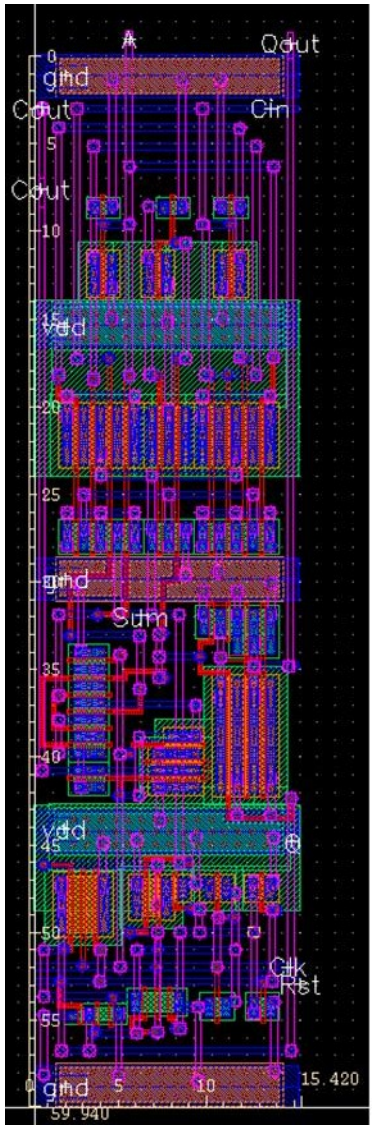
Goal for Lab 3

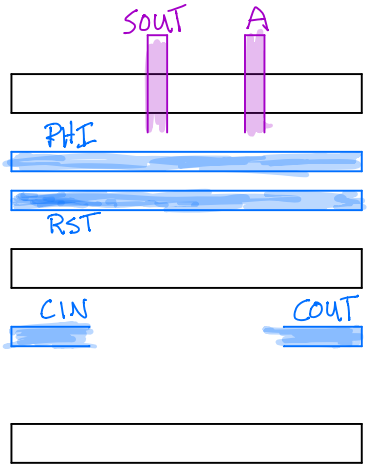
- Create this design in two different ways
  (558 uses half adders instead of full adders)
  1. By instantiating bitslices
  2. Using synthesis and place&route

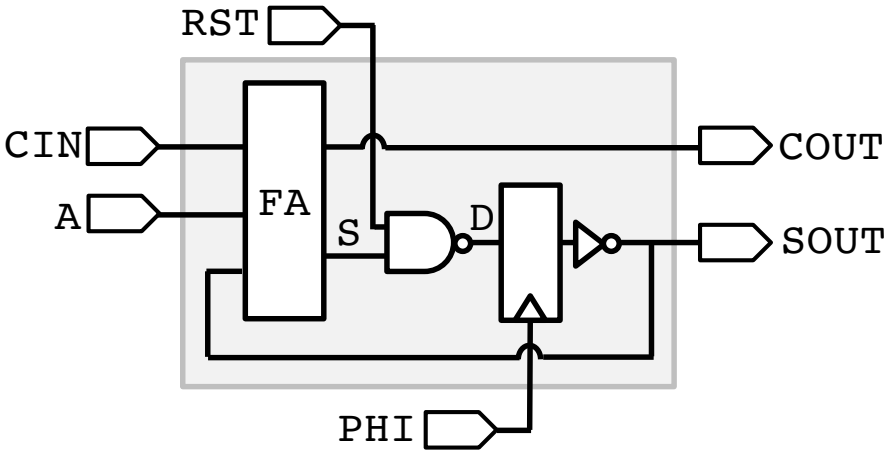


This is a 4-bit ripple carry adder. What is critical path of RCA?

