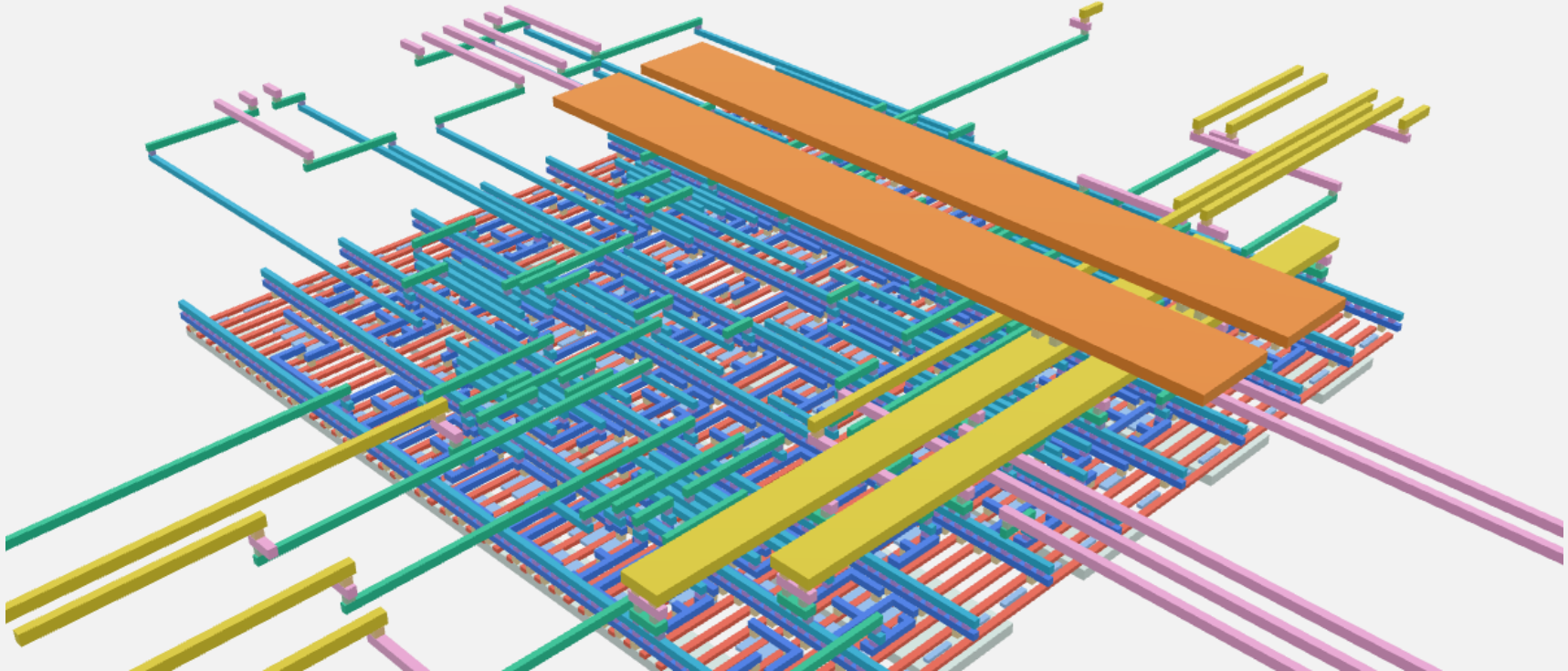
1.1

CSE140 – Digital Design

2026 Summer Session 1

Abarajithan G



About Me

- PhD Student (4th year), Computer Engineering, UCSD
- Intern @ Qualcomm (2023), AMD (2024)
- Website: abapages.com

- Former experience:
  - Undergrad Internship at CSIRO (Australia) – Robotics & AI (2018)
  - BS – ENTC UoM, Sri Lanka (2020)
  - Junior Instructor at ENTC, UoM
  - RTL Design Engineer at Lemurian Labs (Canada)



Join Discord

Discussion about policy, assignments



Feedback Fruits

In-class quiz & feedback

Canvas → CSE140 → Assignments → Slides 1.1

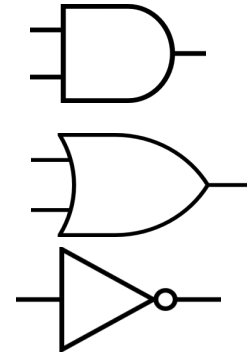


First day info sheet (1)

- My full name is:
- My preferred name is:
- Pronouns I go by (if any):
- My career goals are:
- Place(s) I grew up:
- Something I am good at, or that makes me proud of myself:
- I have reliable access to Internet/Canvas:
- A word that describes how I feel about this class at the moment:
- Number of hours per week I'll be spending for this course (outside the class):
- I appreciate when my instructors:
- I dislike when my instructors:
- One current question I have about this class:

First day info sheet (2)

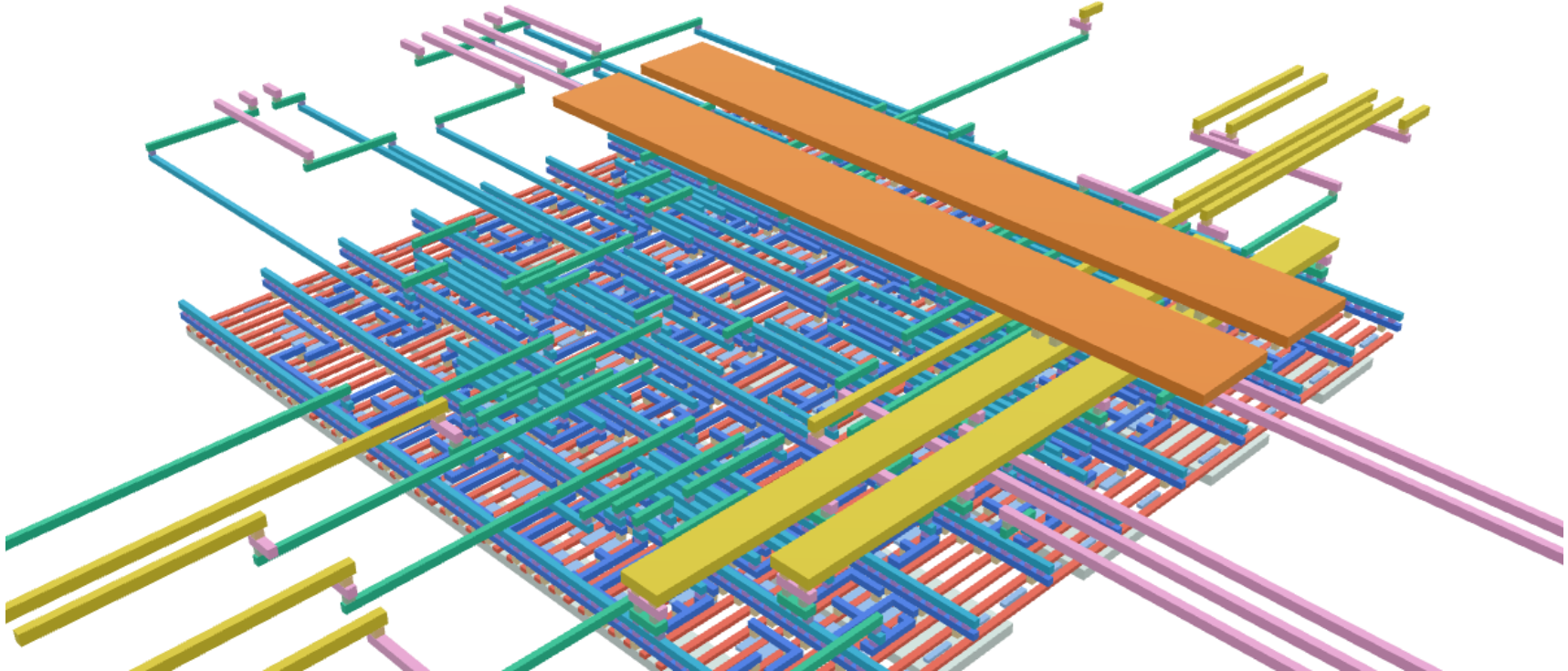
- I’m taking this class because:
- I chose to do this in summer because:
- The course I’m taking next:
- These logic gates are:



- This truth table corresponds to ___ gate:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Course Overview & Logistics



Assessments & Grading

- 1 assignment / week
- Two paper exams
  - Due to AI
  - If you miss, contact via Dean
  - Alternative: oral exam
- In-class quizzes

Weight	Component
10% each	Assignments (4)
15%	Midterm exam, in person and paper-based
40%	Final exam, in person and paper-based
5%	In-class quiz correctness
100%	**Total**
-2%	For each missed session (if any in-class quiz was not answered)
+0.5%	Per unused late day
up to +5%	Meaningful participation in Discord & FeedbackFruits with peers
up to +5%	FIR Filter Resource Optimization
up to +10%	Neural Network Resource Optimization

- **Attendance**: Mandatory for all lectures & discussions. 2% penalty for each missed session
- **Late work:** Four late days for the whole course. -10% per day after that.
- **AI Use:** Can use as search engine / to explain concepts. IDE based AIs (codex…etc) and any kind of generation of solution is not allowed.
- **Plagiarism:** We check all assignments after the course. F for all parties involved.

