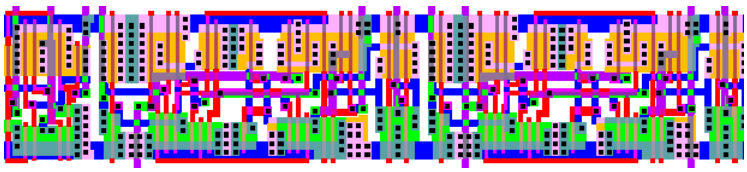
CS/EE 5710/6710
Advanced IC Design I

Digital VLSI Design



CS/EE 5710/6710

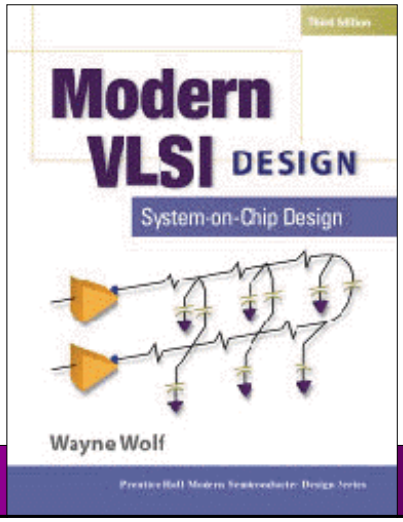
- Advanced IC Design I
  - Digital VLSI Design
  - T Th 12:25-1:45, EMCB 104
- Instructor: Dr. Erik Brunvand
  - MEB 4160
  - Office hours: After class, or by appointment
- TA: Vamshi Kadaru
- TA : Sudheesh Madayi
  - Office hours: to be determined

CS/EE 5710/6710

- Web Page - all sorts of information!
- Http://www.cs.utah.edu/classes/cs5710
- Contact:
  - cs5710@cs.utah.edu
    - Goes to everyone in the class
    - You need to sign up - see the web page
  - teach-cs5710@cs.utah.edu
    - Goes to instructor and TAs

Textbook

- Modern VLSI Design, 3rd edition
  - Prentice-Hall, copyright 2002



Class Goal

- To learn about modern digital CMOS IC design
  - Class project - teams will build moderate sized chip
    - We'll form teams in a few weeks
  - Modulo funding constraints, these chips can be fabricated through MOSIS
    - NSF funded chip fabrication service
- We'll use tools from Cadence and Synopsys
  - These only run on Solaris, so you'll need a CADE account
    - I also assume you know something about UNIX

Prerequisites

- Digital design is required! (CS/EE 3700)
  - Boolean algebra
  - Combinational circuit design and optimization
    - K-map minimization, SOP, POS, DeMorgan, bubble-pushing, etc.
    - Arithmetic circuits, 2's complement numbers
  - Sequential Circuit design and optimization
    - Latch/flip-flop design
    - Finite state machine design/implementation
    - Communicating FSMs
    - Using FSMs to control datapaths

Recommendations

- Computer Architecture experience is helpful
  - Instruction set architecture (ISA)
  - Assembly language execution model
  - Instruction encoding
  - Simple pipelining
- I assume you've used some sort of CAD tools for digital circuits
  - Schematic capture
  - Simulation

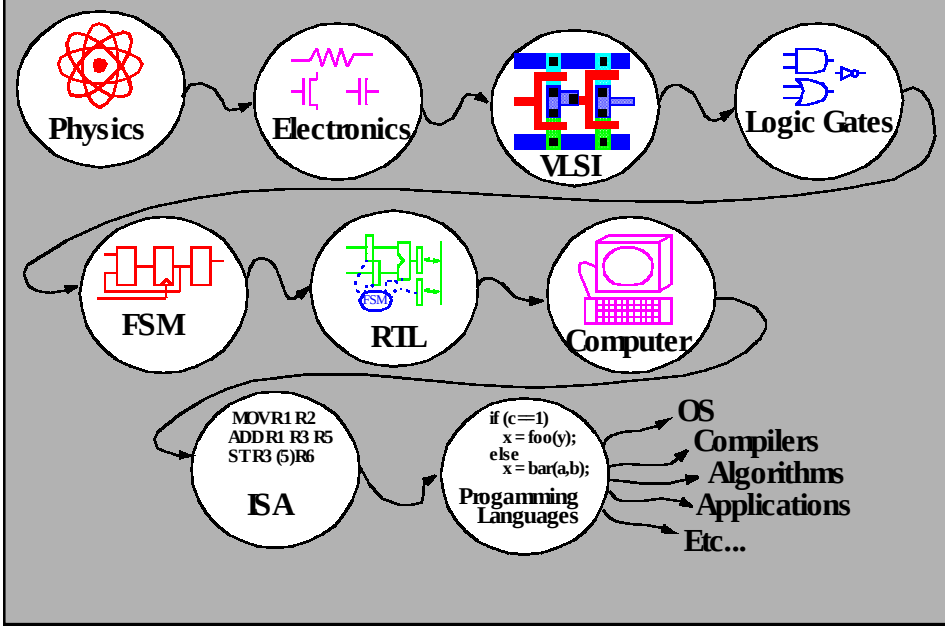
First Assignment

- CAD Assignment #1
  - Cadence Composer tutorial
  - Simple circuit design with simulation
    - Learn basic Verilog for testbench
  - Available on the web site
  - Due on Friday, August 30th, 5:00pm

Assignments/Grading

- Written HW and quizzes (5%)
- CAD assignments (cell designs) (40%)
- Design review (5%)
- Mid-term exam (10%)
- Final Project (40%)
  - See the syllabus (web page) for more details about grading breakdown

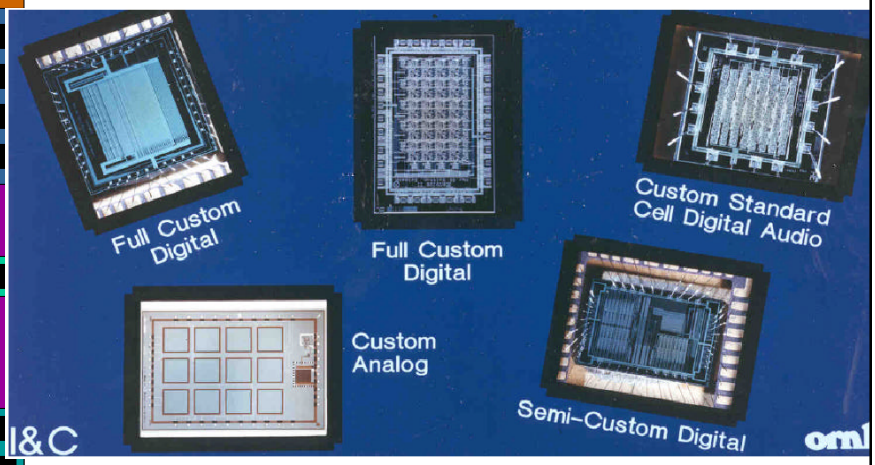
The Big Picture



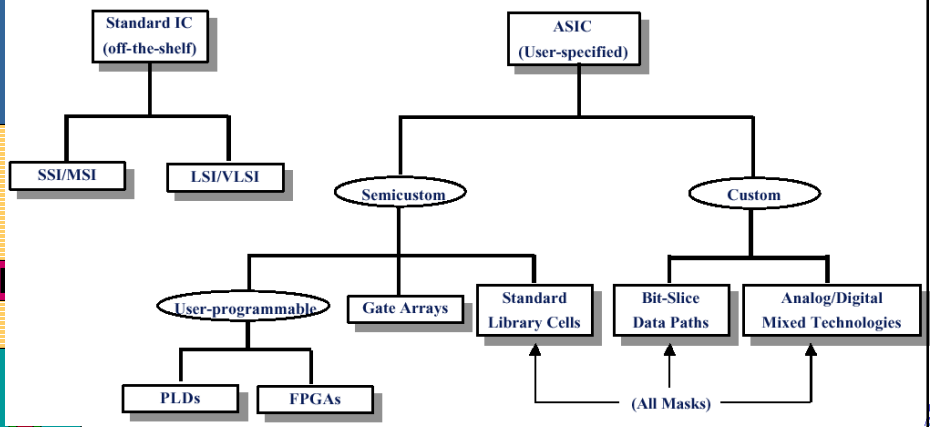
Why Use VLSI?

- Why use electronics?
  - Electrons are easy to move and control
    - Much easier to move/control electrons than real stuff
  - Move information, not things
    - Cheaper, less energy
- Why use VLSI?
  - Very efficient way to build electronic systems
  - Pattern multiple devices on a shared substrate
  - Cost isn't dependent on number of devices
    - Based more on area of the chip

A Variety of VLSI



A Variety of VLSI

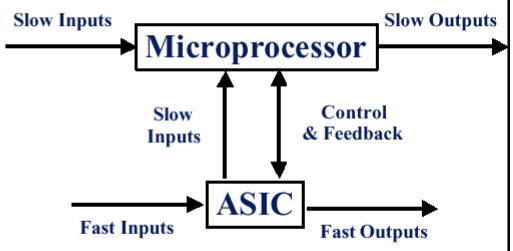


Cost of VLSI

- Per-gate cost decreases as density goes up
  - **Microprocessors and other off-the-shelf LSI/VLSI chips are the *most* cost-effective because thousands of gates are available in a single chip. ($0.001/gate; 2000 gates/pin)**
  - **SSI/MSI glue logic chips are the *least* cost-effective because only a few gates are available in a single chip. ($0.01/gate; 1-3 gates/pin)**

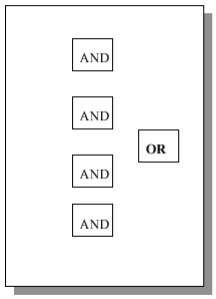
System Functionality Split

- The most economical means of implementing logic functions is to use a microprocessor.
- When the microprocessor is too slow or too busy to handle some fast inputs and outputs, an ASIC can be used to implement "random" logic.