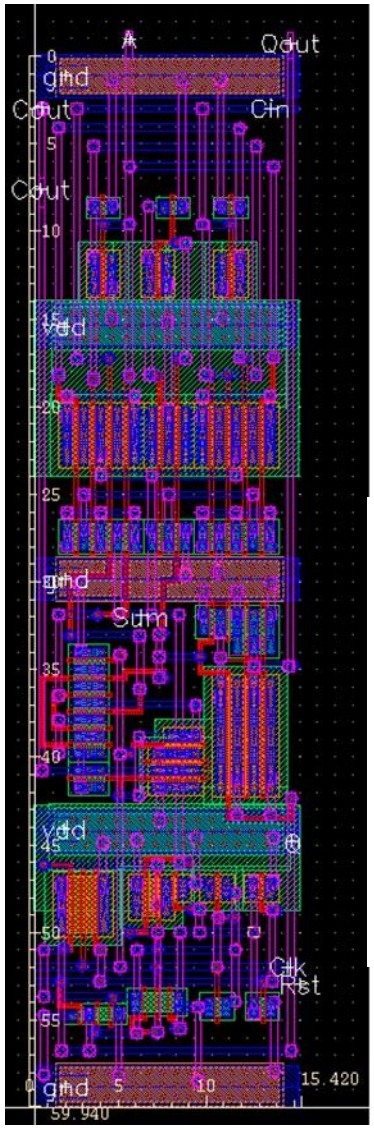
Part A: Instantiating Bitslices

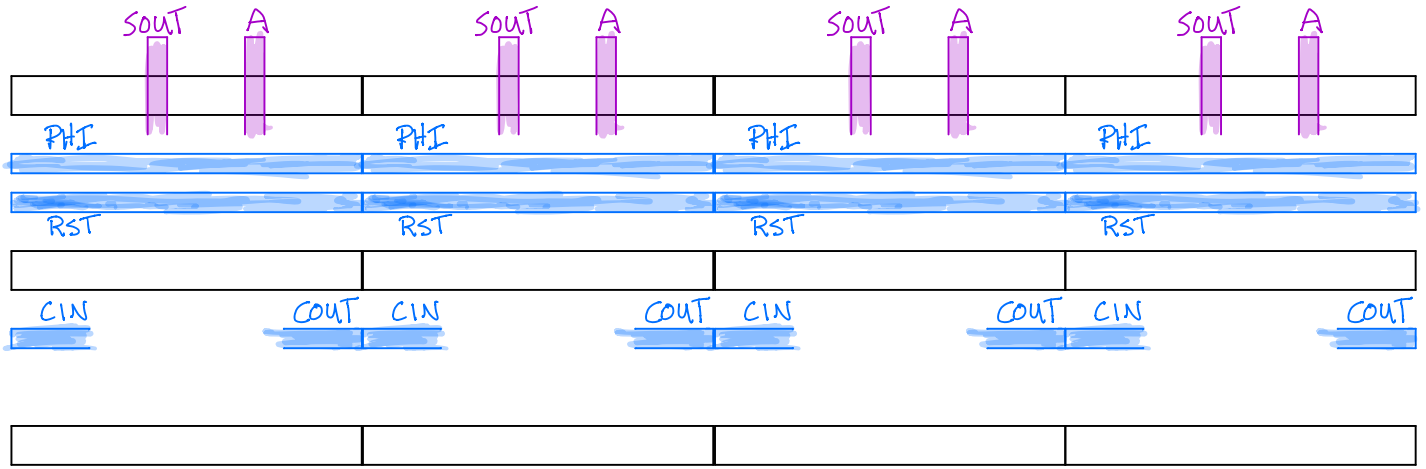


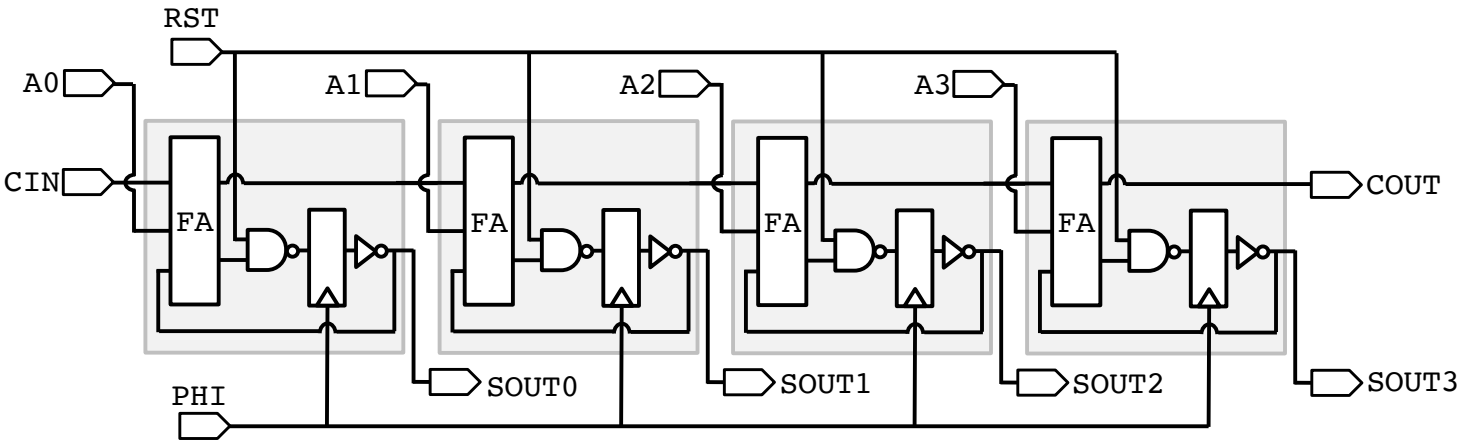




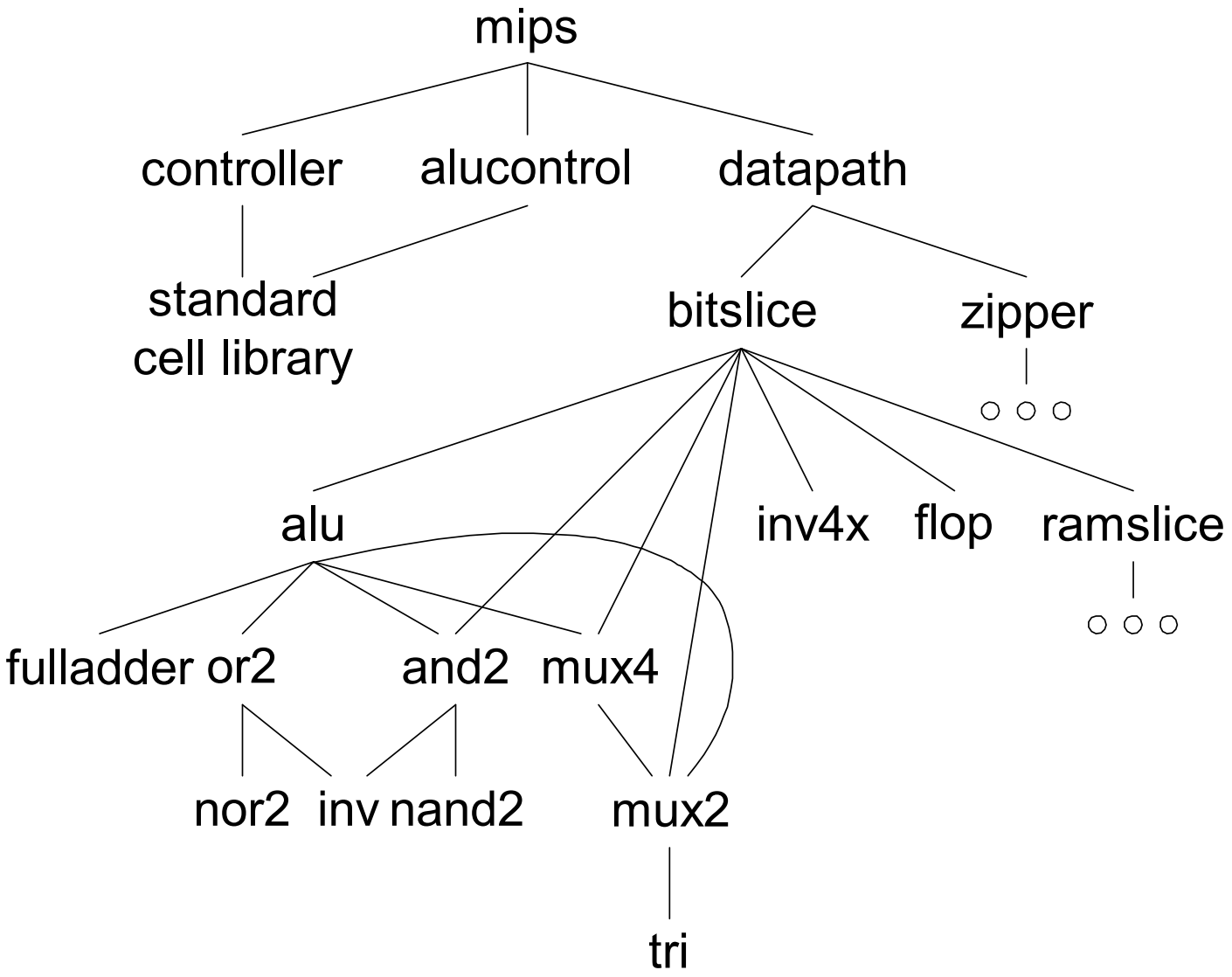
Part A: Instantiating Bitslices





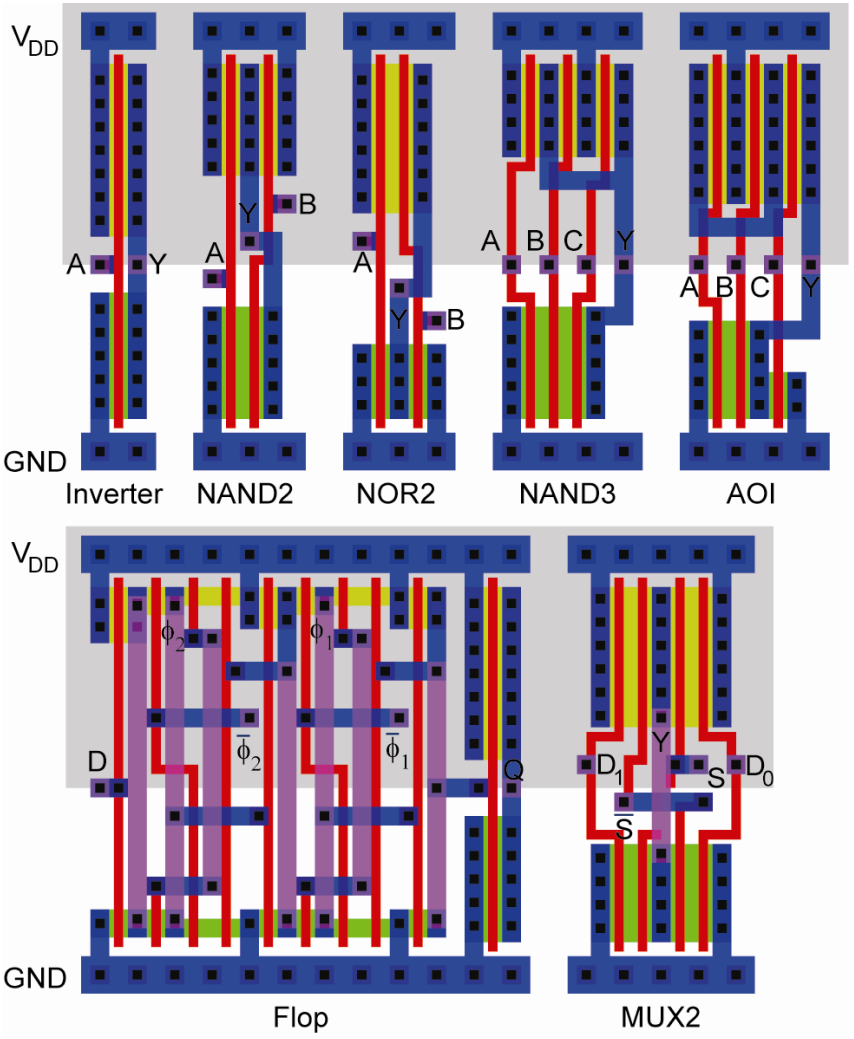


Hierarchical Design

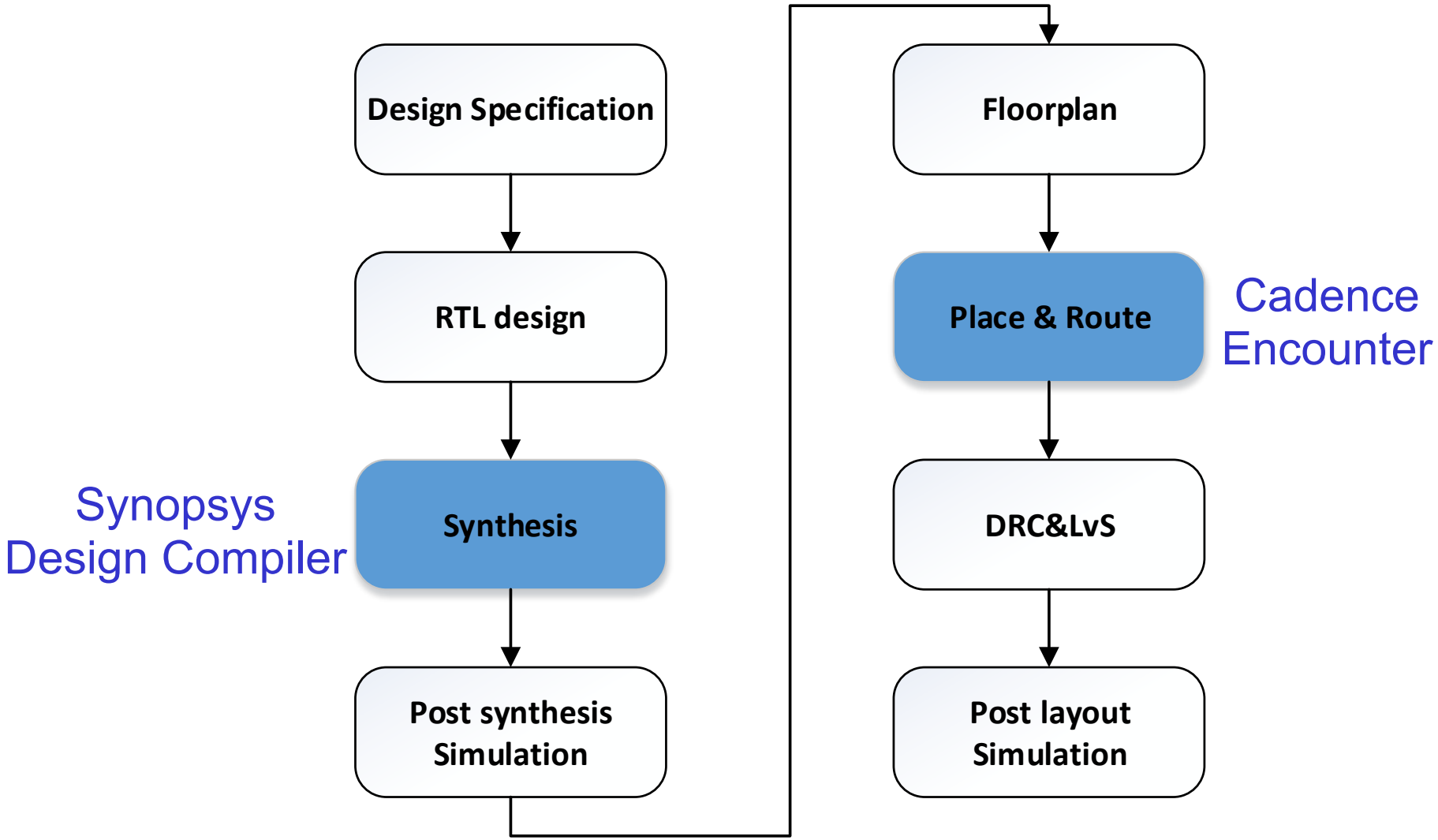


Standard Cells

- Uniform cell height
- Uniform well height
- M1 V_{DD} and GND rails
- M2 Access to I/Os
- Well / substrate taps
- Exploits regularity



Design Automation Flow for Lab 3