Questionnaire on Syllabus

1. Attendance recovery:
2. Late day:
3. AI use:
4. Exam emergency:
5. Plagiarism:

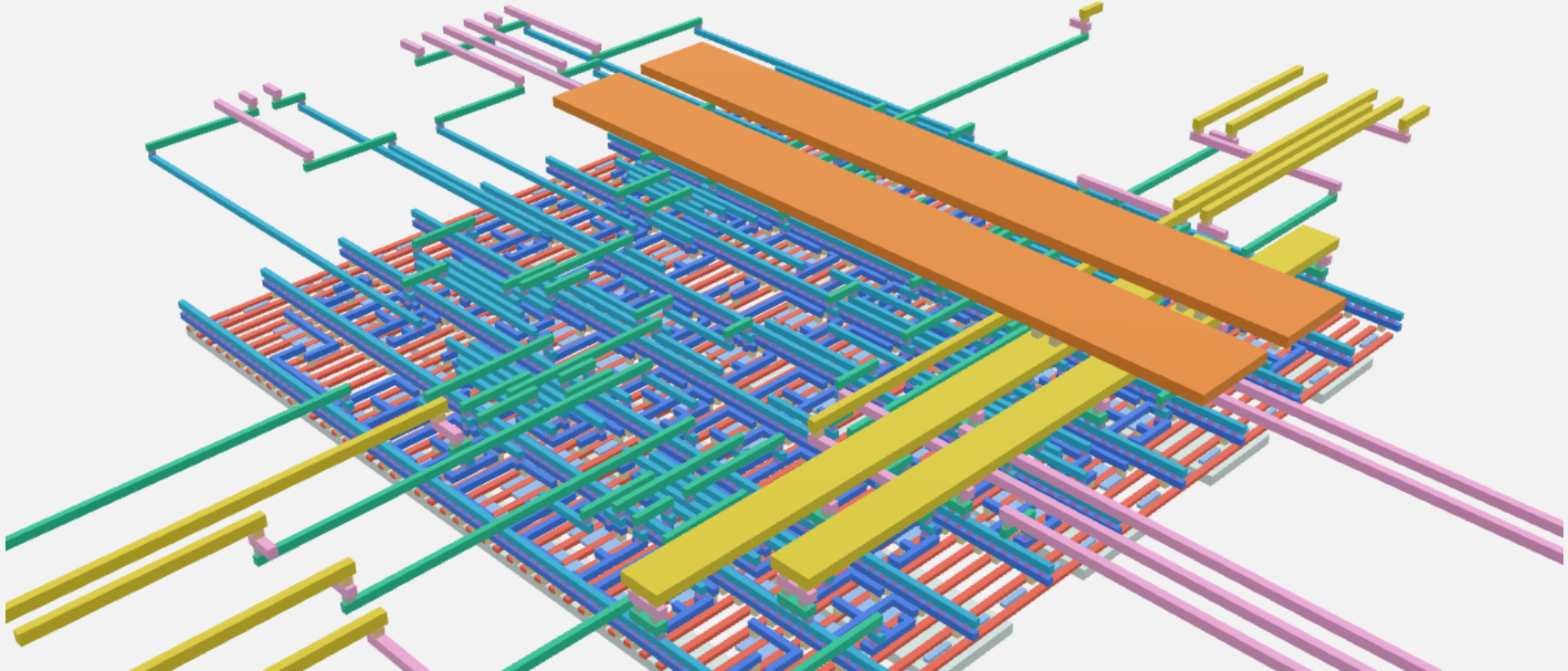
Your Basic Needs Come First

- Are you eating properly? Do you have adequate access to nutritious food? Do you have stable housing? Are you homeless or couch surfing?
- If you or someone you know is suffering from food and/or housing insecurities, please note: The Triton Food Pantry (in the old Student Center: fb.com/ tritonfoodpantry), is free and anonymous, and includes produce.
- Financial aid resources, the possibility of emergency grant funding, and off-campus housing referral resources are available.
- CAPS (caps.ucsd.edu) and college deans can connect students to the above resources, as well as other community resources and support.

Sexual Violence and Harassment

- The Office for the Prevention of Harassment & Discrimination (OPHD) provides assistance to students, faculty, and staff regarding reports of bias, harassment, and discrimination. OPHD is the UC San Diego Title IX office. Title IX of the Education Amendments of 1972 is the federal law that prohibits sex discrimination in educational institutions that are recipients of federal funds. Jacobs School students have the right to an educational environment that is free from harassment and discrimination.
- Students have options for reporting incidents of sexual violence and sexual harassment. Sexual violence includes sexual assault, dating violence, domestic violence, and stalking. Information about reporting options may be obtained at OPHD at (858) 534-8298, ophd@ucsd.edu or ophd.ucsd.edu. Students may receive confidential assistance at CARE at the Sexual Assault Resource Center at (858) 534-5793, sarc@ucsd.edu or care.ucsd.edu or Counseling and Psychological Services (CAPS) at (858) 534-3755 or caps.ucsd.edu.
- Students may feel more comfortable discussing their particular concern with a trusted employee. This may be a Jacobs School student affairs staff member, a department Chair, a faculty member or other University official. These individuals have an obligation to report incidents of sexual violence and sexual harassment to OPHD. This does not necessarily mean that a formal complaint will be filed.
- If you find yourself in an uncomfortable situation, ask for help. The Jacobs School is committed to upholding University policies regarding nondiscrimination, sexual violence and sexual harassment.

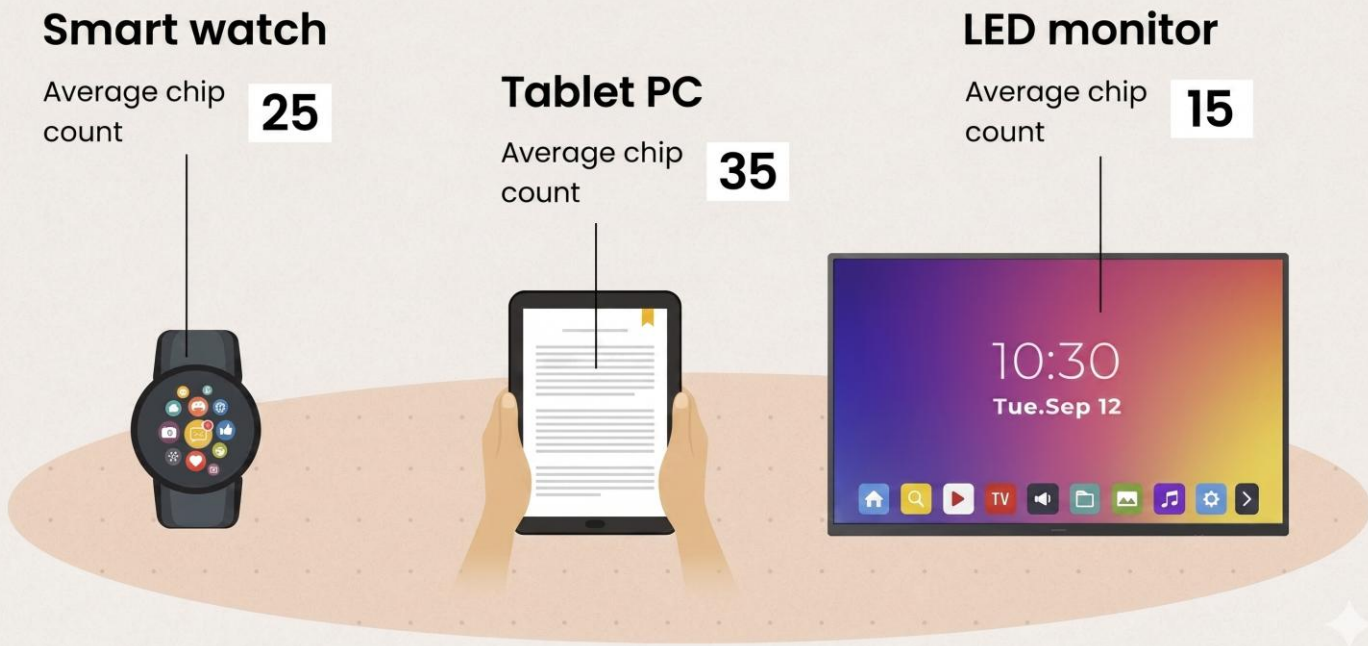
Inside a Chip



Chips, chips everywhere!

Number of chips in an average

- Toaster : **2**
- Electric toothbrush : **3**
- Laptop : **80**
- Smartphone : **70**
- Electric Vehicle : **1000**

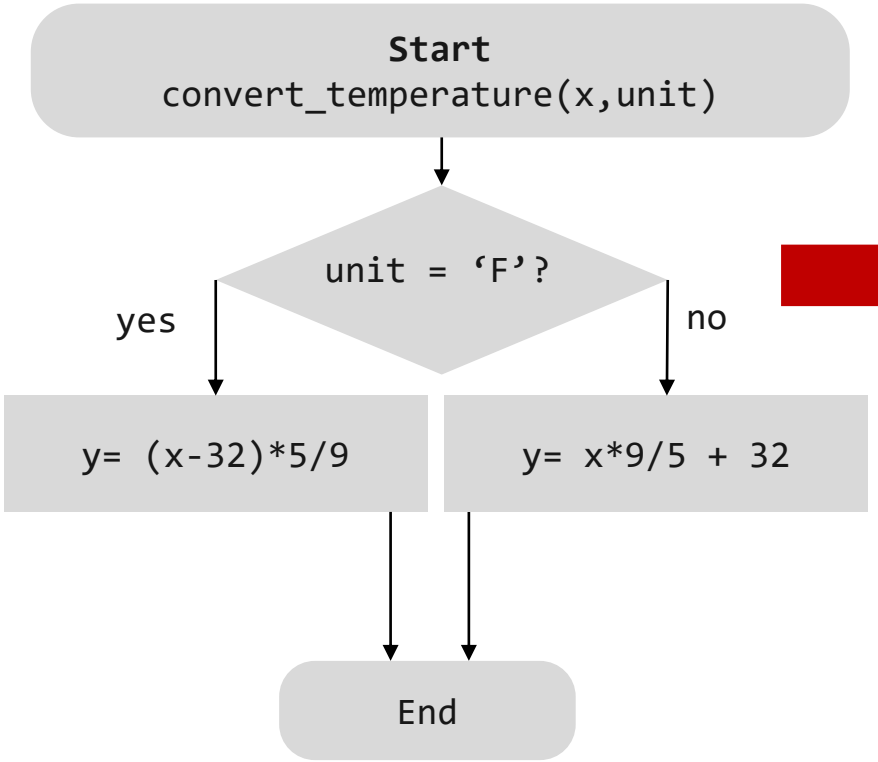


What CPU does your laptop have?