PLDs

➢Best for simple designs with lots of I/O

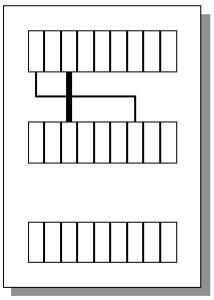


- Vendor prefabricates multiple sets of ANDs and ORs with programmable connections
- User specifies connections to implement desired logic functions
- Replaces 300 to 8000 gates with single package of 20-84 pins
- Electrically programmable (and erasable) by the user one at a time within minutes
- PC-based development system costs $3-5K

Mask Gate Arrays

- Best for moderate-sized designs with time-to-market constraints
  - First Sparc chip was a gate array…

Gate Arrays

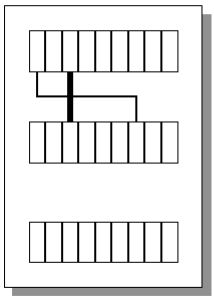


- Vendor prefabricates rows of gates and stockpiles wafers
- User specifies two layers to implement logic functions
- Replaces 10,000 to 10,000,000 gates (or more)
- After place & route, masks are made for *two* layers
- Workstation-based development system costs $ 50K
- Turnaround time for prototypes is 3-5 weeks

FPGAs

- Best for low-quantity or fast-changing apps

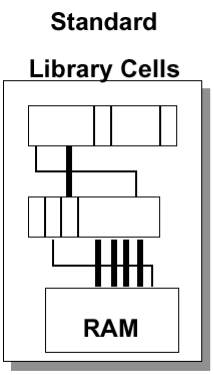
FPGAs



- Vendor prefabricates parts with rows of gates and programmable connections
- User specifies connections to implement logic functions
- Replaces 2,000 to 2,000,000 gates (or more)
- Electrically programmable (some are erasable) by the user one at a time within hours
- Production quantity < 20,000
- PC-based development system costs $5-10K

Standard Cells

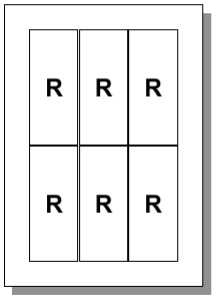
➢ Best for high-quality apps with extra functionality (mult, CPU, RAM, etc.)



- Vendor develops library disk files of significant functions
- User selects cells and specifies two layers of interconnections
- After place & route, masks are made for *all* layers
- Replaces 100,000 to 10,000,000 gates (or more)
- Workstation-based development system costs $ 100K
- Turnaround time for prototypes is 8 weeks

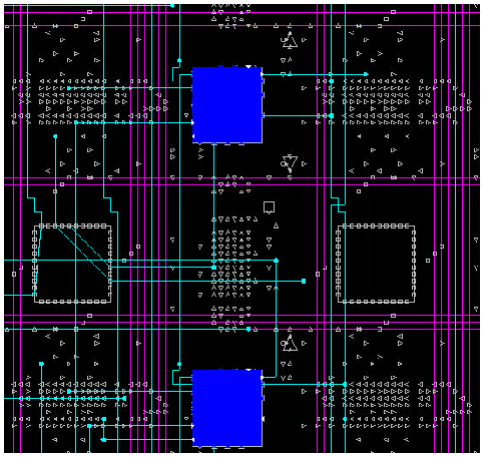
Bit-Slice Datapaths

➢ Best for special-purpose data processing



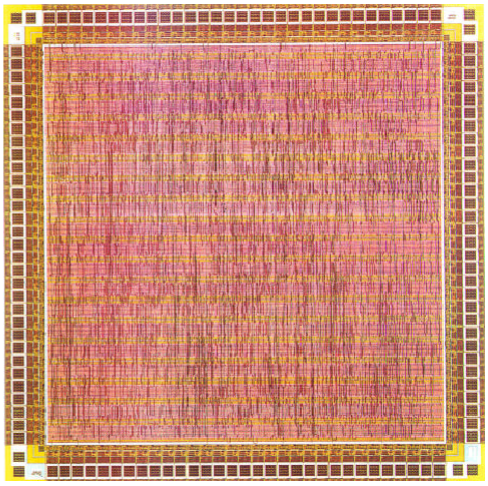
- User develops bit-slices to be replicated
- Most efficient use of silicon
- Masks are made for all layers
- Replaces 100,000 to 10,000,000 gates (or more)
- Workstation-based development system costs $ 150K
- Turnaround time for prototypes is 8 weeks

Example: Xilinx FPGA



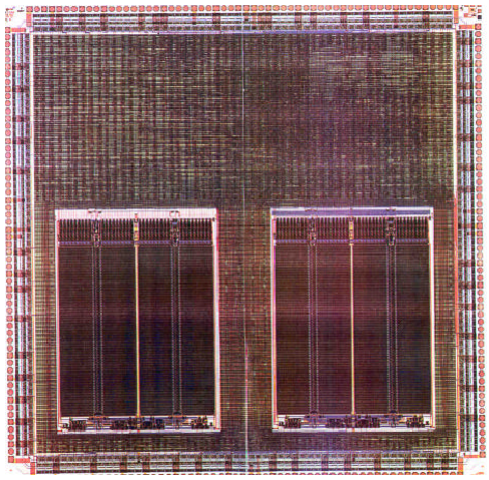
Example: Gate Array

- The fabricator provides basic gates with space for interconnect.
- The application designer submits a logic net-list which defines the interconnect layers.



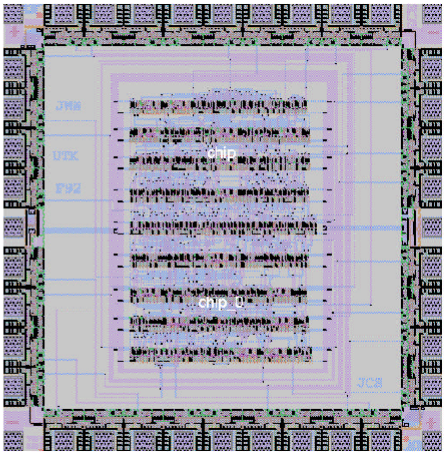
Example: Gate Array

- May contain embedded RAM/ROM



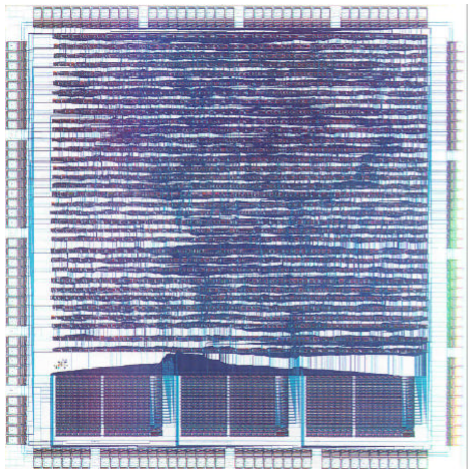
Example: Standard Cells

- **Only logic cells which are needed are fabricated.**
- **Higher performance and less area can be achieved but fabrication takes longer.**

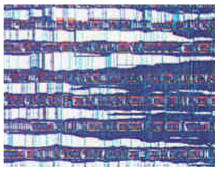


Example: Standard Cells

- May also contain RAM/ROM or other embedded macrocells

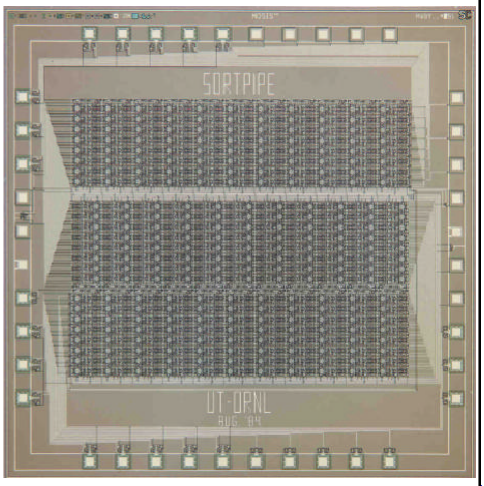


Space between rows for wiring can be varied as needed.



Example: Bit-Sliced Datapath

- **The basic cell has been designed with its neighbors in mind.**
- **One bit has been arrayred 8 times to form a byte.**
- **Each byte is connected to its neighbor to form a datapath or systolic array.**



Some Recent Chips

- Intel Pentium 4
  - 42M transistors
  - 217 mm^2 die
  - 0.18-micron process
  - 2GHz clock
- Intel/HP Itanium 646
  - 220M transistors
  - 465 mm^2 die
  - 0.18-micron process
  - 1.2GHz clock
- Sparc III
  - 90M transistors
  - 0.13-micron, 7-level metal copper process
  - 64b, 1MB L2$
  - 53W @ 1.3V, 1.1GHz
- Sony Playstation II
  - 128-bit CPU "Emotion Engine"
    - 0.18 micron process
    - 300MHz, 6.2 GFLOPS, 3.2 Gbytes/second
      - 10 floating point multiply-accumulators and 4 floating point dividers
      - 3x floating point performance of 500 MHz PIII
  - Graphic synthesizer cgip
    - 0.25 micron chip
    - 42.7M transistors
    - 16.8x16.8 mm^2 die
    - 2560-bit datapath
    - 48 Gbytes/sec memory bandwidth
    - 75M polygons/sec, 2.4 Gpixels/sec

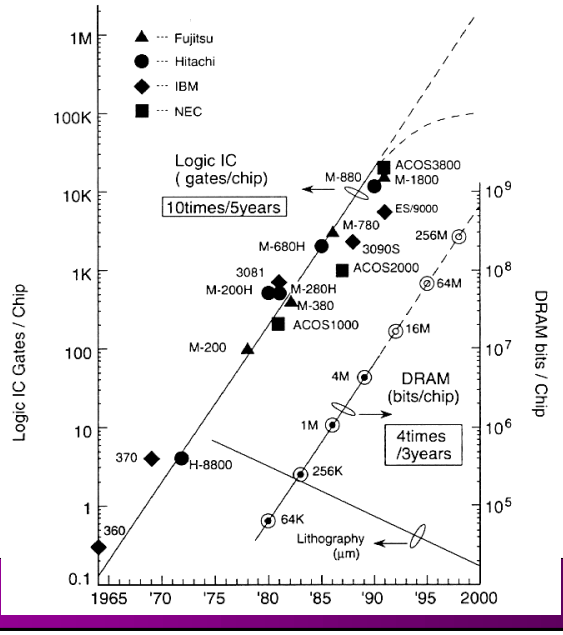
Good News

- Moore's law says we're improving at an exponential rate

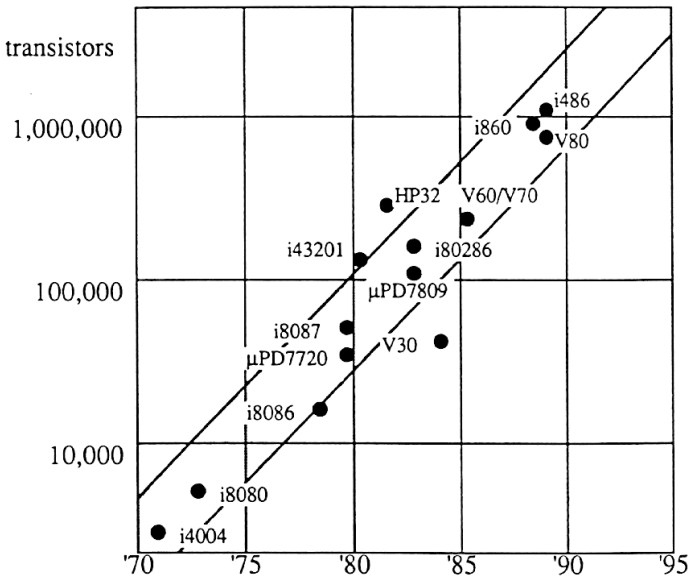
	Capacity	Speed (latency)
Logic	2x in 3 years	2x in 3 years
DRAM	4x in 3 years	2x in 10 years
Disk	4x in 3 years	2x in 10 years