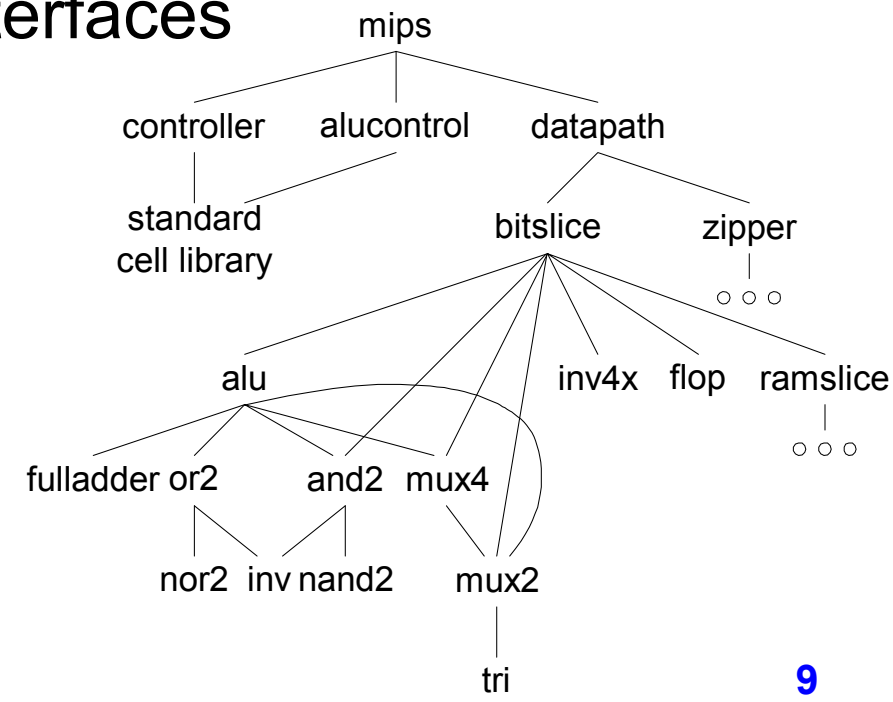


Coping with Complexity

- How to design System-on-Chip?
  - Many millions (even billions!) of transistors
  - Tens to hundreds of engineers
- Structured Design
- Design Partitioning
- On-chip Communication Networks

Structured Design

- **Hierarchy**: Divide and Conquer
  - Recursively break system into modules
- **Regularity**
  - Reuse modules wherever possible
  - Example: Standard cell library
- **Modularity**: well-formed interfaces
  - Allows modules to be treated as black boxes
- **Locality**
  - Physical and temporal