news.samsungsemiconductor.com/global/the-hidden-technology-powering-everyday-life/

Programming

1. Algorithm



2. C code

```c
float convert_temp(float x, char unit) {
    if (unit == 'F') return (x - 32) * 5 / 9;
    else             return x * 9 / 5 + 32;
}
```

Programming

Compiling

3. Assembly

```
convert:
        add     w8, w0, w0, lsl #3
        mov     w9, #5
        mov     w10, #9
        sdiv    w8, w8, w9
        add     w9, w0, w0, lsl #2
        sub     w9, w9, #160
        sdiv    w9, w9, w10
        and     w10, w1, #0xff
        add     w8, w8, #32
        cmp     w10, #70
        csel    w0, w8, w9, ne
        ret
```

Assembling

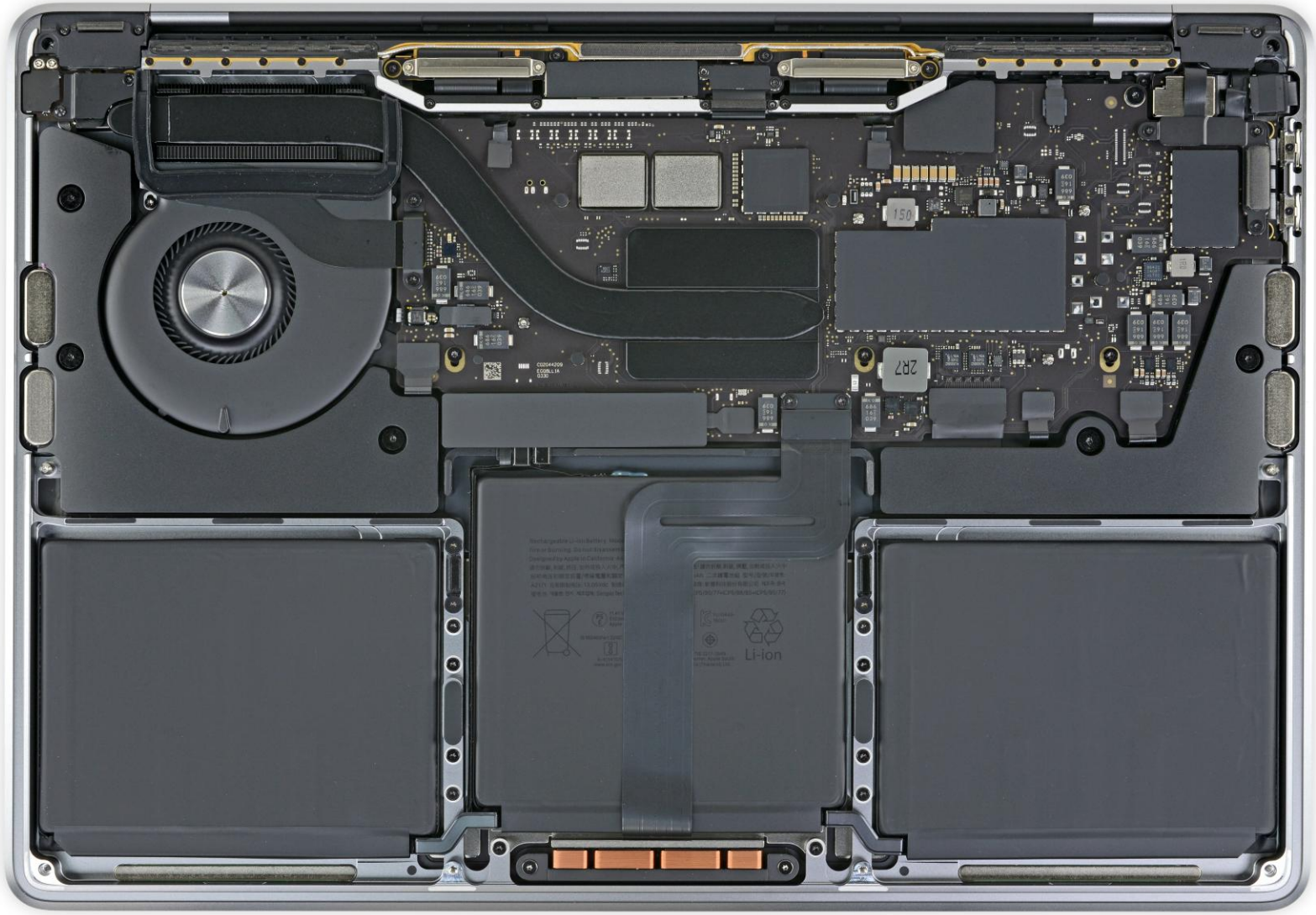
4. Machine code



Execution

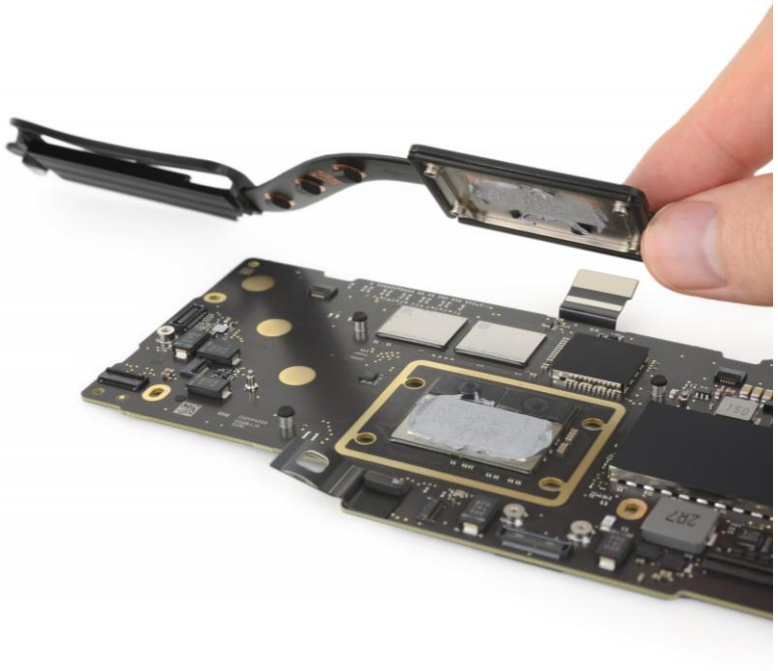


MacBook Pro with Apple M1



https://www.ifixit.com/News/46884/m1-macbook-teardowns-something-old-something-new