- Since fab cost was growing at a relatively modest rate, this implied that cost/function was dropping exponentially
- Cost for each gate dropping by 1/2 every 3 years

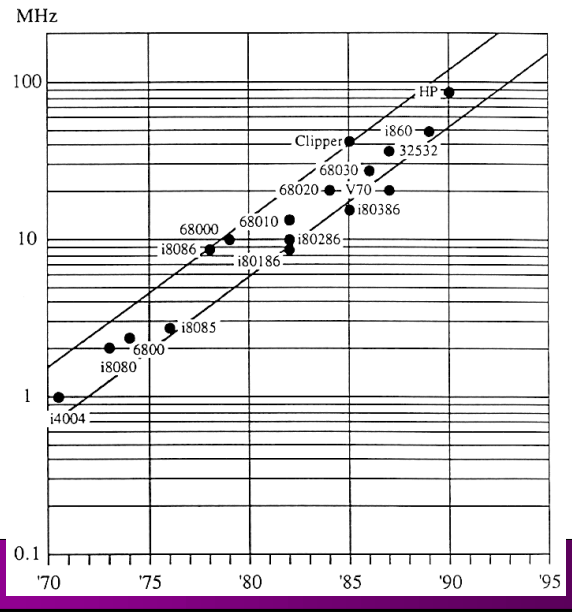
Evolution in Complexity



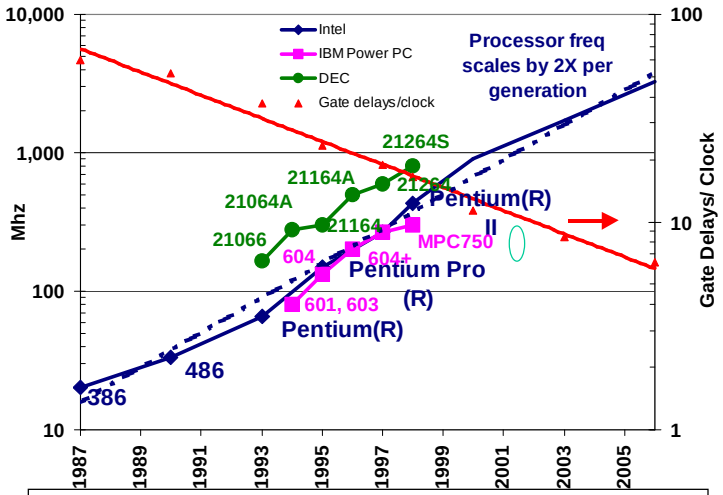
Evolution in Transistors/Chip



Evolution in Speed/Performance



Processor Frequency Trend



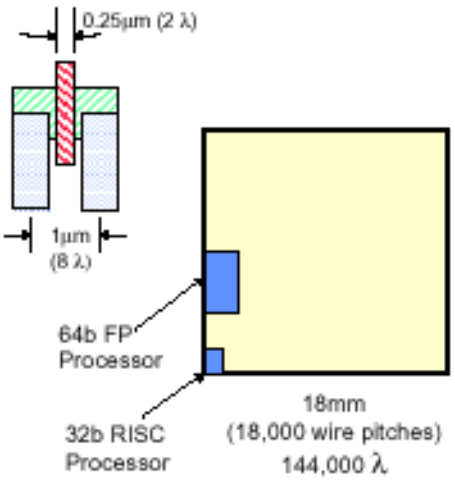
❶ Frequency doubles each generation
❷ Number of gates/clock reduce by 25%

Bad News

- Although fab cost wasn't huge, design cost went through the roof
  - Cost of designing chips is growing exponentially with the complexity of the circuits
  - Now fab costs are starting to catch up…

What Fits on a Chip Today?

- Relatively modest current process
  - 18mm on a side
  - 0.25u gate
  - 1.0u wire pitch
  - 5-level metal
- 32b RISC, 8kx16k L
- SRAM 32x32 L/bit
  - 8kx16k L is 16KB
- DRAM
  - 8kx16k L is 128KB



A Glimpse Into the Future

Silicon in 2010

Die Area: 2.5x2.5 cm
Voltage: 0.6 - 0.9 V
Technology: 0.07 µm

	Density (Gbits/cm2)	Access Time (ns)
DRAM	8.5	10
DRAM (Logic)	2.5	10
	0.3	1.5

2.5 times

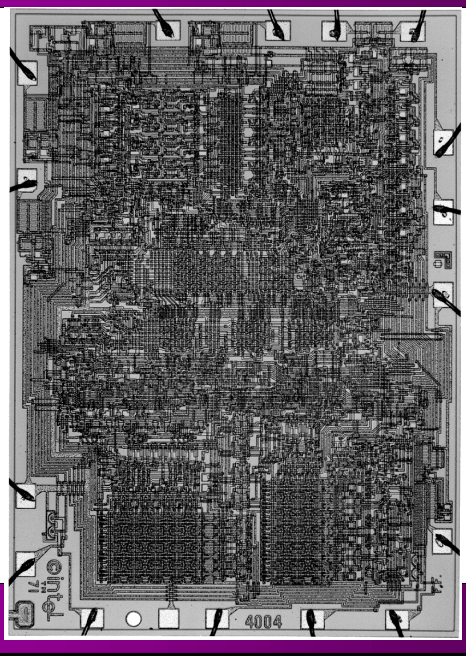
3 times clock rate

	Density (Mgates/cm2)	Max. Pow (W/cm2)	Clock Rate (GHz)
Custom	25	54	3
Std. Cell	10	27	1.5
Gate Array	5	18	1
Single-Mask GA	2.5	12.5	0.7
FPGA	0.4	4.5	0.25

What's on an Integrated Circuit

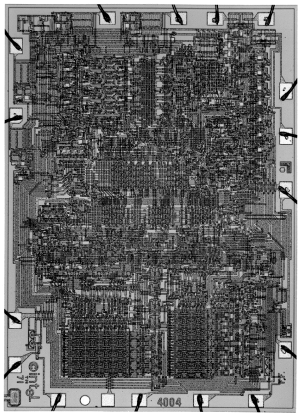
- Actually only two types of things
  - Conducting layers which form wires
    - Used to be only one metal wire layer
    - Now you get 4-6 metal layers
    - Requires insulators between the layers
  - Transistors
    - In digital systems these are used mostly as voltage controlled switches
    - Basically free in terms of overall chip space
    - The chip area is mostly filled with wires…

Intel 4004 Micro-Processor

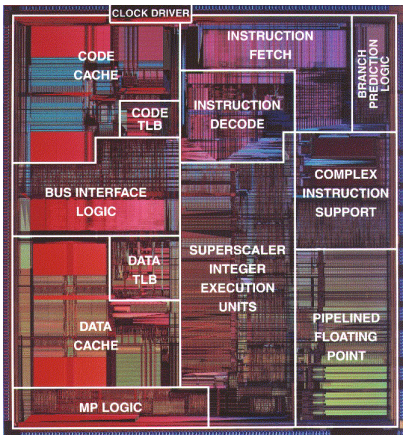


Intel Pentium (II) microprocessor

The Old and the New

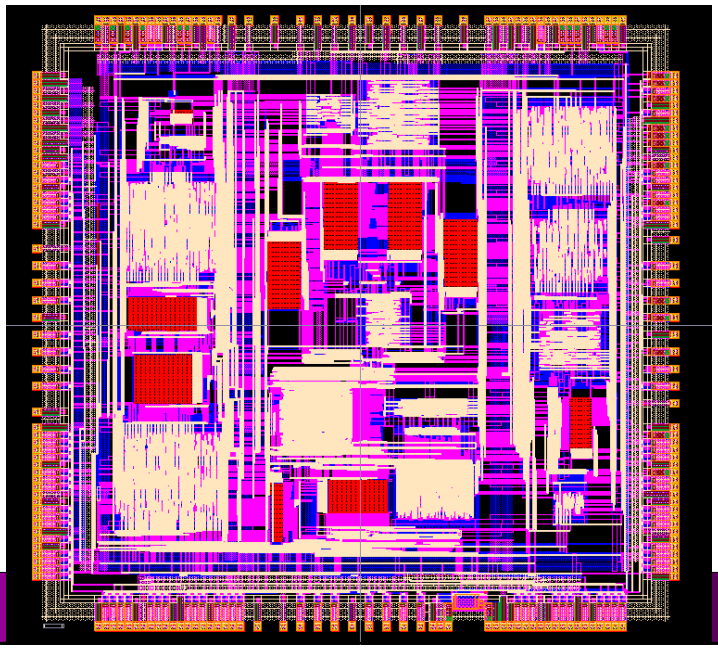


Intel 4004 Microprocessor



Intel Pentium Microprocessor

Example: Avalanche Fledgling



IC Technology

- We'll use the AMI 0.6u 3-level-metal CMOS process
  - We have technology files that define the process
    - MOSIS Scalable CMOS Rev. 8 (SCMOS)
  - Tech files from NCSU CDK
    - NCSU toolkit is designed for custom VLSI layout
    - Design Rule Check (DRC) rules
    - Layout vs. Schematic (LVS) rules

Class Project

- Standard Cell Library
  - Each group will design a small, but useful, standard cell library
    - Use HDL synthesis with this library as a target
    - Use Cadence Silicon Ensemble for place and route
- Custom Datapath
  - Use ICC router to connect HDL-Synthesized control to custom-designed datapath
  - It will be VERY helpful to have a mix of knowledge on your team

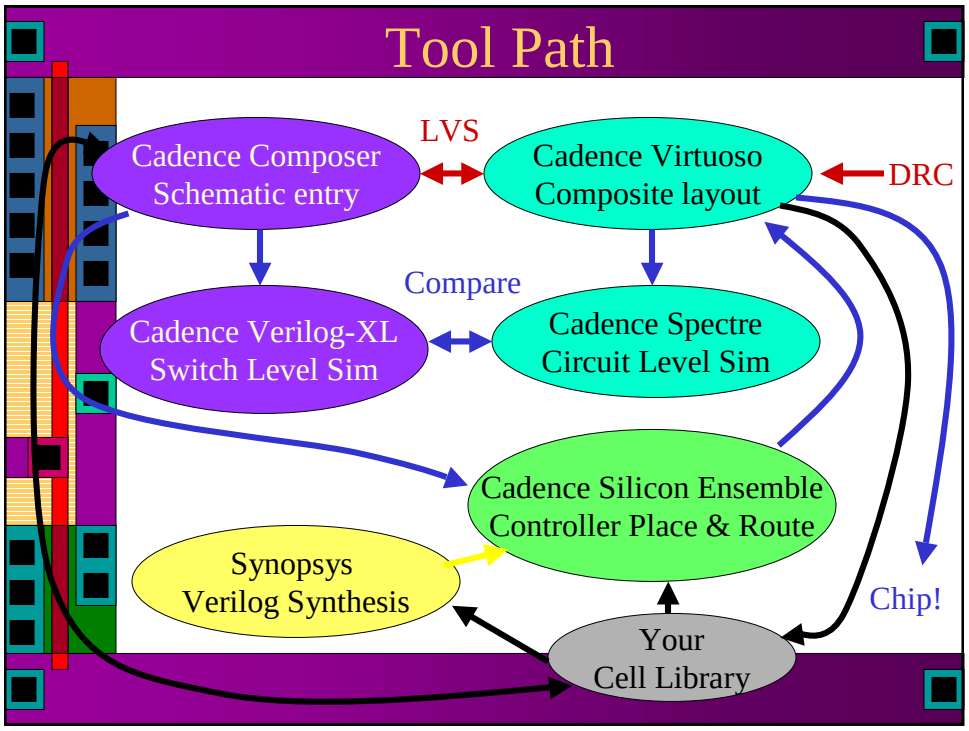
Class Project

- Two complete design views: Schematic and Layout
  - Complete design in Composer schematics, simulated with Verilog
  - Complete design at layout level in Virtuoso with detailed simulation using Spectre
  - Validate they are the same with LVS
- Custom layout for datapath
- Synthesized controller using Synopsys, Silicon Ensemble, and your cell library
- Final assembly back in Virtuoso