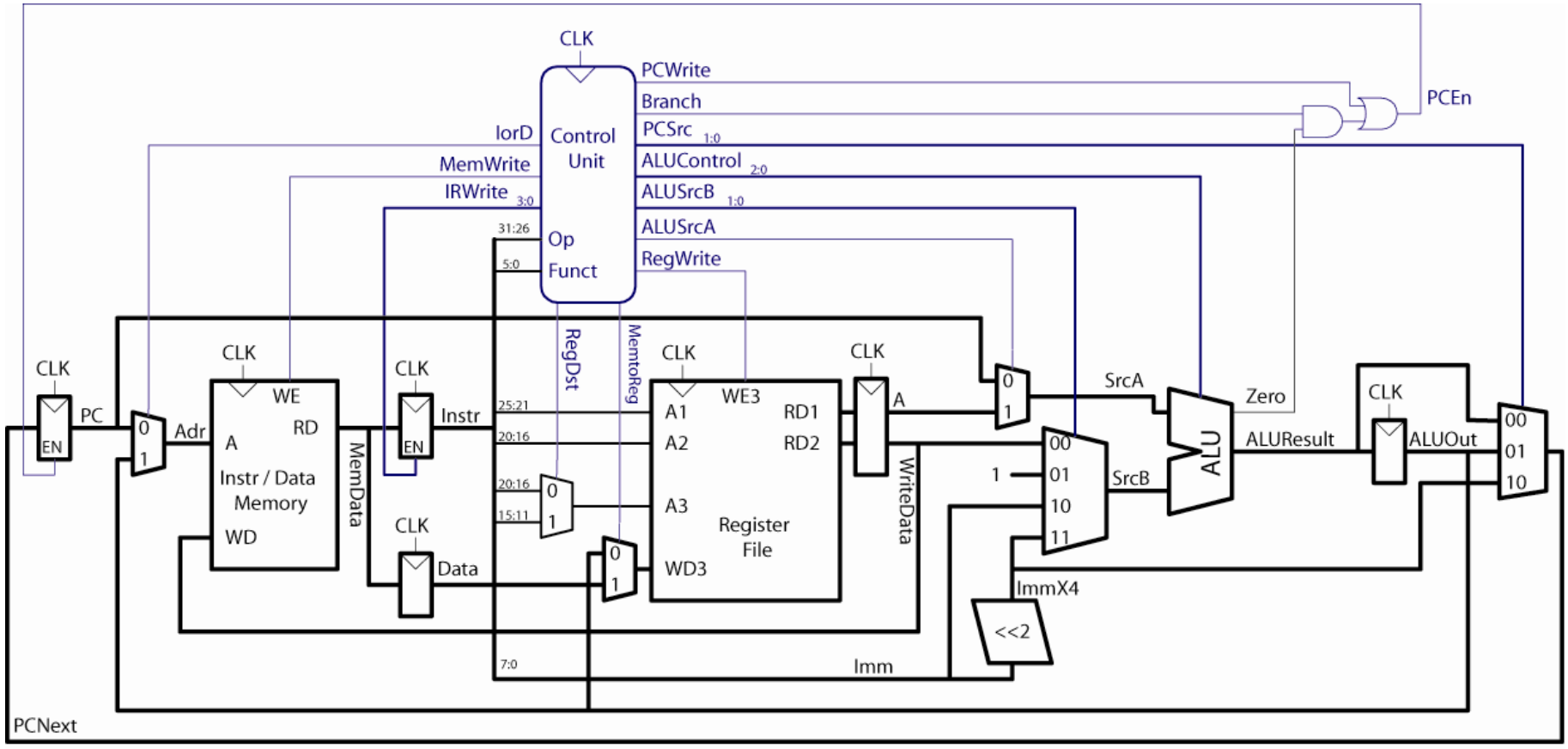


Design Partitioning

- **Architecture**: User's perspective, what does it do?
  - Instruction set, registers
  - MIPS, x86, Alpha, PIC, ARM, …
- **Microarchitecture**
  - Single cycle, multcycle, pipelined, superscalar?
- **Logic**: how are functional blocks constructed
  - Ripple carry, carry lookahead, carry select adders
- **Circuit**: how are transistors used
  - Complementary CMOS, pass transistors, domino
- **Physical**: chip layout
  - Datapaths, memories, random logic

MIPS Microarchitecture

- Multicycle μarchitecture ([Paterson04], [Harris07])



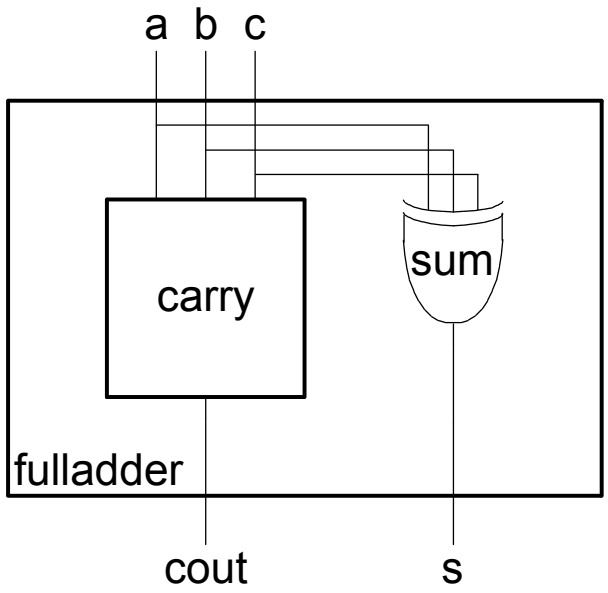
HDLs

- Hardware Description Languages
  - Widely used in logic design
  - Verilog and VHDL
- Describe hardware using code
  - Document logic functions
  - Simulate logic before building
  - Synthesize code into gates and layout
    - Requires a library of standard cells

Matches your design objectives to the costs of available library parts

Verilog Example

```
module fulladder(input  a, b, c,
                 output s, cout);

  sum      s1(a, b, c, s);
  carry    c1(a, b, c, cout);
endmodule
```




```
module carry(input  a, b, c,
             output cout)

  assign cout = (a&b) | (a&c) | (b&c);
endmodule
```

Circuit Design

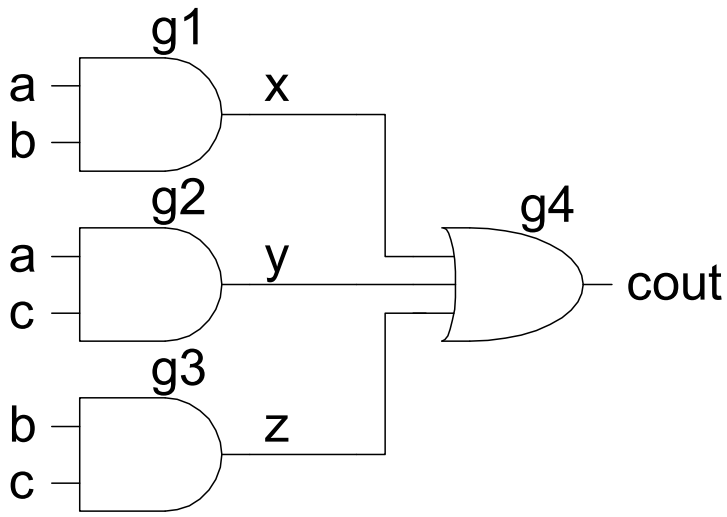
- How should logic be implemented?
  - NANDs and NORs vs. ANDs and ORs?
  - Fan-in and fan-out?
  - How wide should transistors be?
  - How many logic stages?
- We've seen how choices affect speed, area, power
- Logic synthesis makes these choices for you
  - Good enough for many applications
  - Hand-crafted circuits are still better