Inside a MacBook

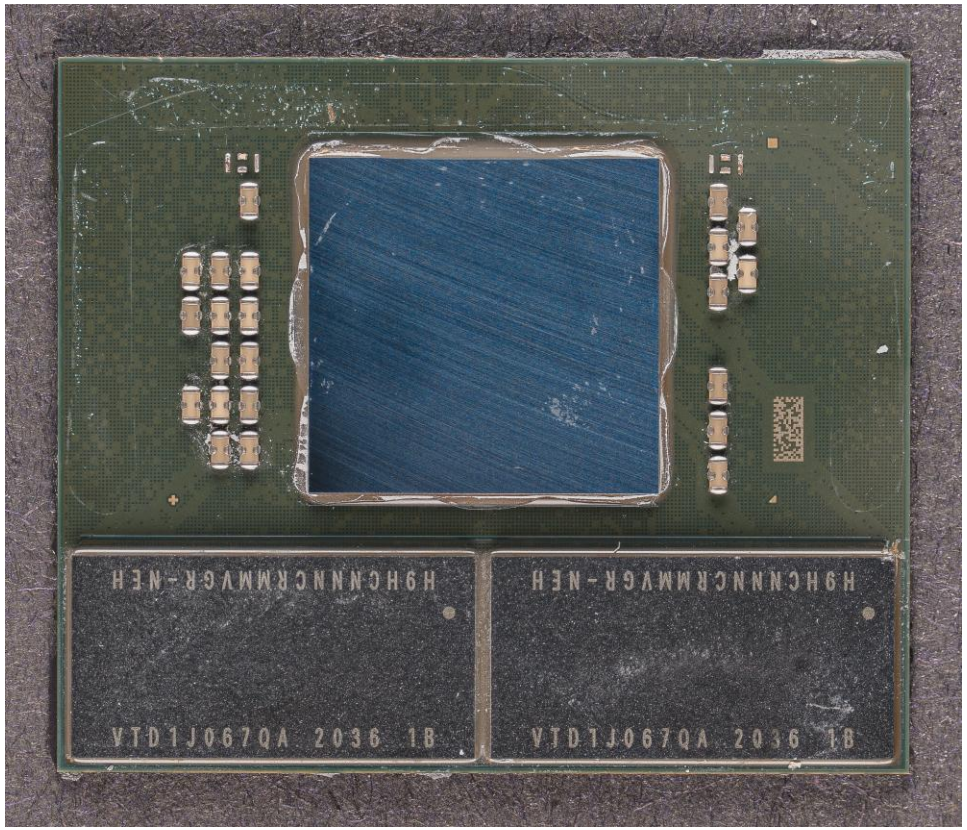


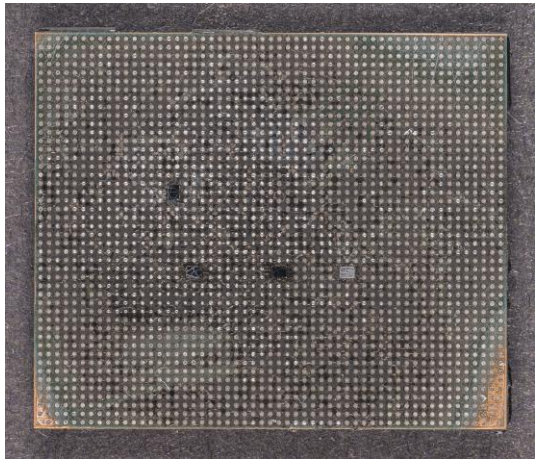


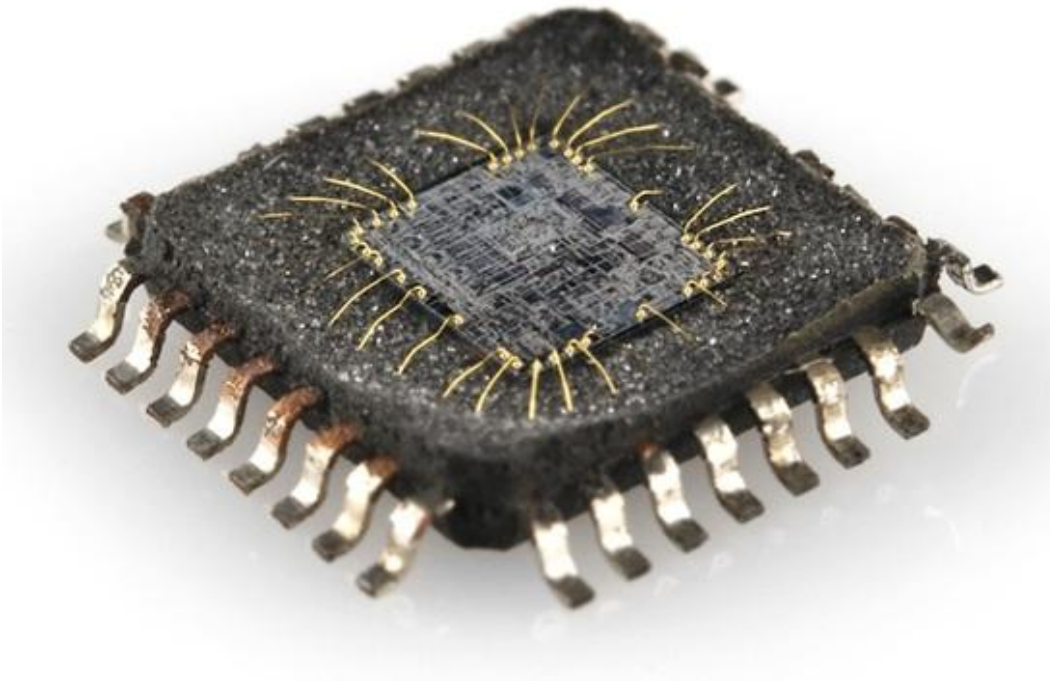


https://www.ifixit.com/News/46884/m1-macbook-teardowns-something-old-something-new

M1 Package





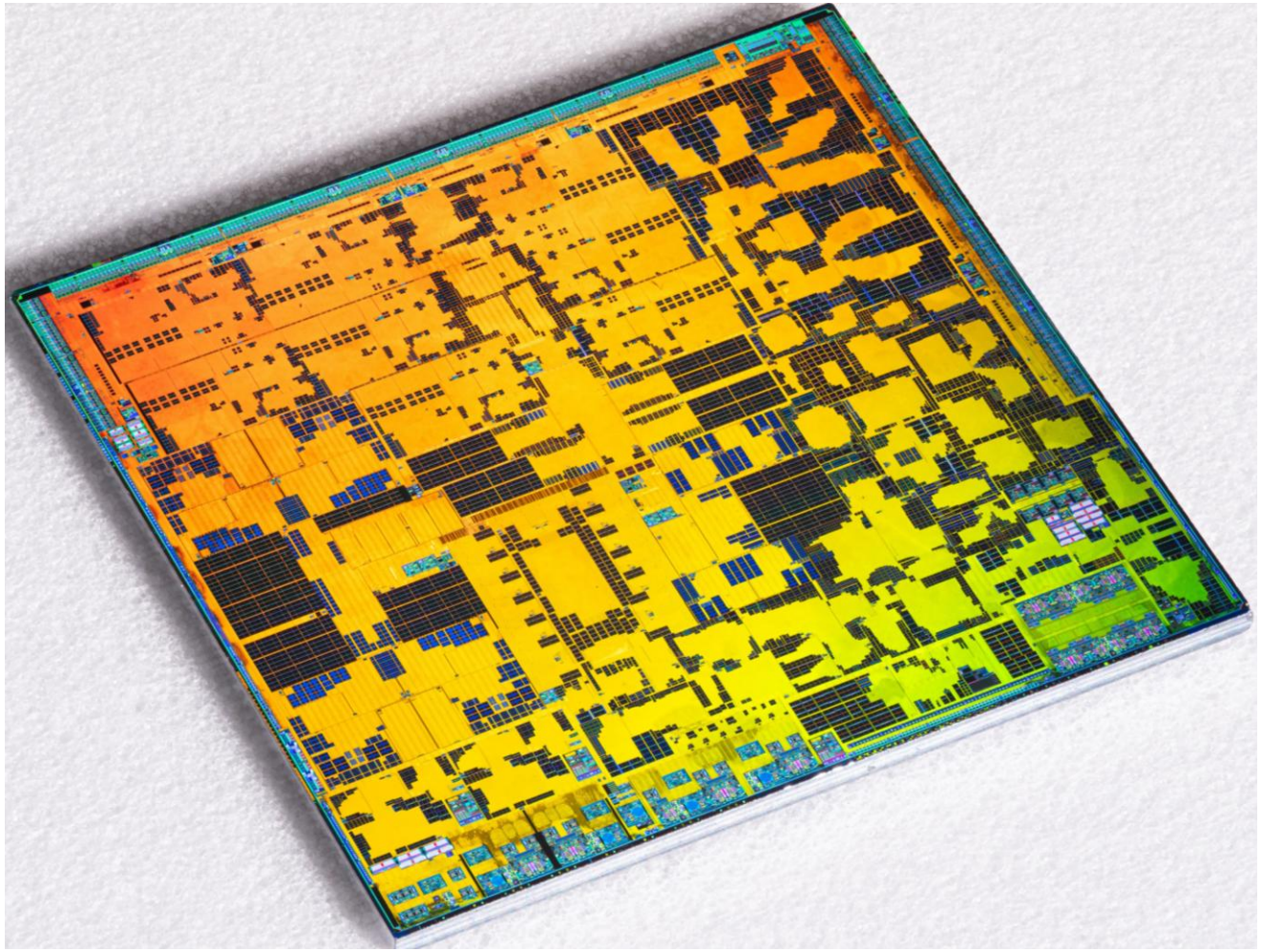


Atmega328p microcontroller used in Arduino

https://blog.adafruit.com/2010/06/27/sparkfun-investigates-fake-atmel-ics/

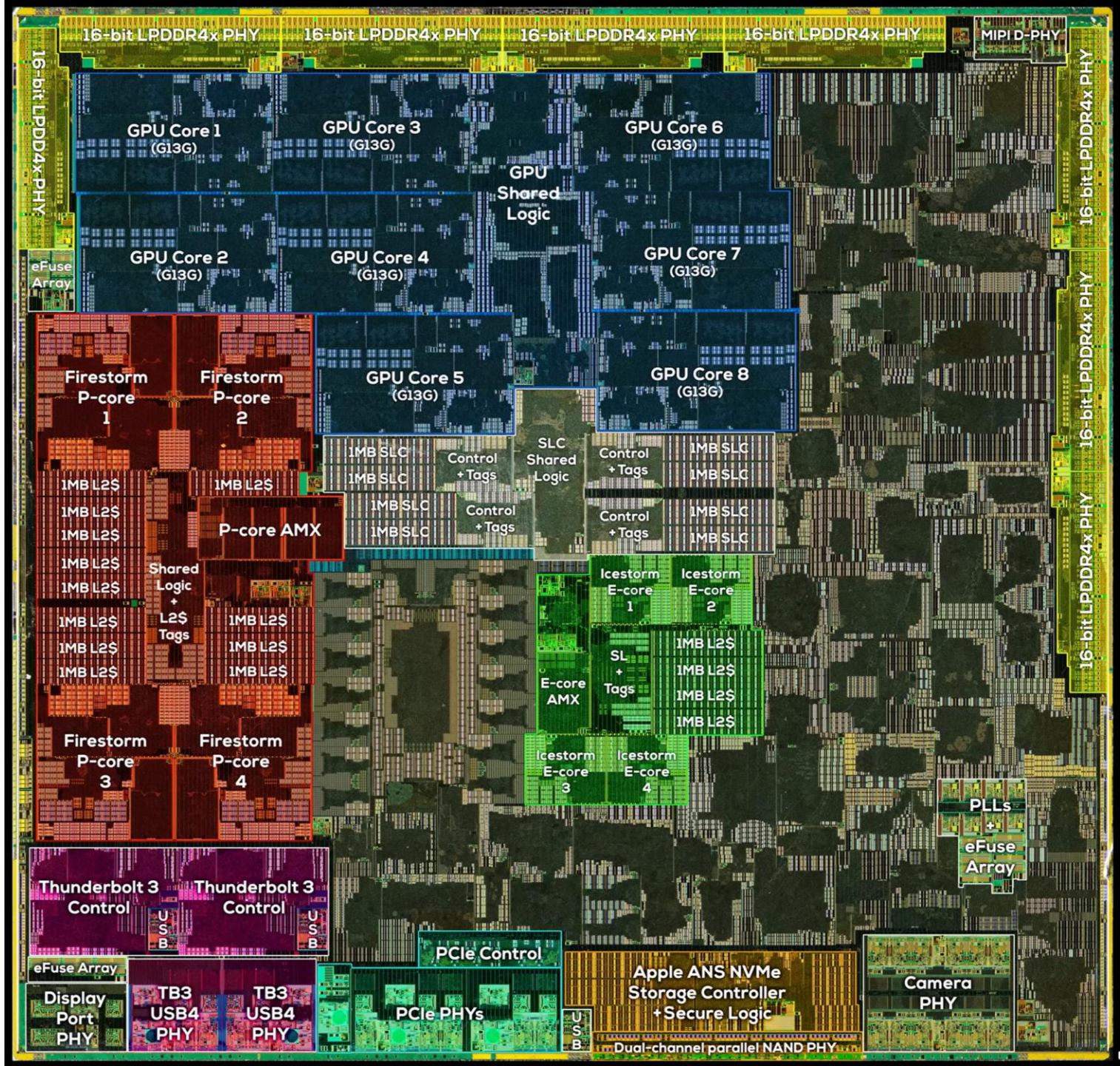
https://www.flickr.com/photos/130561288@N04/