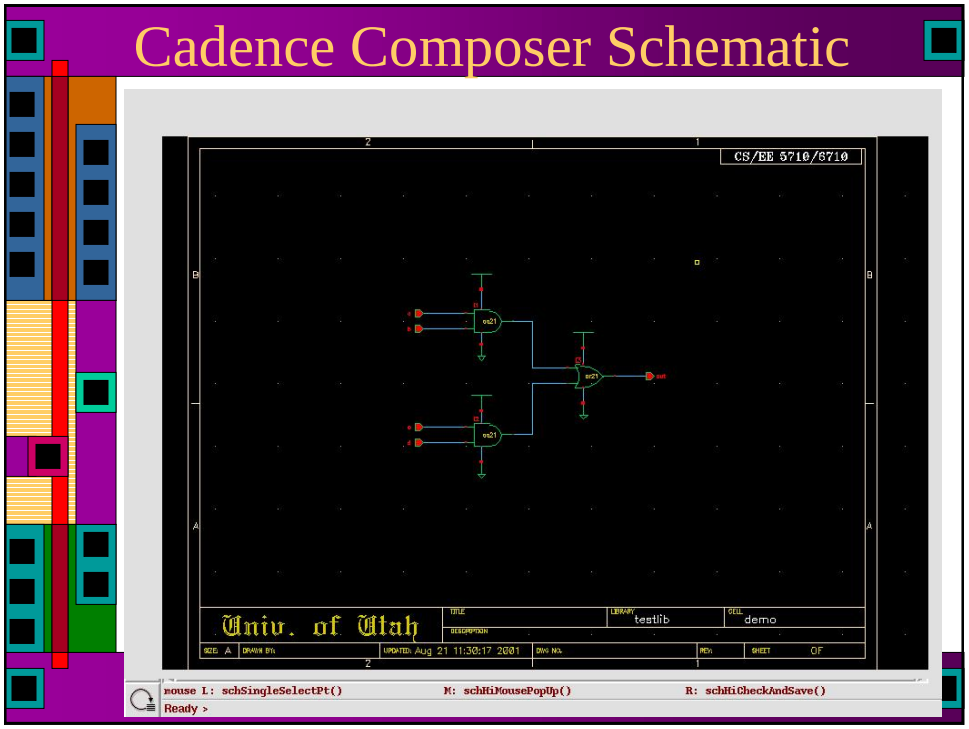
Timetable

- This project may be a race to the finish!
  - There is no slack in this schedule!!!
  - VLSI design always takes longer than you think
    - Even if you take that rule into account!
  - After you have 90% finished, there's only 90% left…
  - All team members will have to contribute!
    - Team peer evaluations twice a semester