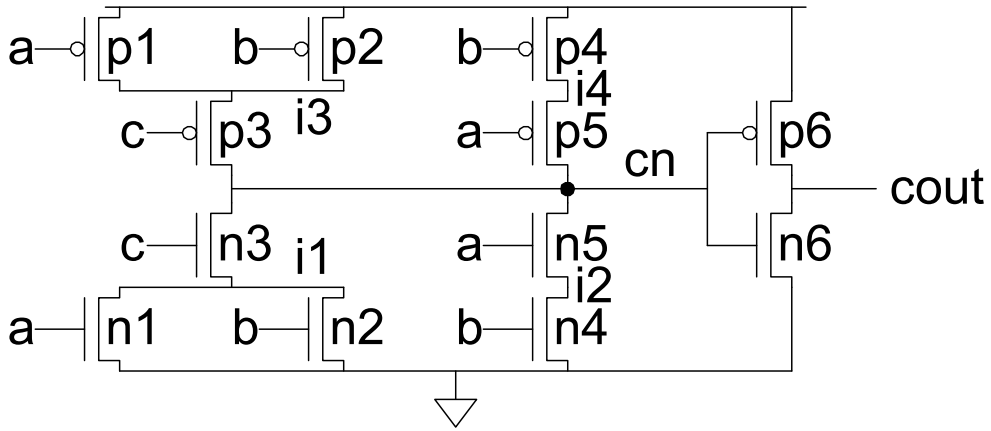
Example: Carry Logic

- `assign cout = (a&b) | (a&c) | (b&c);`

Example: Carry Logic

- `assign cout = (a&b) | (a&c) | (b&c);`





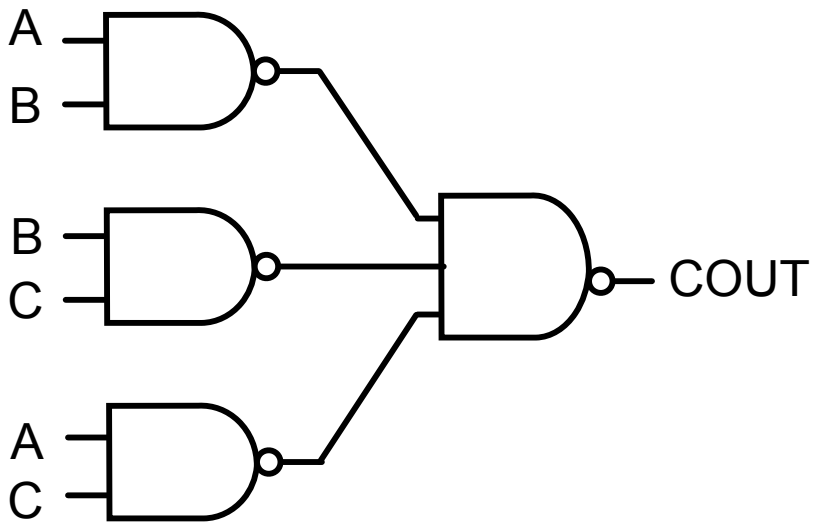
Which is better… by transistor count? delay?
Which one looks like hand-crafted logic?
What would Design Compiler choose?

Example: Hand-Optimize Carry Logic

- `assign cout = (a&b) | (a&c) | (b&c);`

Example: Hand-Optimize Carry Logic

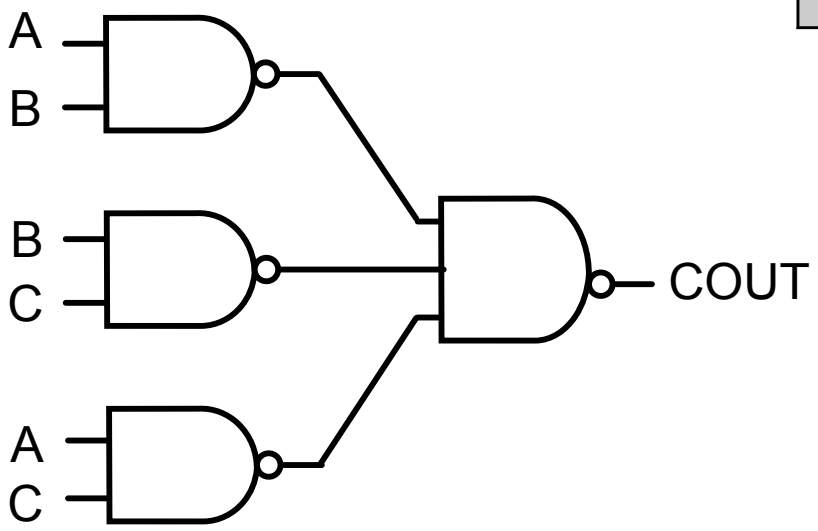
- `assign cout = (a&b) | (a&c) | (b&c);`



Example: Hand-Optimize Carry Logic

- `assign cout = (a&b) | (a&c) | (b&c);`

cell	qty	area
NAND2_X1	3	0.7980
NAND3_X1	1	1.0640
total	4	**3.4580**



Example: Repeat with Design Compiler

- `assign cout = (a&b) | (a&c) | (b&c);`

Exampl ompiler

- assign

Untitled — Edited

```
dholcomb@vlsicad2:~/lab3_dc_demo$more carry.v

module carry( a, b, c, cout);
input a,b,c;
output cout;
assign cout=(a&b)|(a&c)|(b&c);
endmodule
```

```
dholcomb@vlsicad2:~/lab3_dc_demo$dc_shell -f run_synth.tcl
```

```
dholcomb@vlsicad2:~/lab3_dc_demo$more carry_final.v

module carrytest ( a, b, c, cout );
  input a, b, c;
  output cout;
  wire   n4, n5, n6;

  OAI21_X1 U5 ( .B1(n4), .B2(n5), .A(n6), .ZN(cout) );
  OAI21_X1 U6 ( .B1(a), .B2(b), .A(c), .ZN(n6) );
  INV_X1 U7 ( .A(b), .ZN(n5) );
  INV_X1 U8 ( .A(a), .ZN(n4) );
endmodule
```

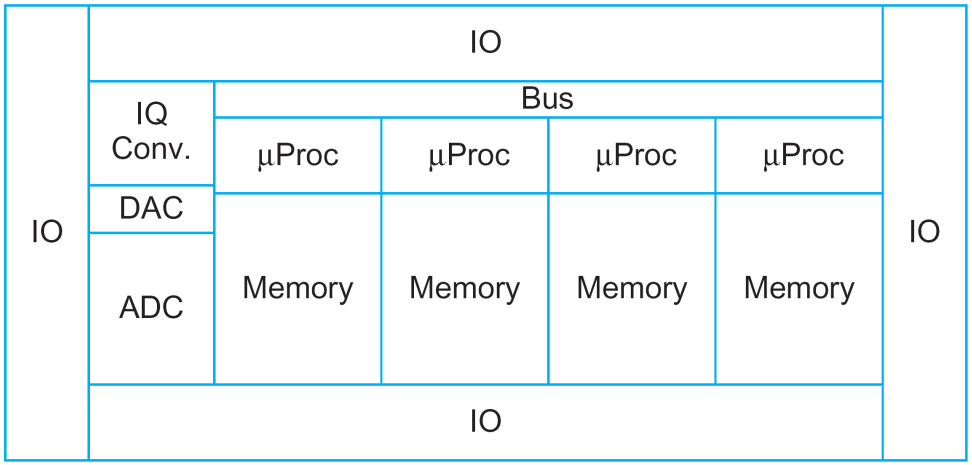
```
dholcomb@vlsicad2:~/lab3_dc_demo$more cell_report_final.rpt

****************************************
Report : cell
Design : carry
Version: E-2010.12-SP5-2
****************************************

Cell          Reference       Library              Area  Attributes
--------------------------------------------------------------------------------
U5            OAI21_X1        NangateOpenCellLibrary
                                                   1.0640
U6            OAI21_X1        NangateOpenCellLibrary
                                                   1.0640
U7            INV_X1          NangateOpenCellLibrary
                                                   0.5320
U8            INV_X1          NangateOpenCellLibrary
                                                   0.5320
--------------------------------------------------------------------------------
Total 4 cells                                      3.1920
```

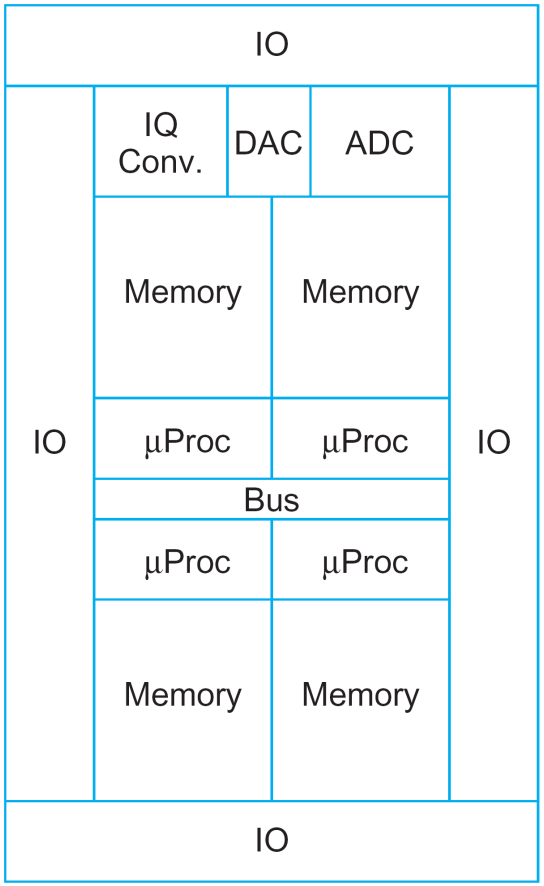
Physical Design

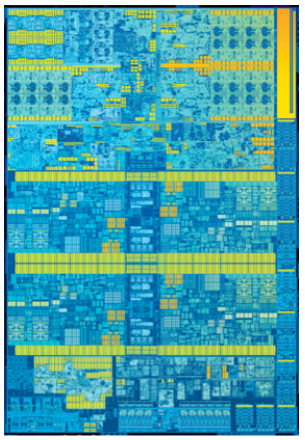
- Floorplan
- Standard cells
  - Place & route
- Datapaths
  - Slice planning
- Area estimation