M1 Die

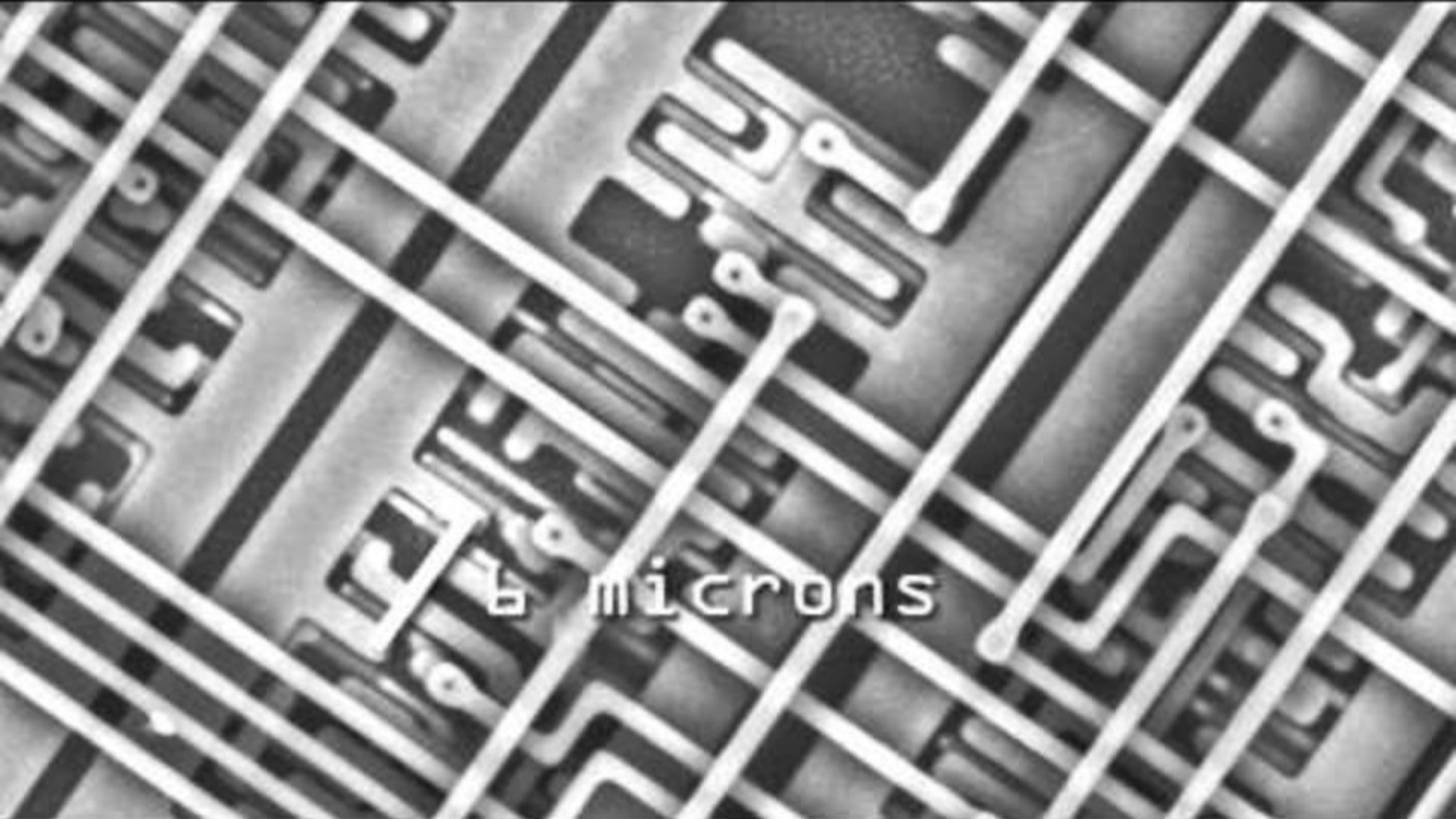


https://www.flickr.com/photos/130561288@N04/

Annotated Die Shot of M1 Chip

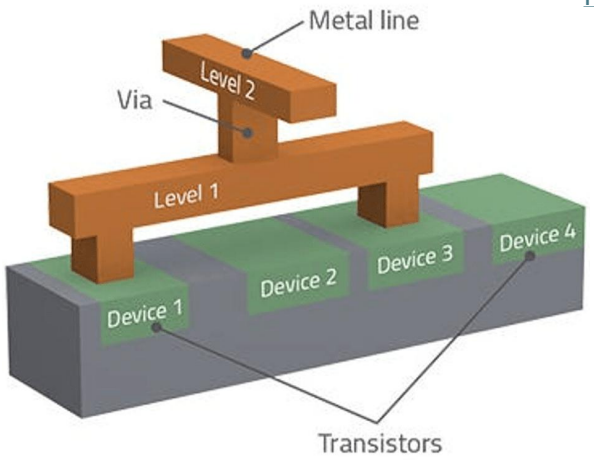


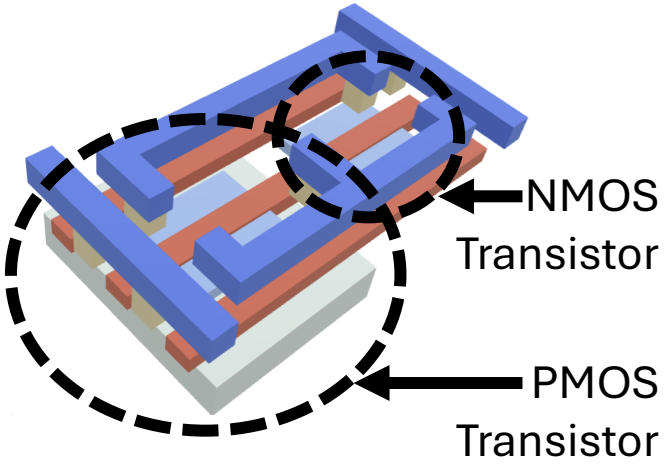
More info at
Apple M1 Chip Deep-Dive:
youtube.com/watch?v=mHEWMiHgyU8