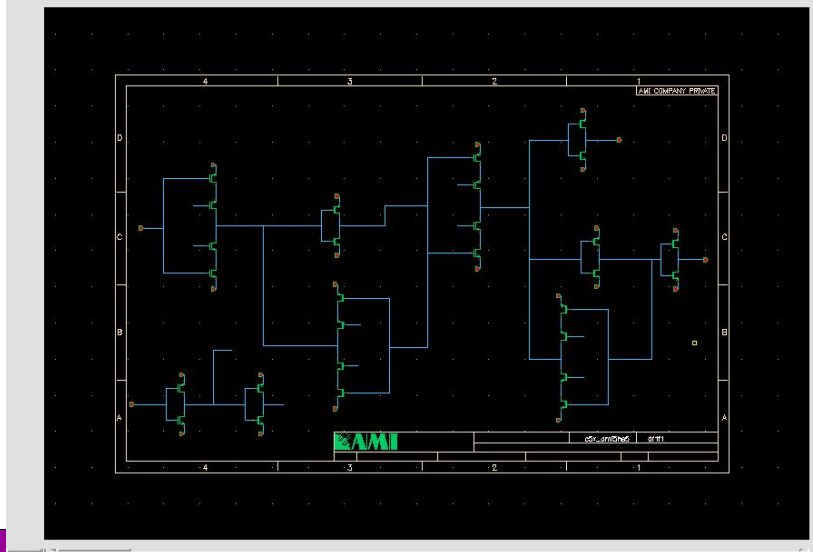
Tool Path



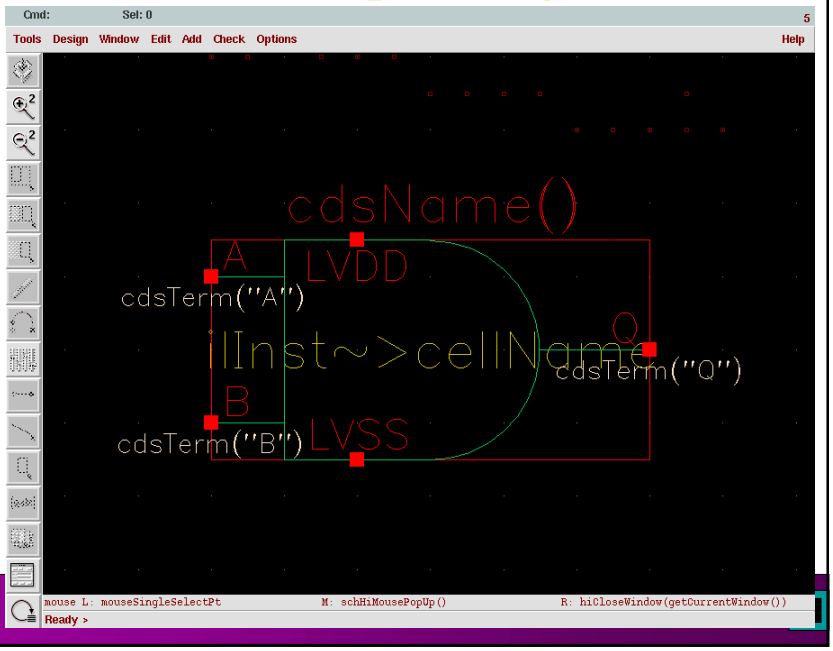
Cadence Composer Schematic



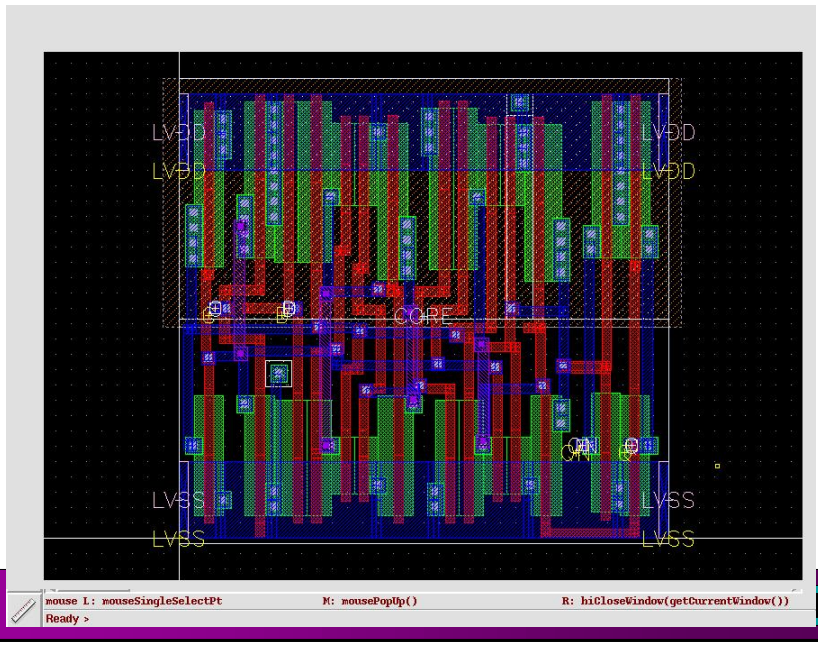
Cadence Composer Schematic



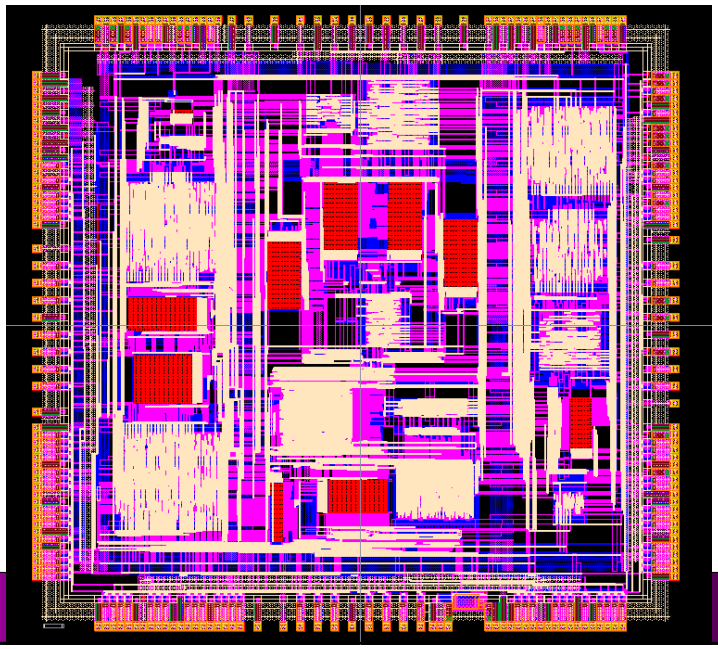
Cadence Composer Symbol



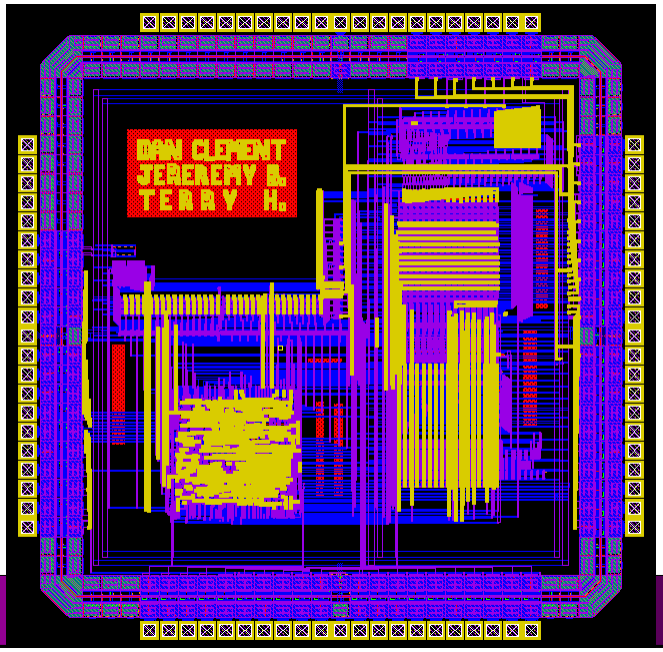
Cadence Virtuoso Layout



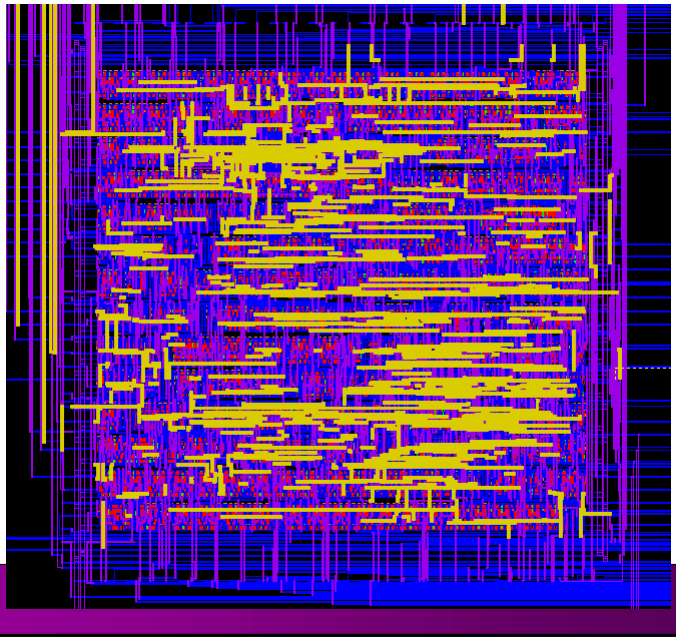
Example: Avalanche Fledgling



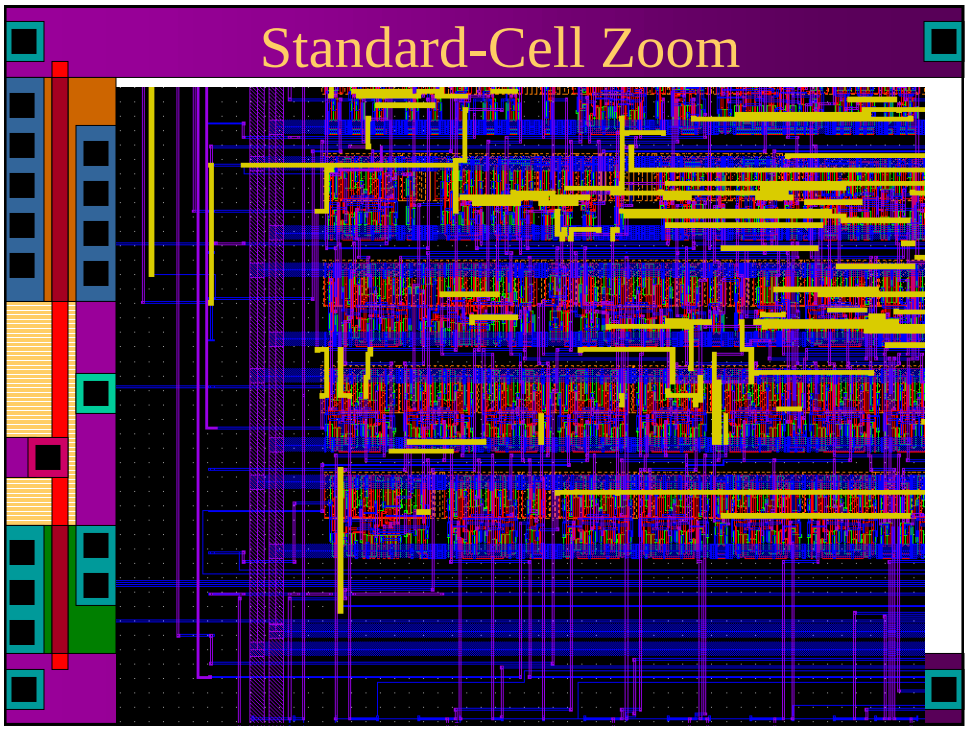
More Realistic: Last Year



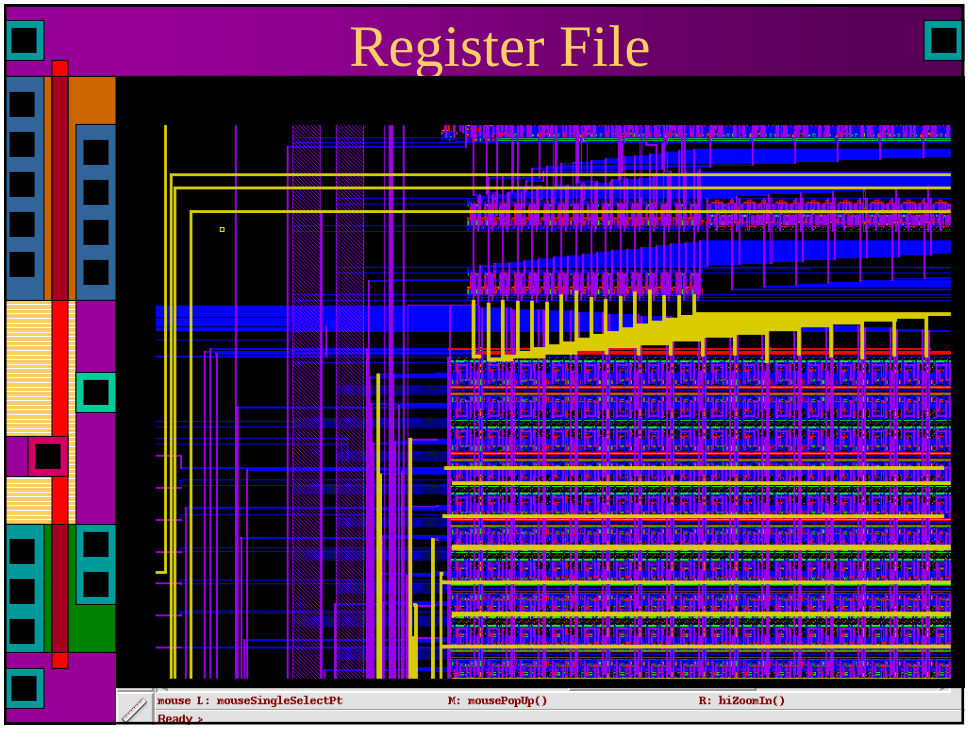
Standard-Cell Part



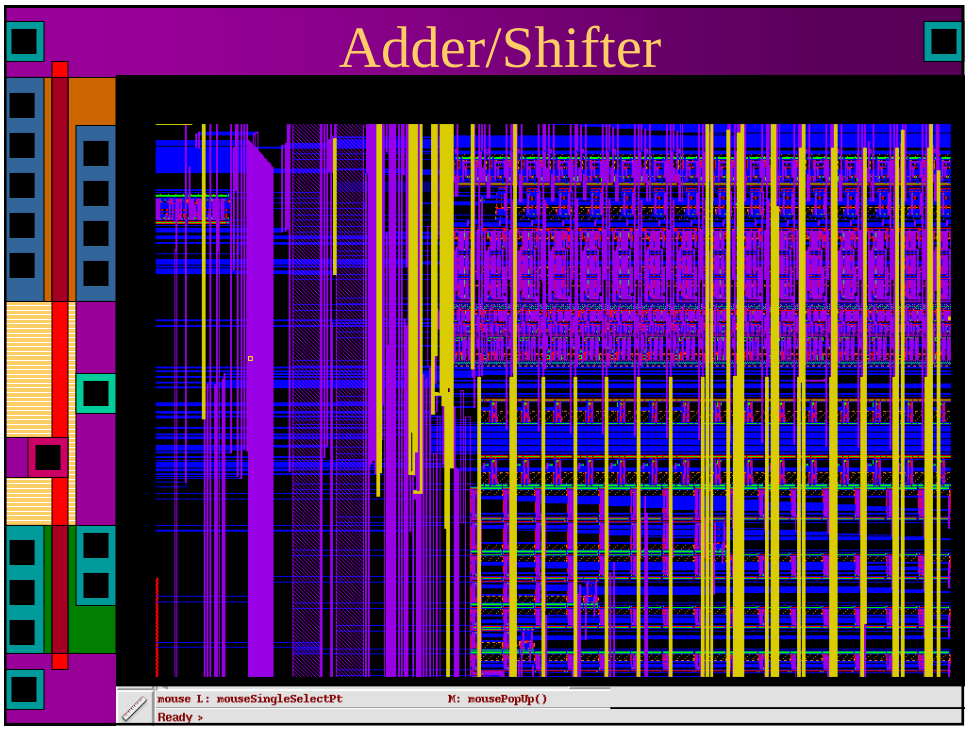
Standard-Cell Zoom



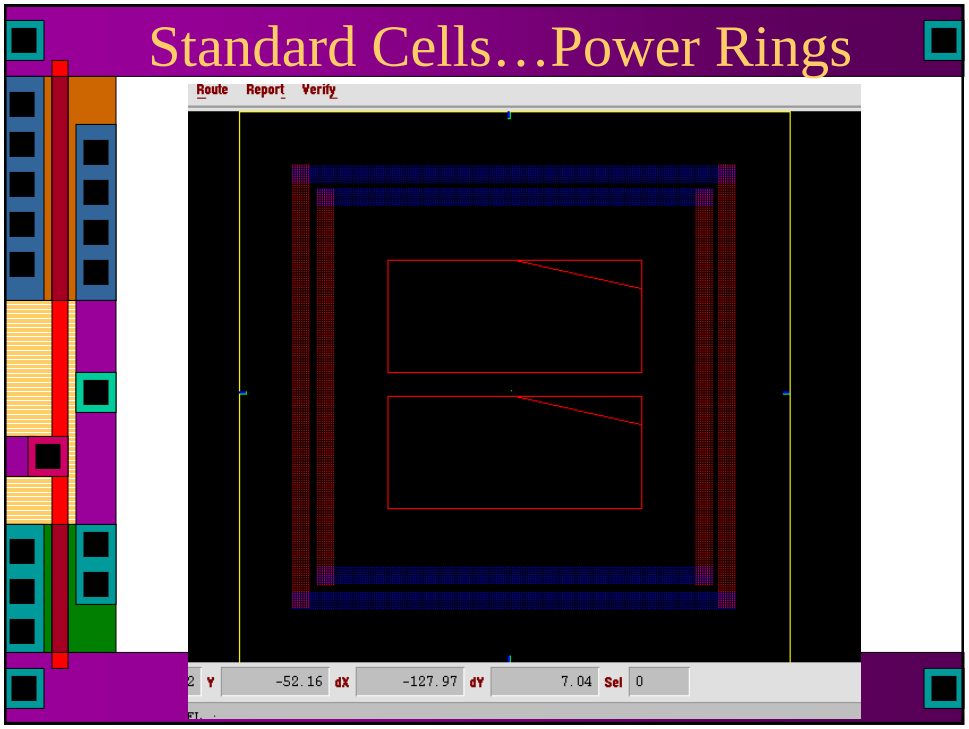
Register File



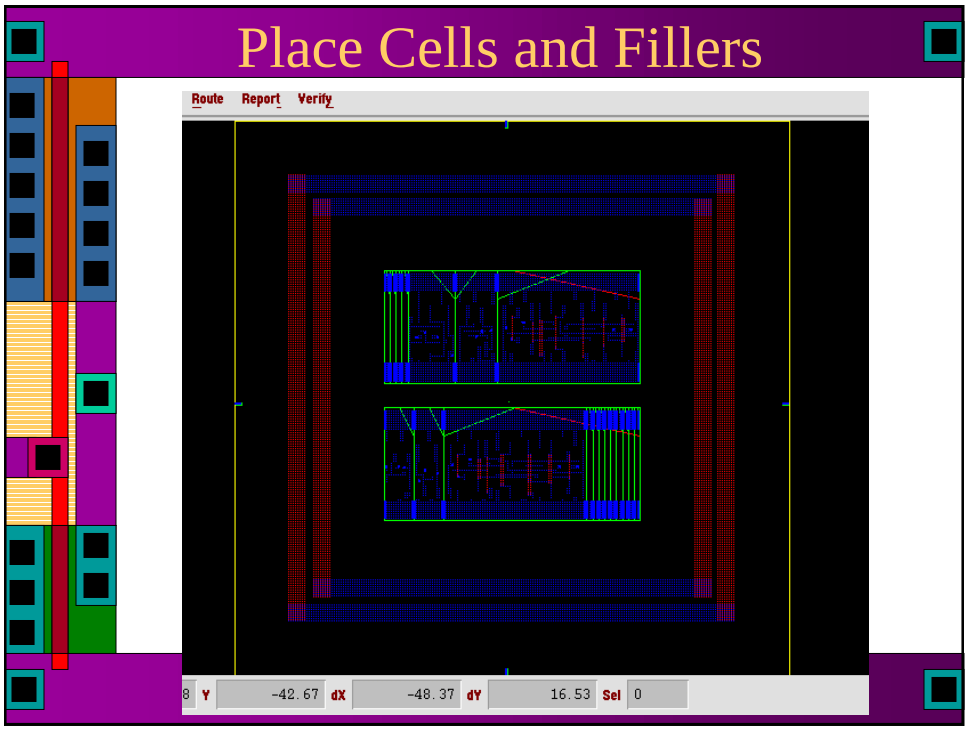
Adder/Shifter



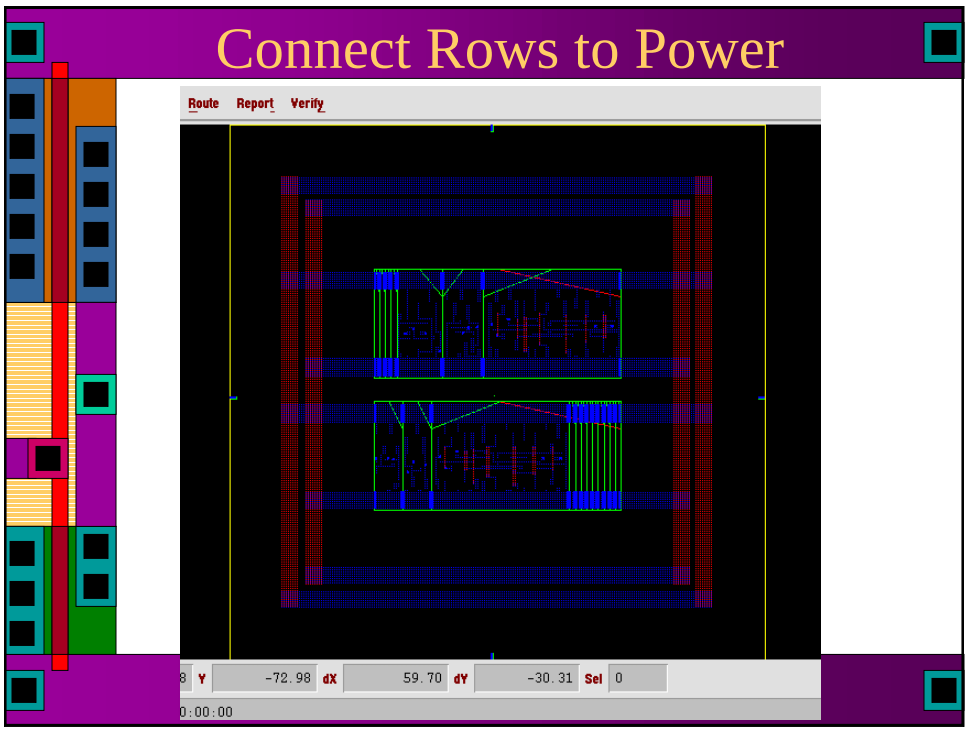
Standard Cells…Power Rings

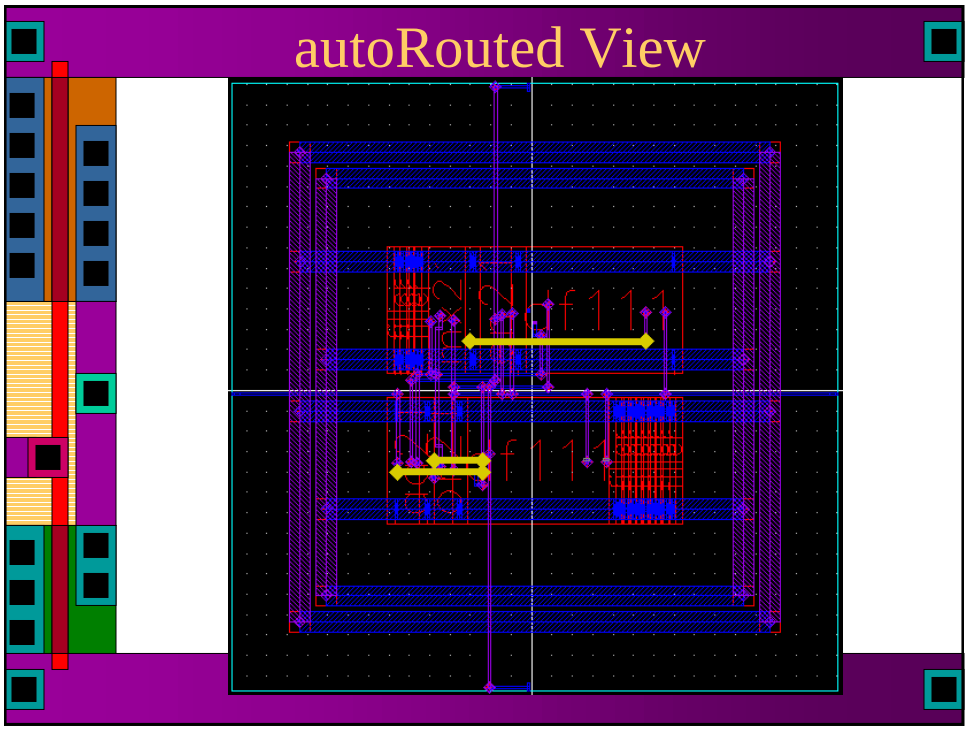


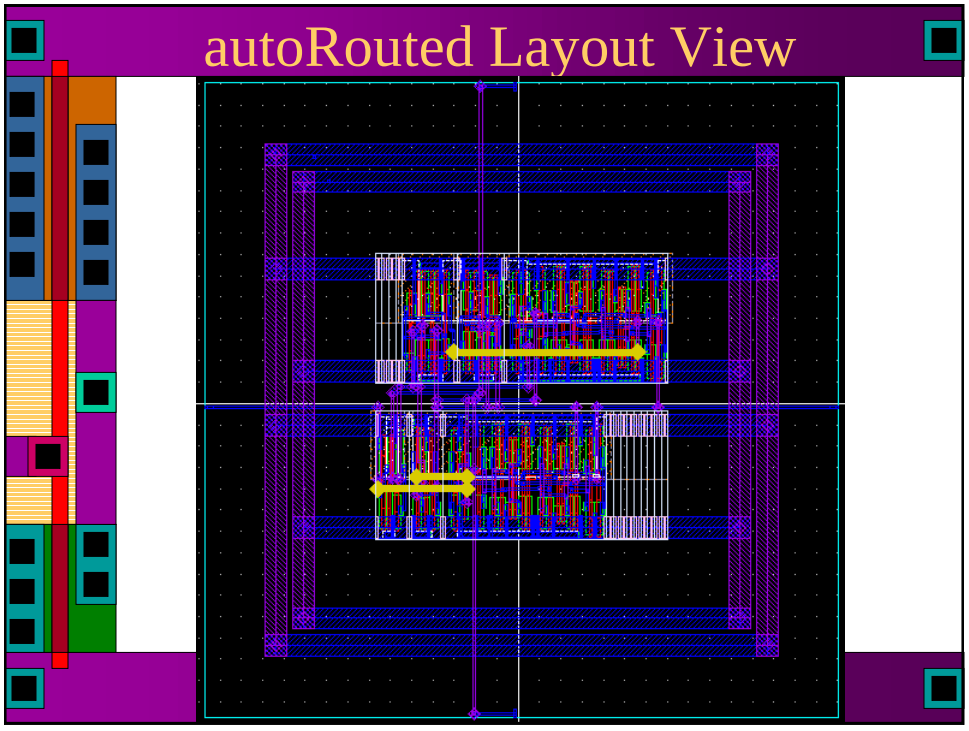
Place Cells and Fillers



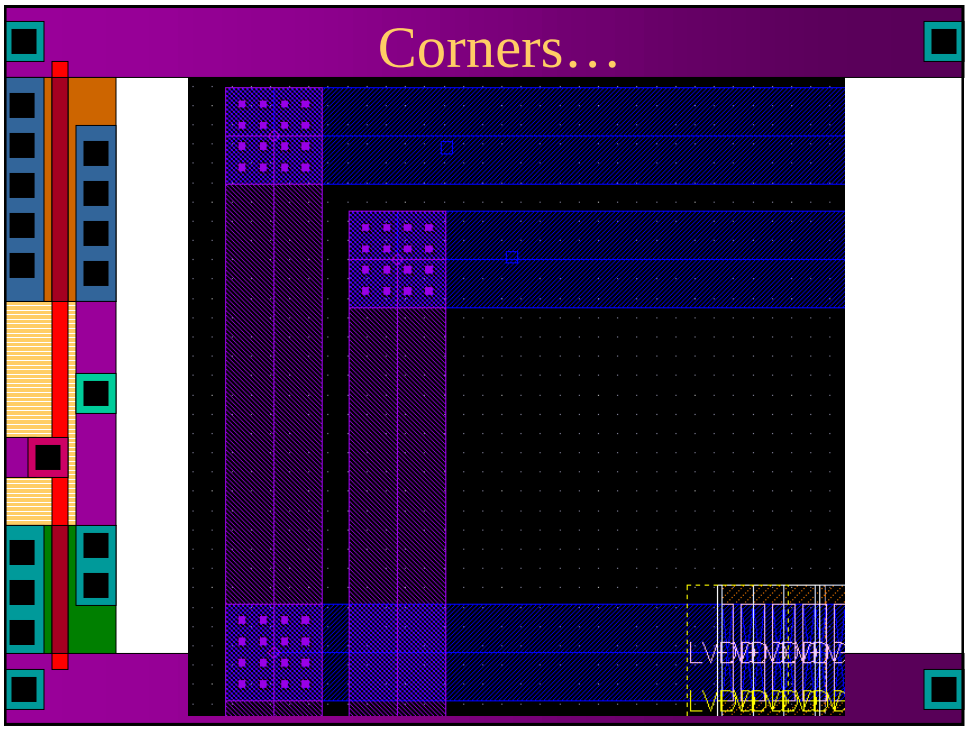
Connect Rows to Power



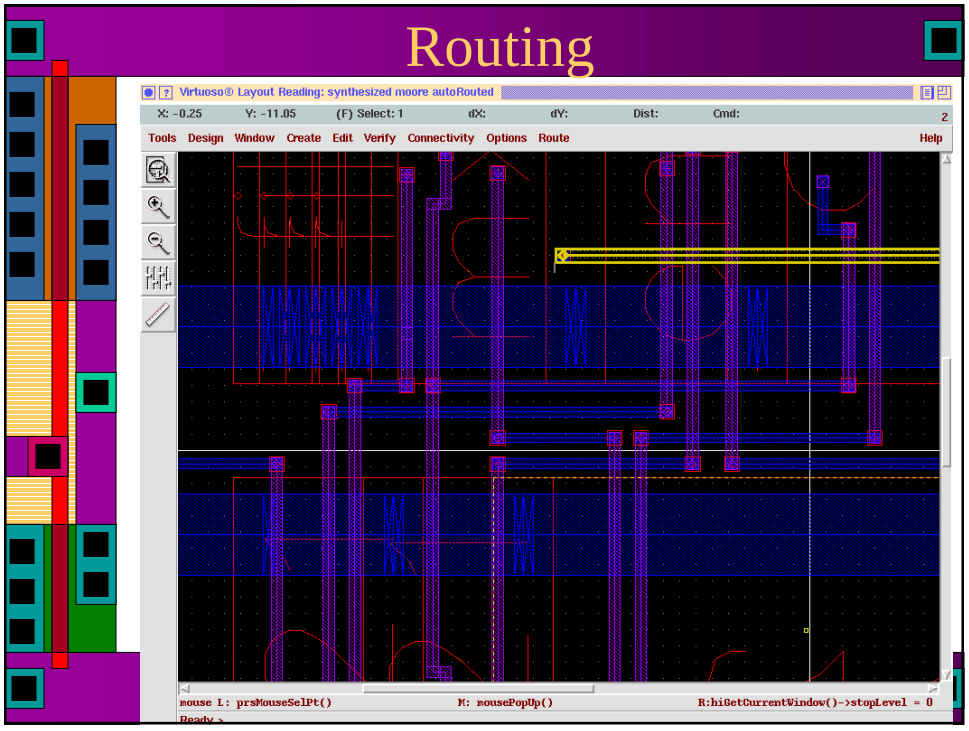

autoRouted View


autoRouted Layout View

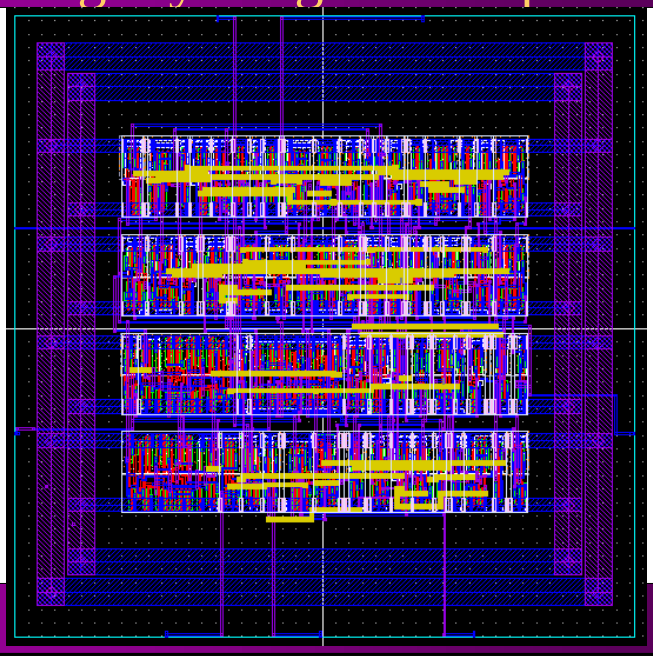