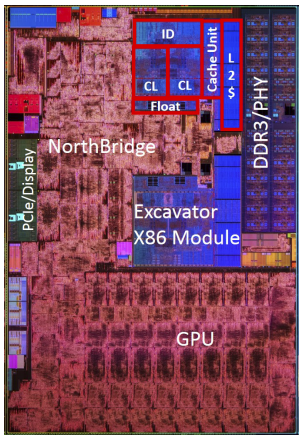
Floorplanning

- Global decisions about how blocks will be arranged
  - Communication
  - Avoid hotspots
  - Isolate sensitive circuits
  - Pick row height for standard cells









Area Estimation

- Need area estimates to make floorplan
  - Compare to another block you already designed
  - Or estimate from transistor counts
  - Budget room for large wiring tracks
  - Your mileage may vary

Table 1.10 Typical layout densities

Element	Area
random logic (2-level metal process)	1000 – 1500 λ^2 / transistor
datapath	250 – 750 λ^2 / transistor or 6 WL + 360 λ^2 / transistor
SRAM	1000 λ^2 / bit
DRAM (in a DRAM process)	100 λ^2 / bit
ROM	100 λ^2 / bit

Discuss

- Why is datapath more dense than random logic?
- Estimate size (in μm^2) of SRAM bit in 45nm tech

Table 1.10 Typical layout densities

Element	Area
random logic (2-level metal process)	1000 – 1500 λ^2 / transistor
datapath	250 – 750 λ^2 / transistor or 6 WL + 360 λ^2 / transistor
SRAM	1000 λ^2 / bit
DRAM (in a DRAM process)	100 λ^2 / bit
ROM	100 λ^2 / bit

Discuss

- Why is datapath more dense than random logic?
- Estimate size (in μm²) of SRAM bit in 45nm tech

Mistry et al. Intel (on Moodle)

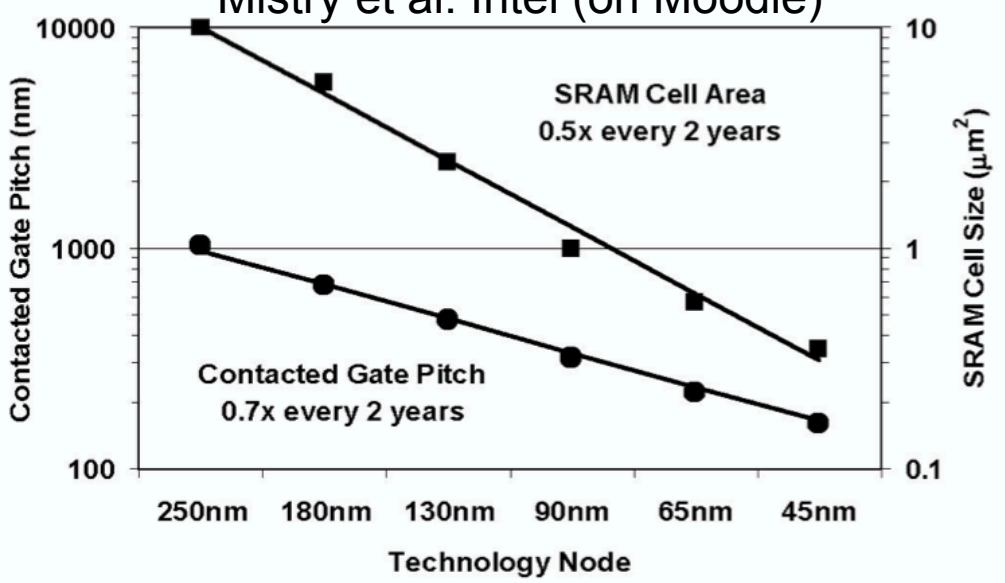


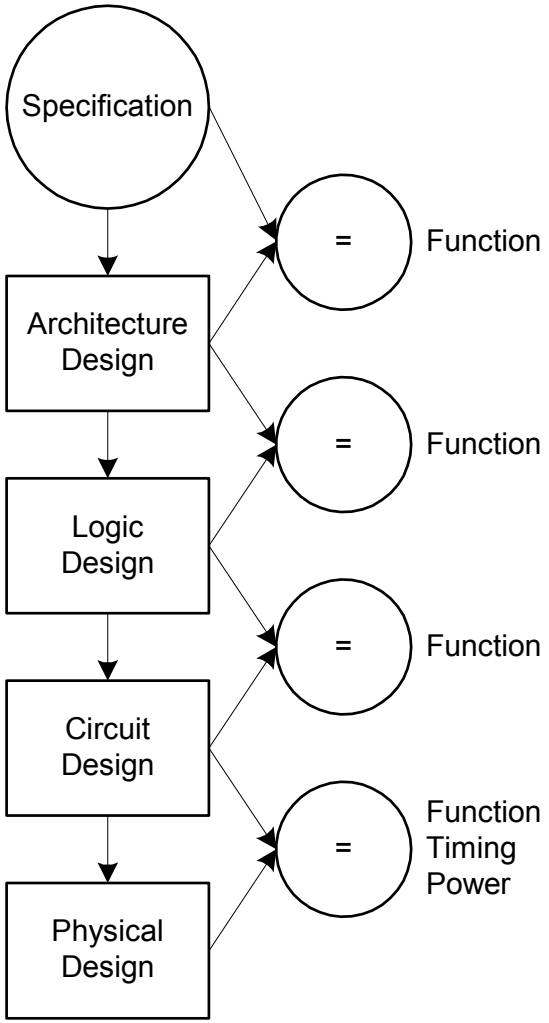
Fig. 3 Intel contacted gate pitch and SRAM cell scaling trend

Table 1.10 Typical layout den

Element	
random logic (2-level metal process)	1000 – 1500 λ^2 / transistor
datapath	250 – 750 λ^2 / transistor or 6 WL + 360 λ^2 / transistor
SRAM	1000 λ^2 / bit
DRAM (in a DRAM process)	100 λ^2 / bit
ROM	100 λ^2 / bit