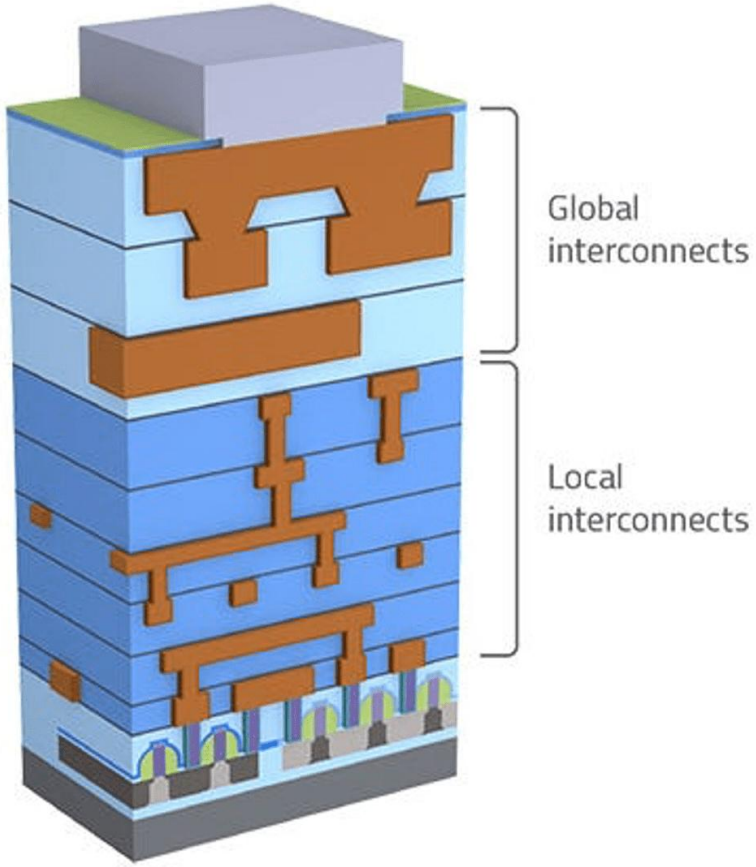

b microns

Inside the Die

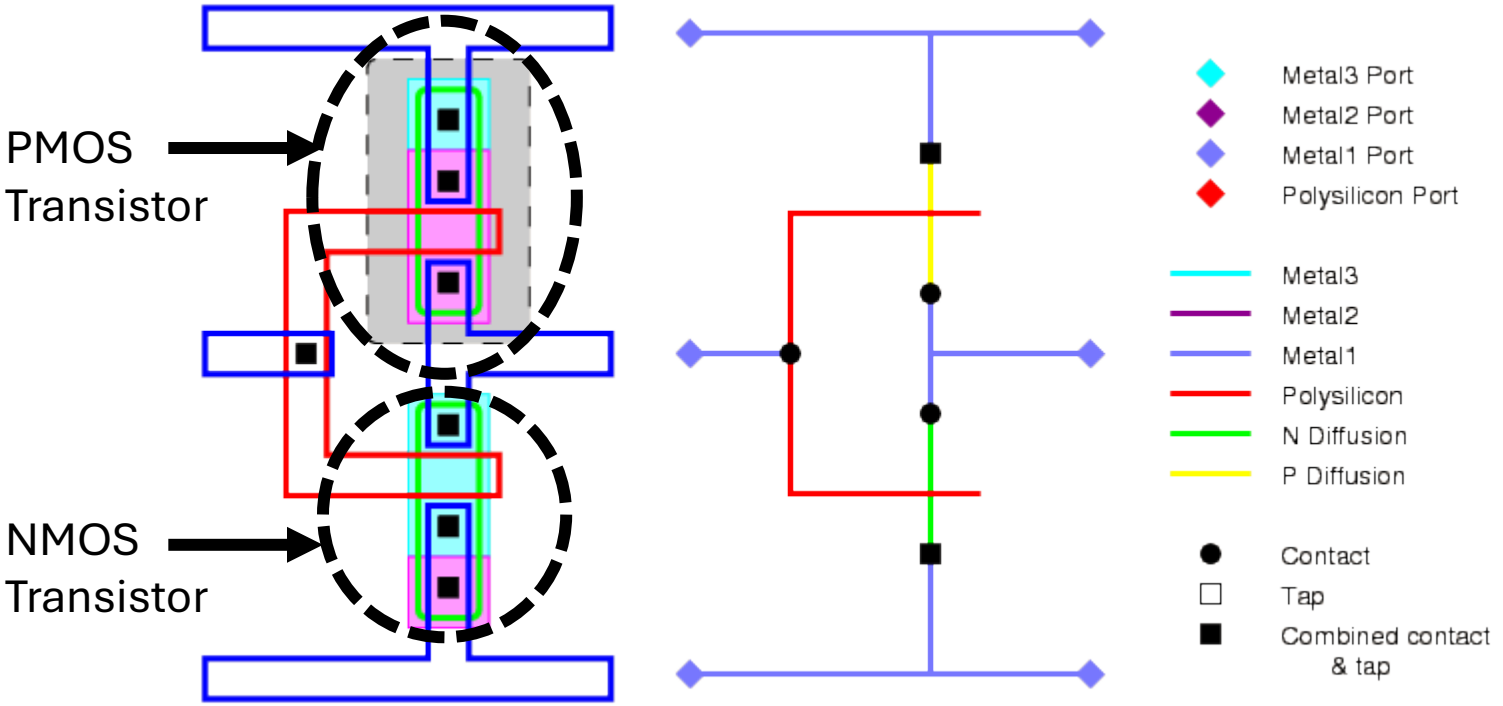
https://abapages.com/digital-design/3d-cells.html







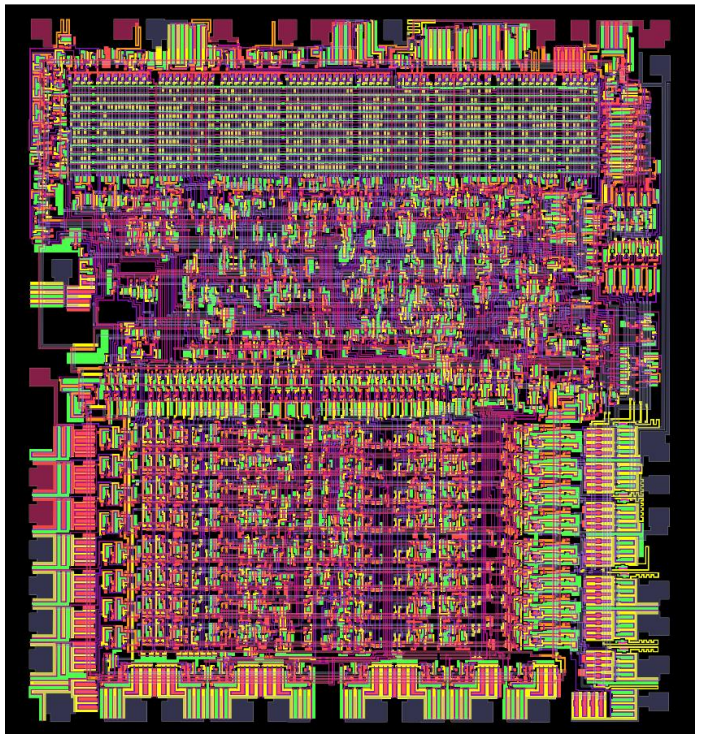
https://semiengineering.com/interconnects-approach-tipping-point/



https://www.southampton.ac.uk/~bim/notes/cad/guides/sticks.html

MOS 6502 Visualization

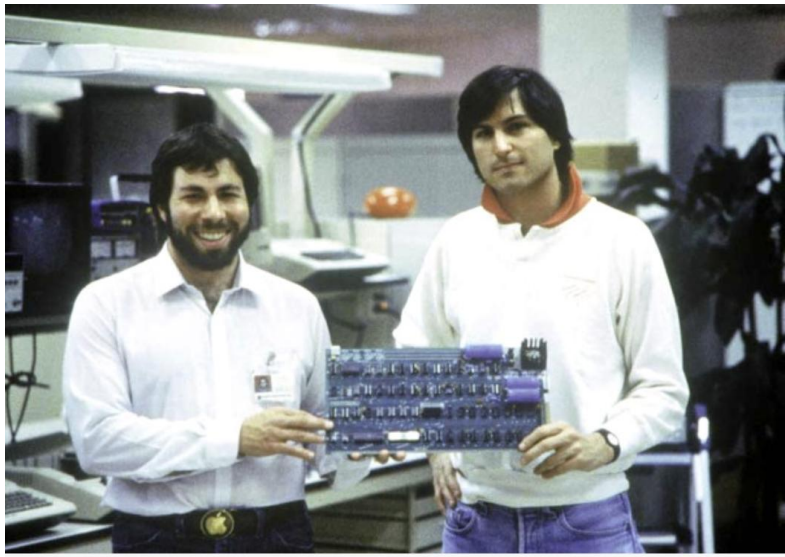
- MOS 6502
  - 1975 (48 years ago)
  - 1-3 MHz
  - 8-bit data, 16-bit instructions





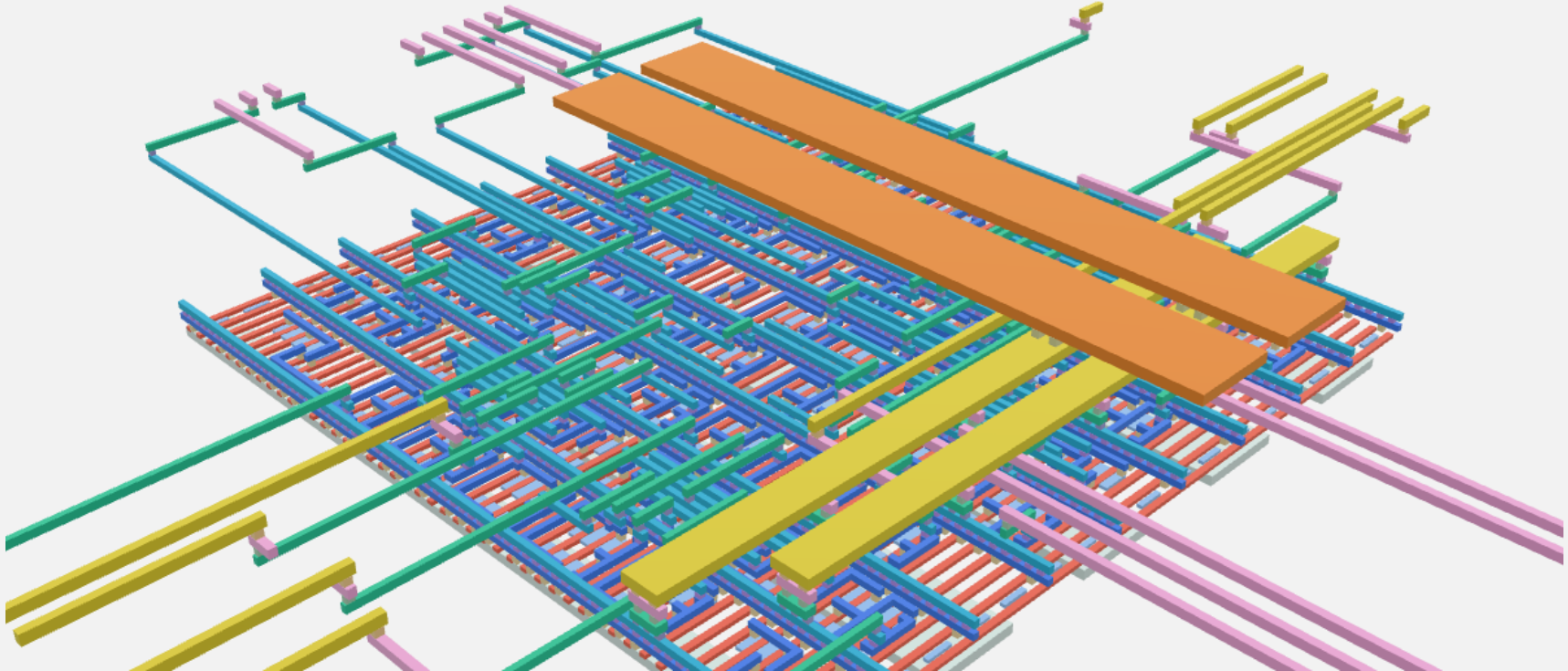
- Used in
  - Apple I, Apple II
  - Atari
  - Nintendo