Corners…



Routing



Slightly Larger Example

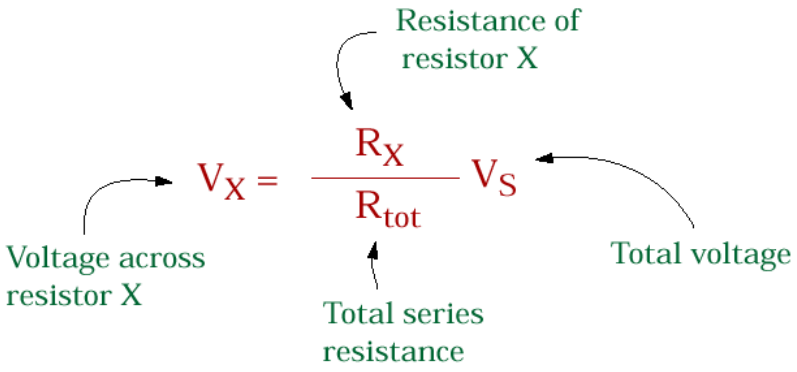


Electronics Summary

- Voltage is a measure of electrical potential energy
- Current is moving charge caused by voltage
- Resistance reduces current flow
  - Ohm's Law: $V = I R$
- Power is work over time
  - $P = V I = I^2R$
- Capacitors store charge
  - It takes time to charge/ discharge a capacitor
  - Time to charge/discharge is related exponentially to RC
  - It takes energy to charge a capacitor
  - Energy stored in a capacitor is $(1/2)\ C\ V^2$

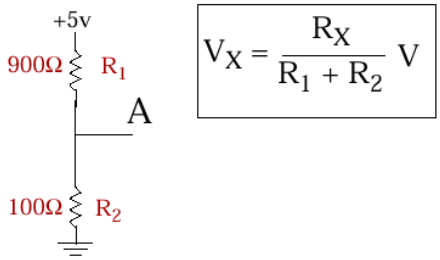
Reminder: Voltage Division

- Find the voltage across any series-connected resistors

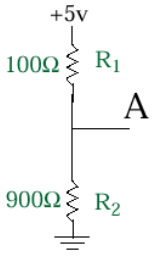


Example of Voltage Division

- Find the voltage at point A with respect to GND



$$V_X = \frac{R_X}{R_1 + R_2} V$$

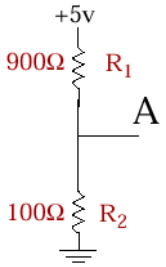


$V_1 = (900/1000)\ 5v = 4.5v$
$V_2 = (100/1000)\ 5v = 0.5v$
So, $V_{A\text{-}GND} = 0.5v$

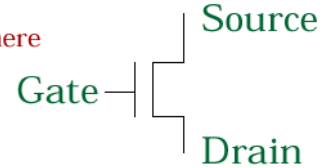
$V_1 = (100/1000)\ 5v = 0.5v$
$V_2 = (900/1000)\ 5v = 4.5v$
So, $V_{A\text{-}GND} = 4.5v$