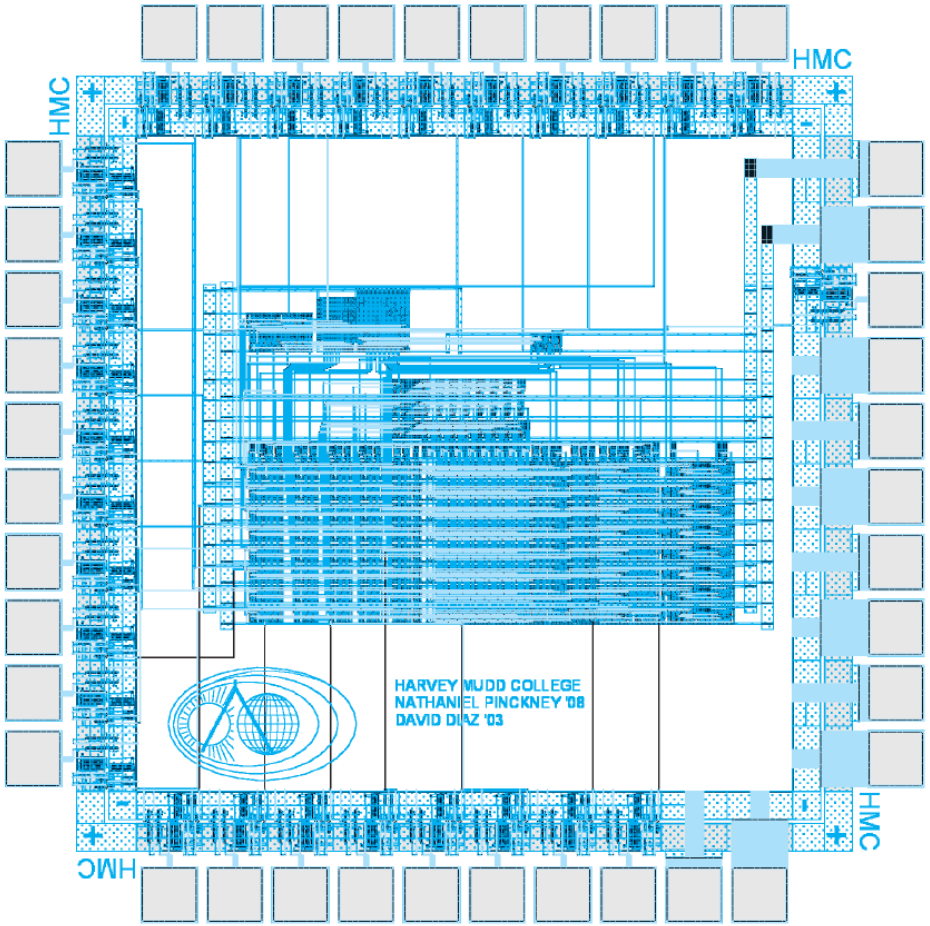
Design Verification

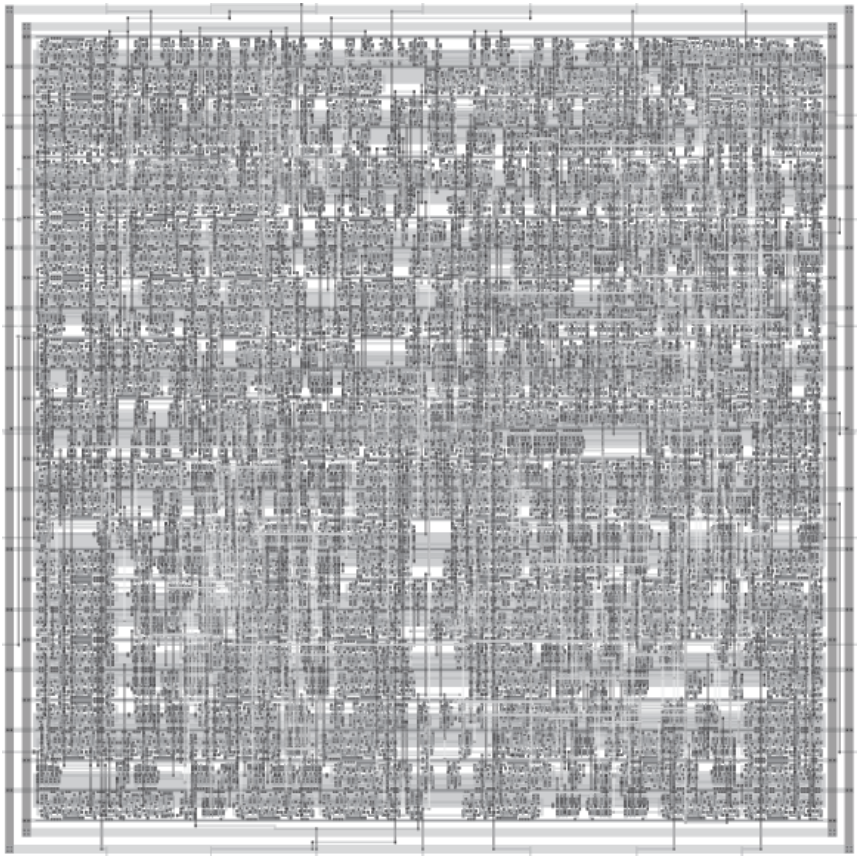
- Fabrication is slow & expensive
  - MOSIS 0.6μm: $1000, 3 months
  - 65 nm: $3M, 1 month
- Debugging chips is very hard
  - Limited visibility into operation
- Prove design is right before building!
  - Logic simulation
  - Ckt. simulation / formal verification
  - Layout vs. schematic comparison
  - Design & electrical rule checks
- Verification is > 50% of effort on most chips!



Custom vs. Synthesis

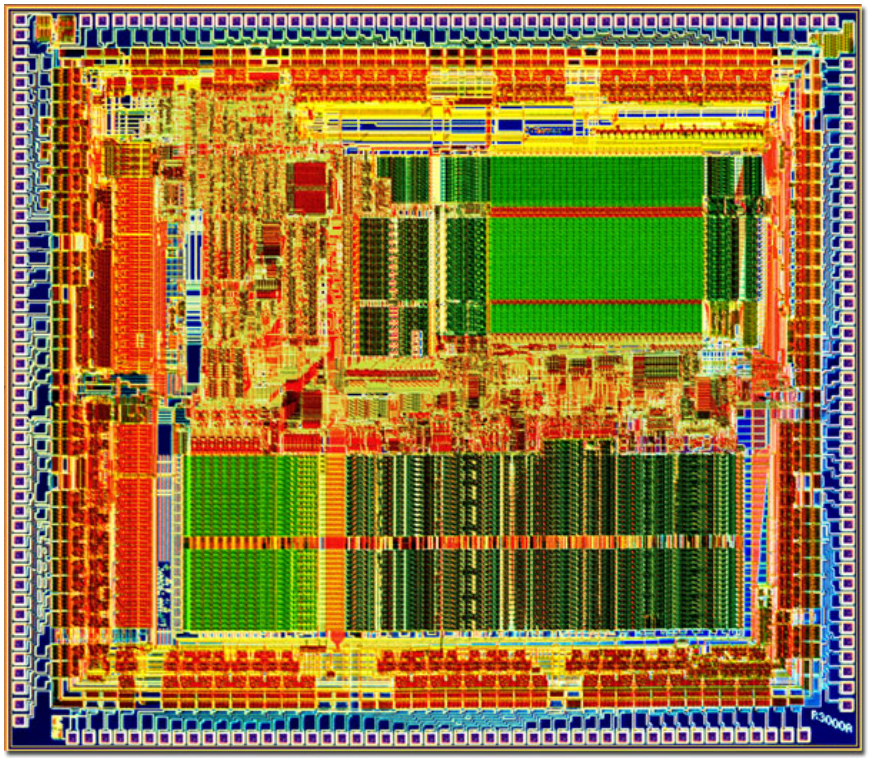
- Custom beats synthesis in this case
- Will your find the same in Lab 3?





MIPS R3000 Processor

- 32-bit 2nd generation commercial processor (1988)
- Led by John Hennessy (Stanford, MIPS Founder)
- 32-64 KB Caches
- 1.2 μm process
- 111K Transistors
- Up to 12-40 MHz
- 66 mm^2 die
- 145 I/O Pins
- V_{DD} = 5 V
- 4 Watts
- SGI Workstations

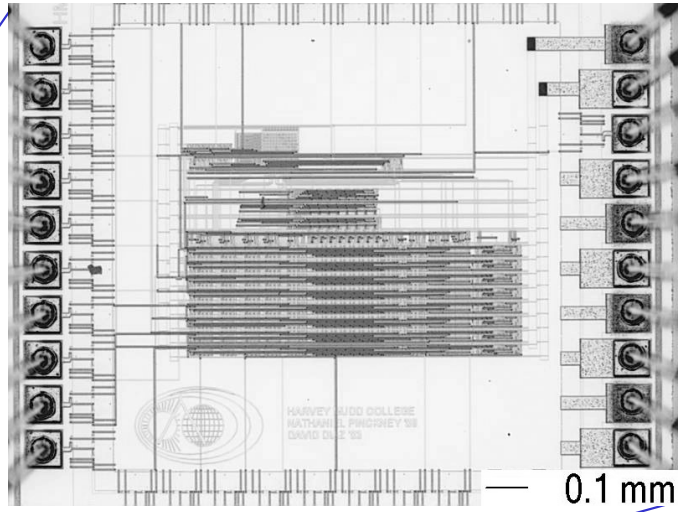


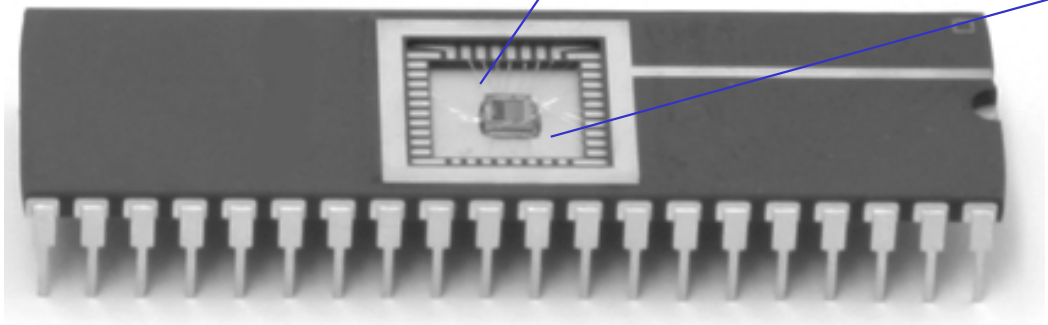
http://gecko54000.free.fr/?documentations=1988_MIPS_R3000