- http://www.visual6502.org/JSSim/expert.html

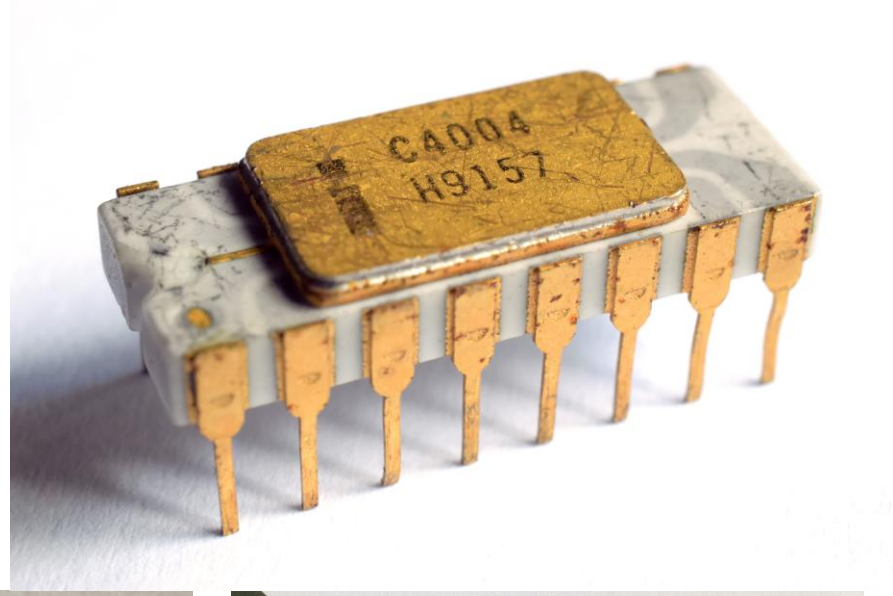


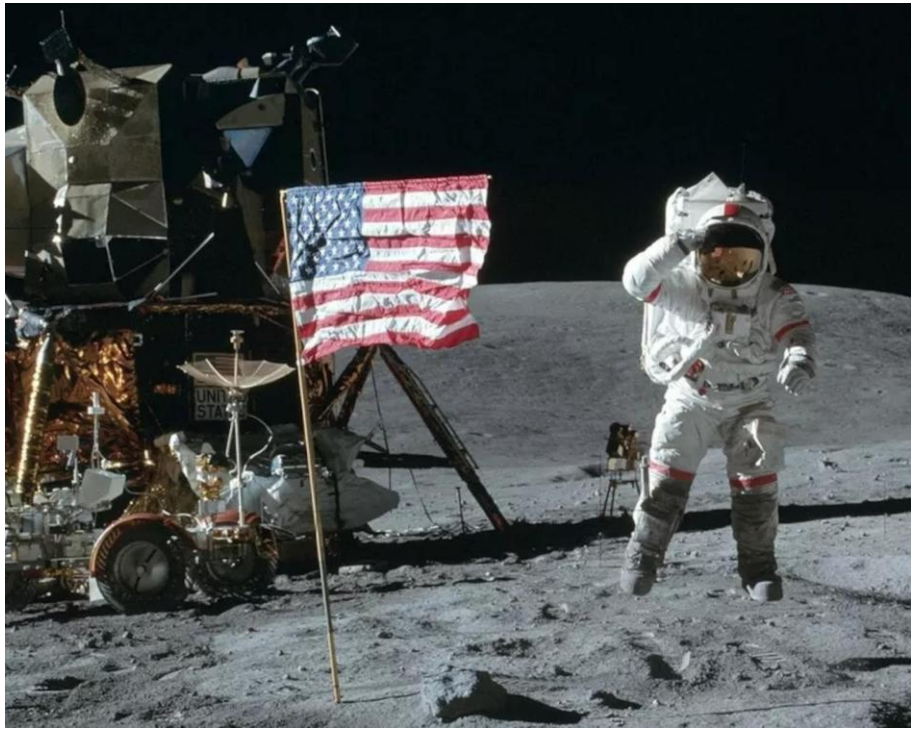
Steve Wozniak, Steve Jobs, and their Apple I.