How Does This Relate to VLSI?

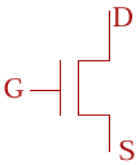
- Recall the voltage division example:
  - Consider what we could do if we had a device that we could switch from high resistance to low resistance
  - We could use it to force A high or low depending on the relative resistance of the elements



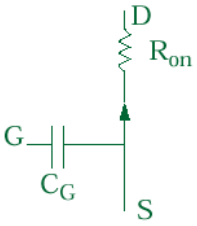
- This is a transistor
  - Specifically a CMOS FET
  - Complementary Metal-Oxide Semiconductor Field Effect Transistor
  - If voltage on Gate is high, then there is a low-resistance between Source and Drain, otherwise it's a very high-resistance



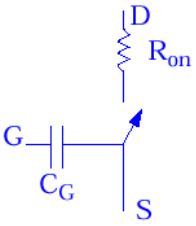
Model of a CMOS Transistor



Switch Level Model



Switch is closed if Gate voltage is high

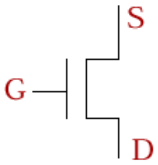


Switch is open if Gate voltage is low

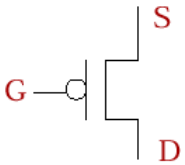
R_{on} = Some resistance in FET itself
C_G = Capacitance of the gate

Two Types of CMOS Transistors

- N-type transistor
  - High voltage on Gate connects Source to Drain
  - Passes 0 well, passes 1 poorly