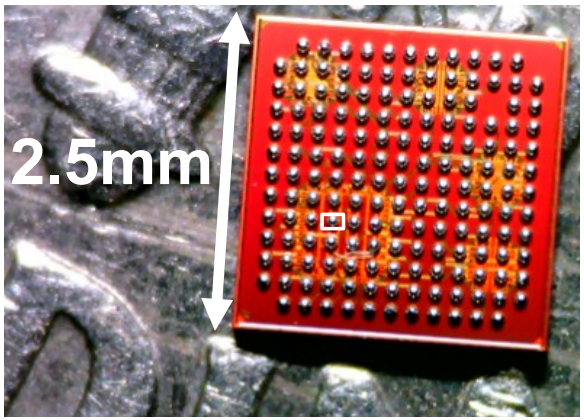
Where are the caches on layout? Where are I/Os?

Fabrication & Packaging

- Tapeout final layout
- Fabrication
  - 6, 8, 12” wafers
  - Optimized for throughput, not latency (10 weeks!)
  - Cut into individual dice
- Packaging
  - Bond gold wires from die I/O pads to package

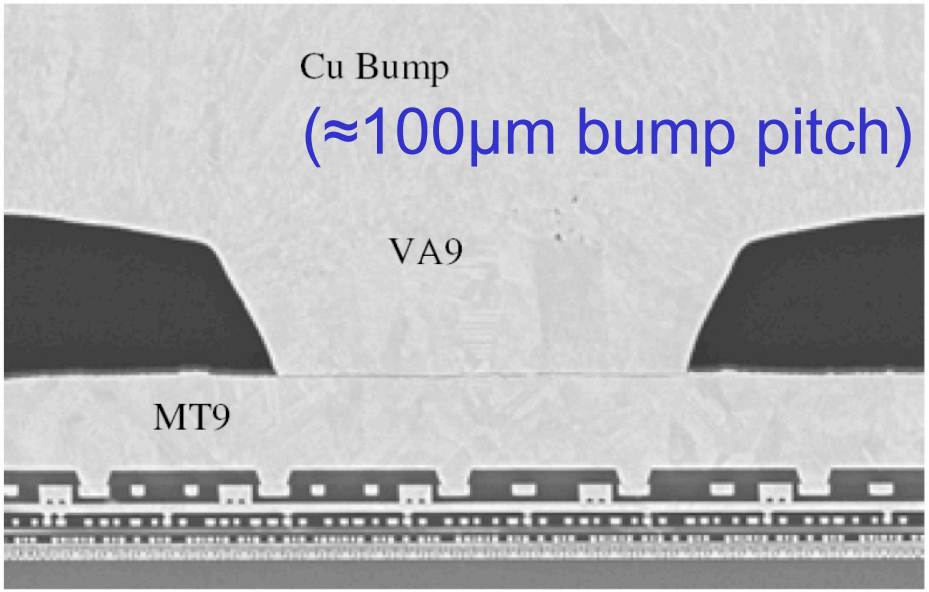


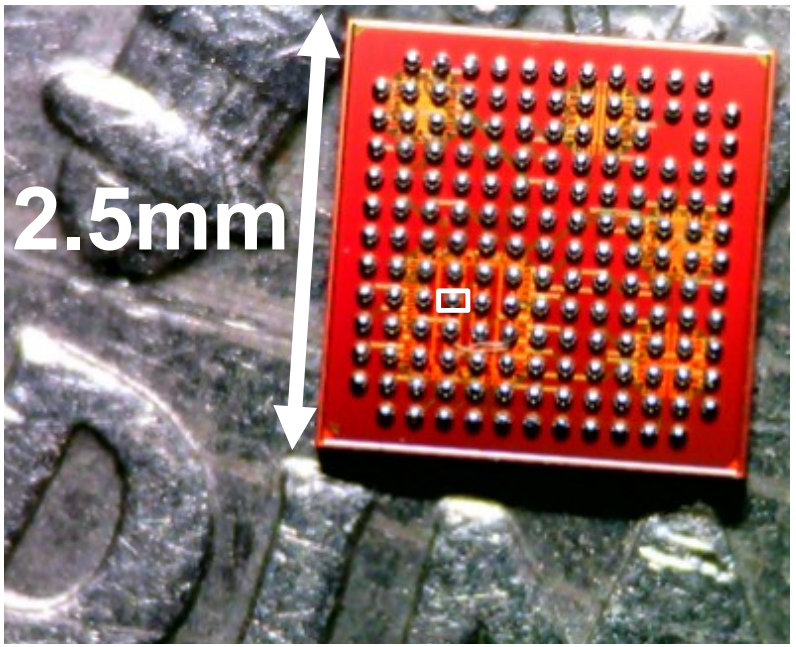




Are transistors toward or away from PCB?

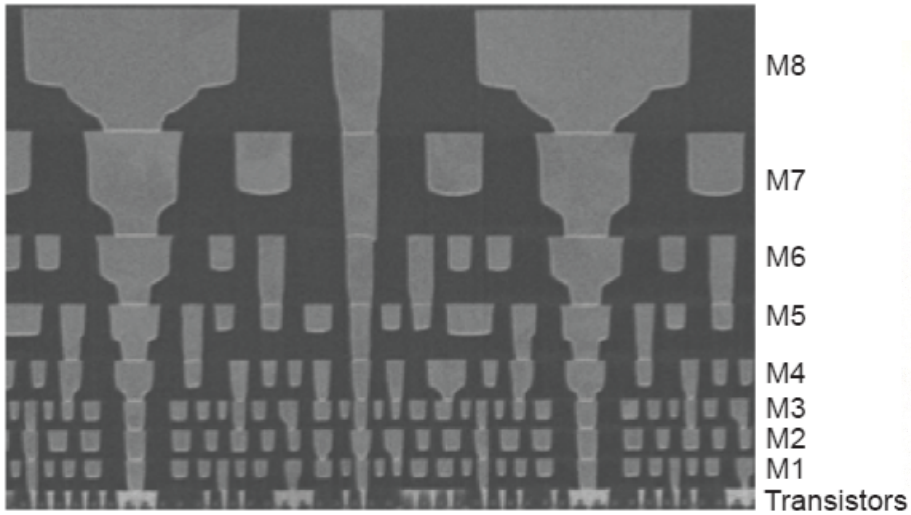
(From Module 7)





1 μm

Intel 45 nm metal stack



Layer	*t* (nm)	*w* (nm)	*s* (nm)	pitch (nm)
M9	7 μm	17.5 μm	13 μm	30.5 μm
M8	720	400	410	810
M7	504	280	280	560
M6	324	180	180	360
M5	252	140	140	280
M4	216	120	120	240
M3	144	80	80	160
M2	144	80	80	160
M1	144	80	80	160

Testing

- Test that chip operates
  - Design errors
  - Manufacturing errors
- A single dust particle or wafer defect kills a die
  - Yields from 90% to < 10%
  - Depends on die size, maturity of process
  - Test each part before shipping to customer

Example:

- Manufacturing process has $p = 0.1/mm^2$ failure rate
- Chip has area of n mm^2
- What fraction of chips will work?