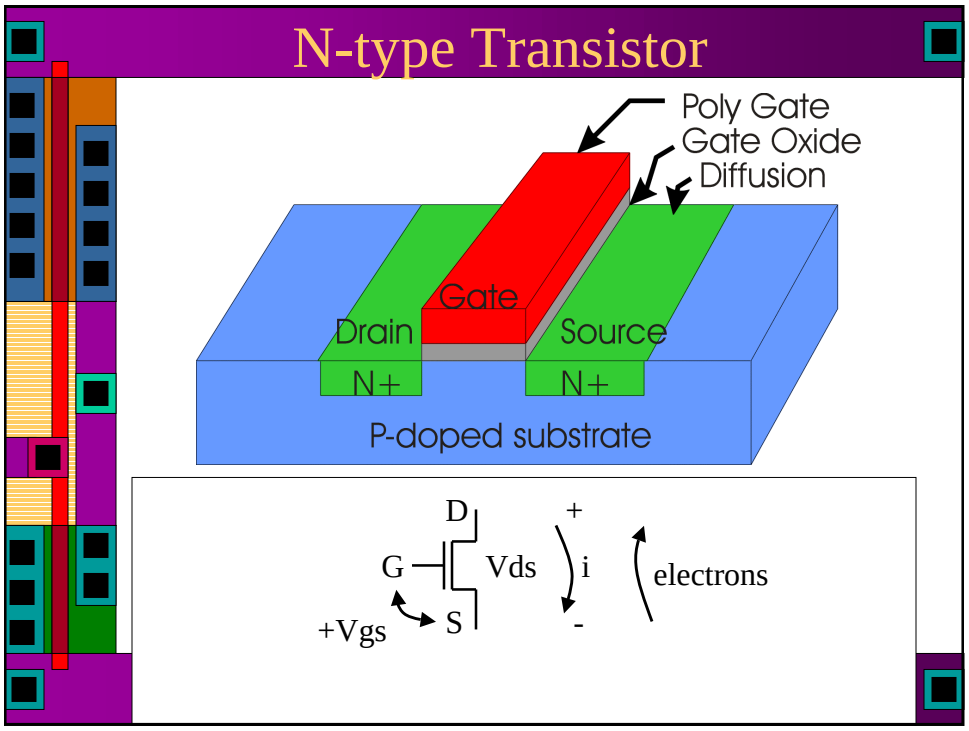


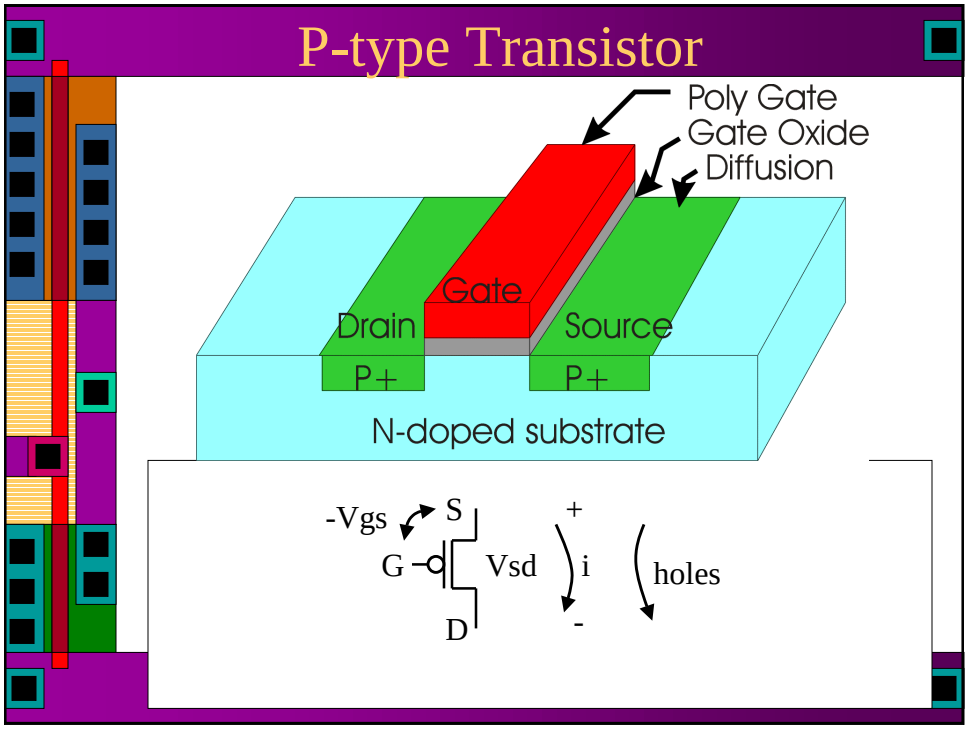
- P-type transistor
  - Low voltage on Gate connects Source to Drain
  - Passes 1 well, passes 0 poorly



CMOS Transistors

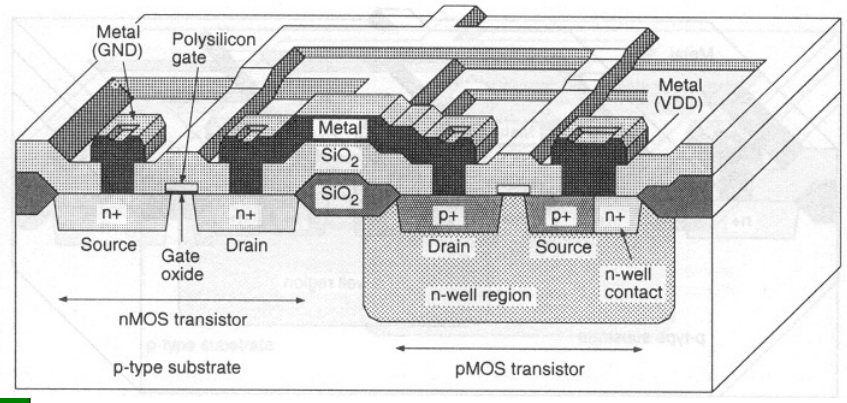
- Complementary Metal Oxide Semiconductor
- Two types of transistors
  - Built on silicon substrate
  - "majority carrier" devices
  - Field-effect transistors
    - An electric field attracts carriers to form a conducting channel in the silicon…
    - We'll get much more of this later…
    - For now, just some basic abstractions


N-type Transistor
Poly Gate
Gate Oxide
Diffusion
Gate
Drain
Source
N+
N+
P-doped substrate
D
G
S
+Vgs
Vds
+
i
-
electrons


P-type Transistor
Poly Gate
Gate Oxide
Diffusion
Gate
Drain
Source
P+
P+
N-doped substrate
-Vgs
S
G
D
Vsd
+
i
-
holes

A Cutaway View

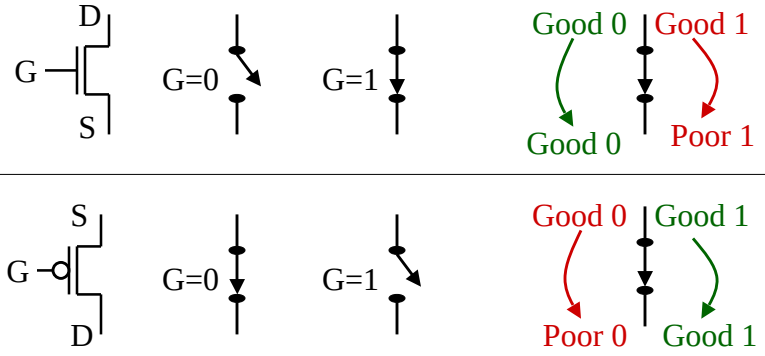
- CMOS structure with both transistor types



Transistors as Switches

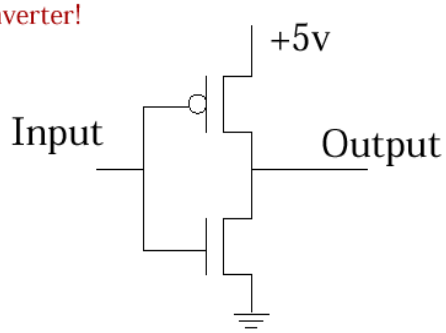
- For now, we'll abstract away most analog details…

Not Perfect Switches!



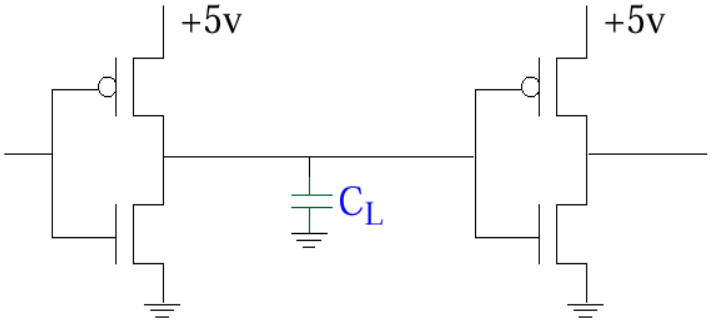
CMOS Inverter

- Consider this connection of transistors
  - If input is at a high voltage, output is low
  - If input is at a low voltage, output is high
- By changing the resistances, it becomes one of two different voltage dividers
  - It's a voltage inverter!

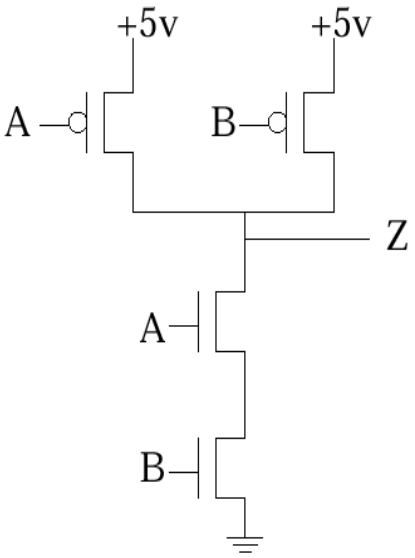


Timing Issues in CMOS

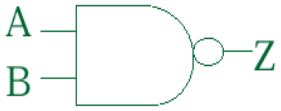
- Recall that it takes time to charge capacitors
- Recall that the gate of a transistor looks like a capacitor
- Wires have resistance and capacitance also!



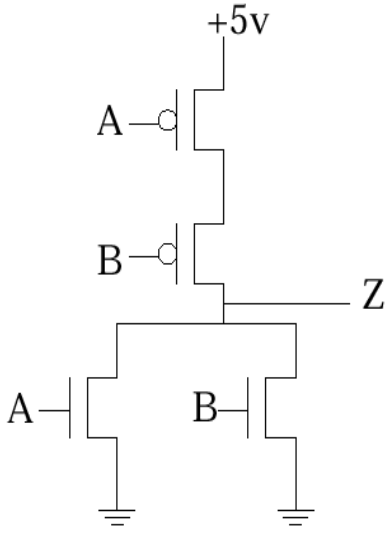
CMOS NAND Gate



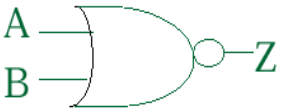
A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0



CMOS NOR Gate

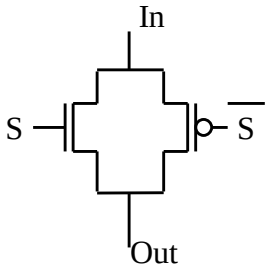


A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0



N-type and P-type Uses

- Because of the imperfect nature of the the transistor switches
  - ALWAYS use N-type to pull low
  - ALWAYS use P-type to pull high
  - If you need to pull both ways, use them both



S=0, In ≠ Out

S=1, In = Out

Power Consumption

- Power is consumed in CMOS by charging and discharging capacitors
  - Note that there no static power dissipation in CMOS
  - There's never a DC path to ground
- Good news:
  - You're not consuming power unless you're switching
- Bad news:
  - Switching activity is caused by clock, which is going faster and faster
- If the first-order power effect is capacitor charging/discharging, and the clock causes this:

$$P = (1/2)\ C\ V^2\ f$$

Switch to Chalkboard

- Complex Gate
- Tri-State
- Latch
- D-register