Making a chip



Intel 4004: The first CPU





1969

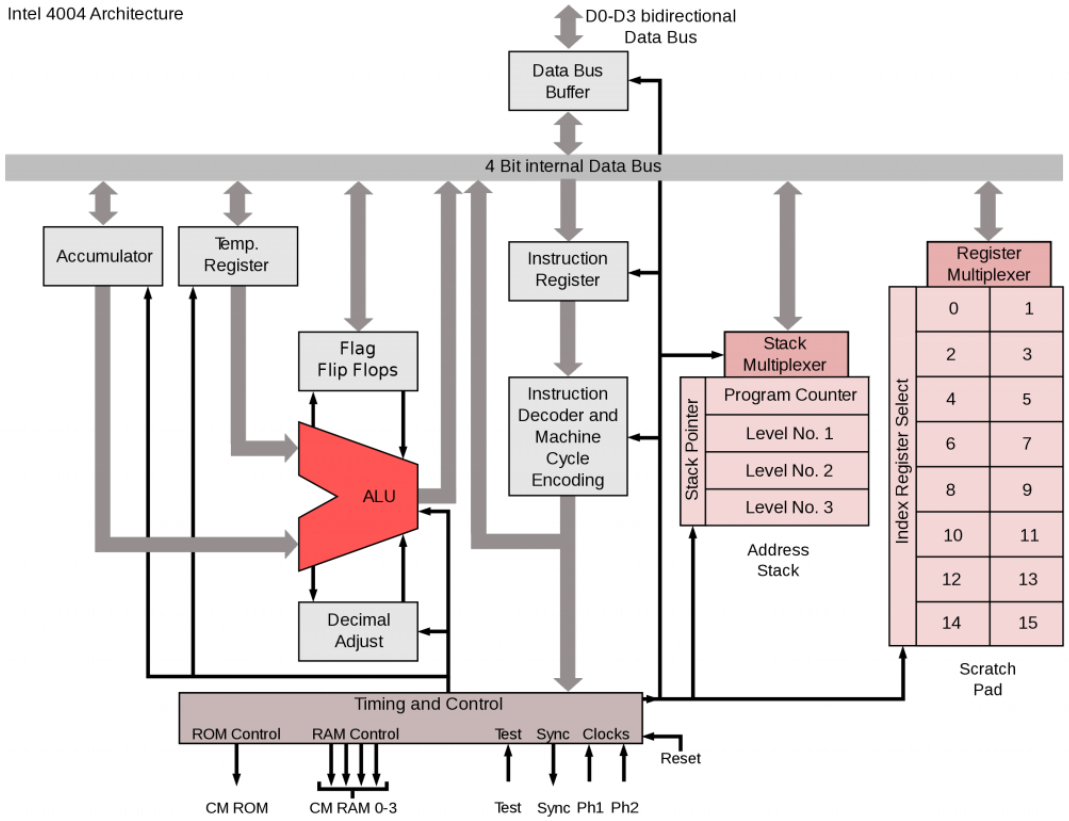


Busicom 141-pf Calculator

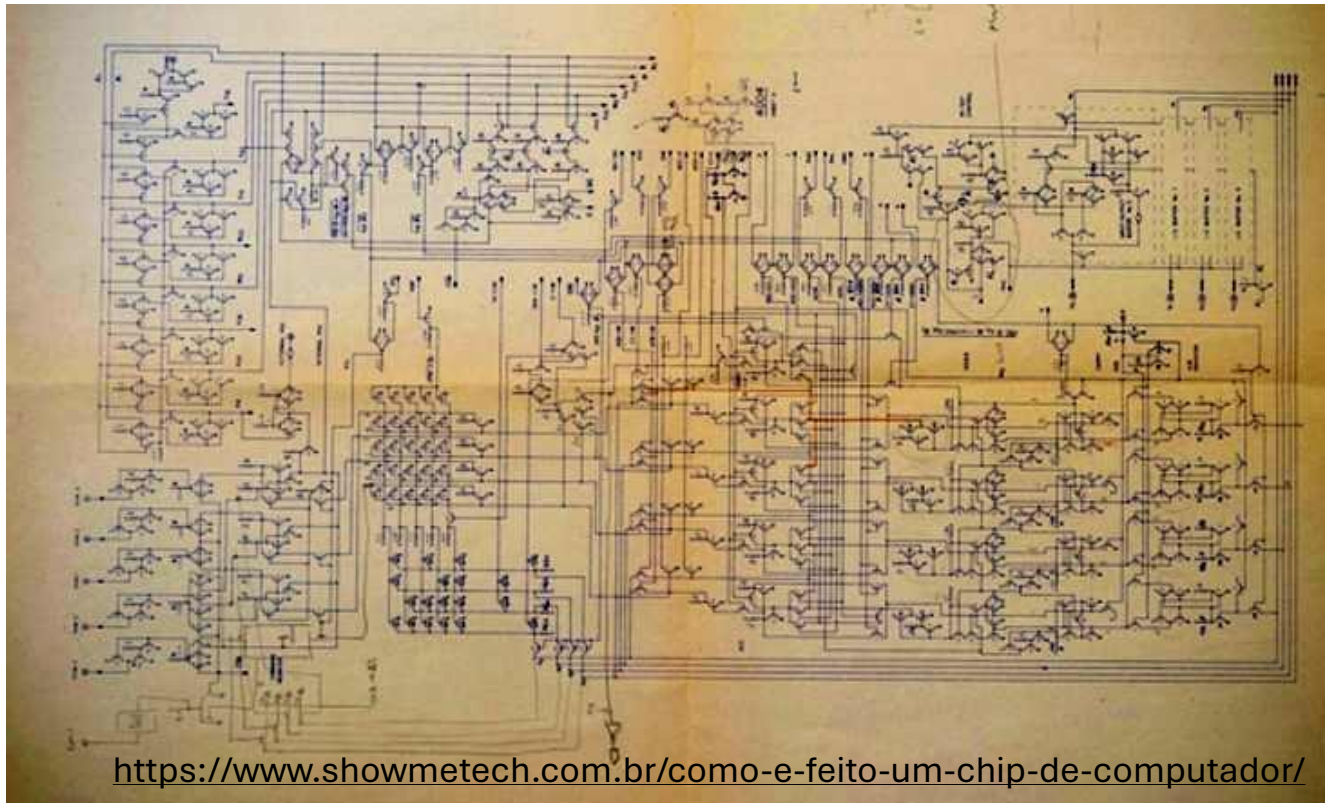


Federico Faggin

Computer Architecture, Schematic



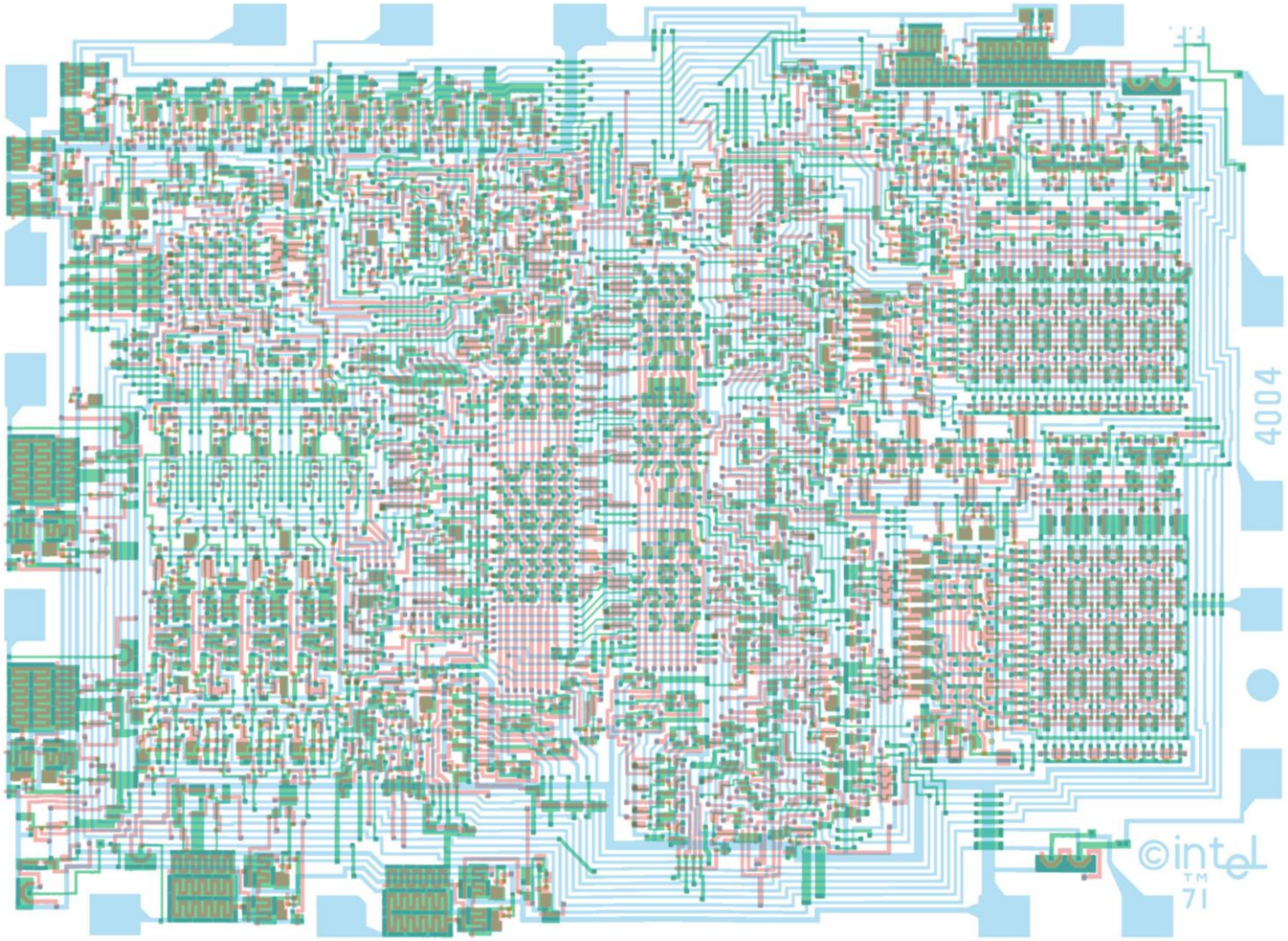
imgbin.com/png/WDSiUuBi/intel-4040-intel-4004-microprocessor-intel-8085-png



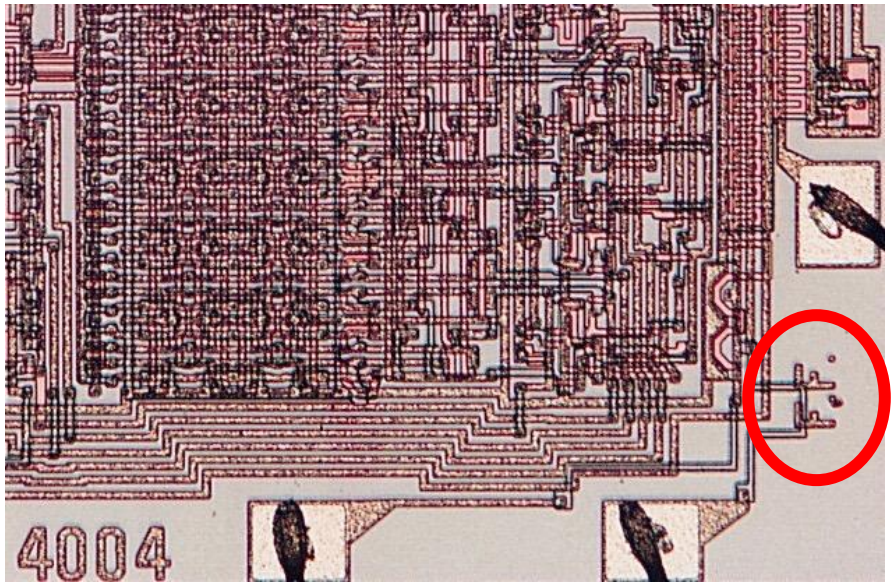
https://www.showmetech.com.br/como-e-feito-um-chip-de-computador/

More:
datasheets.chipdb.org/Intel/MCS-4/4004_schematic.pdf

Composite Layout

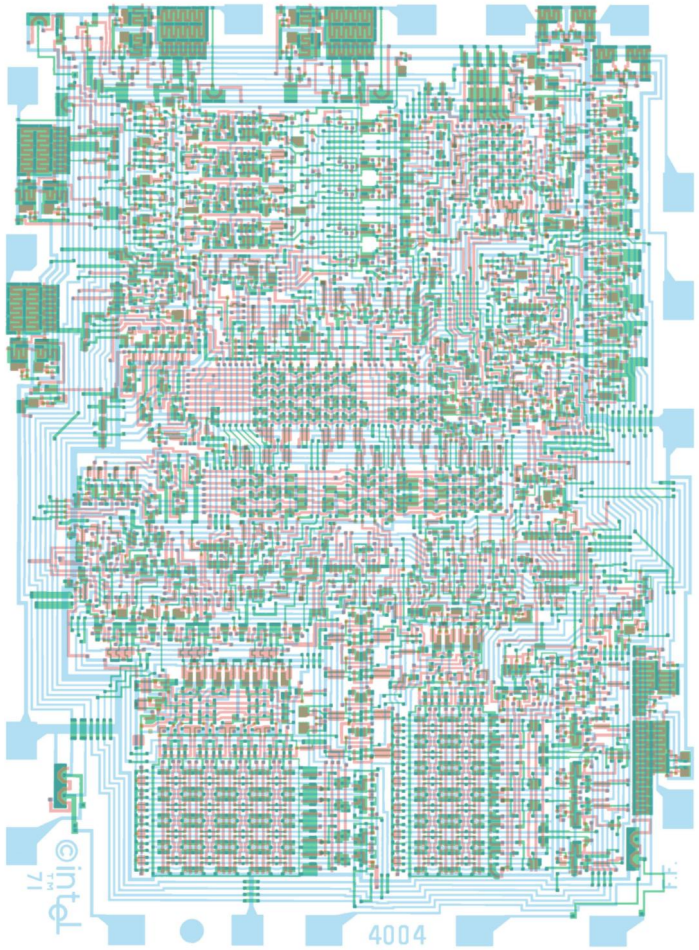


https://alumni.media.mit.edu/~mcnerney/2009-4004/4004-masks-composite.jpg



http://www.intel4004.com/

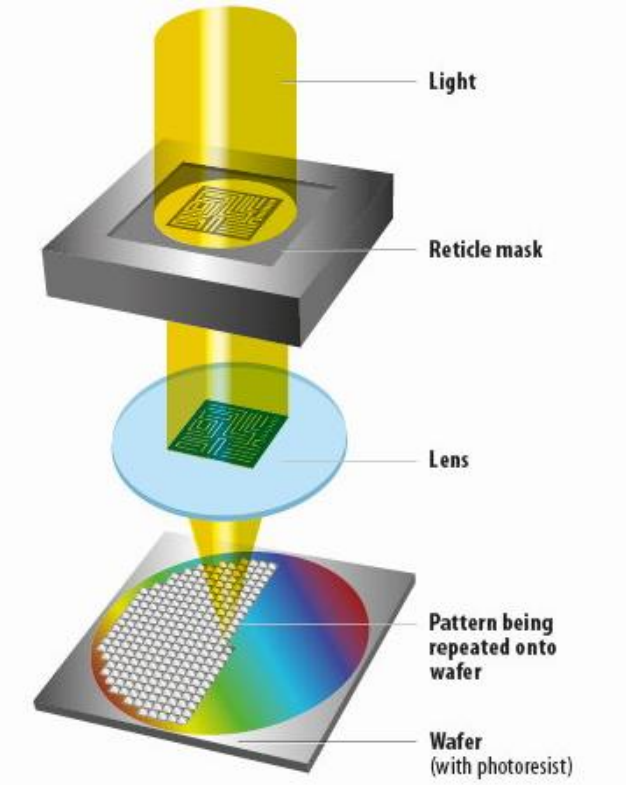
Ruby cutting – Mask making



https://alumni.media.mit.edu/~mcnerney/2009-4004/4004-masks-composite.jpg



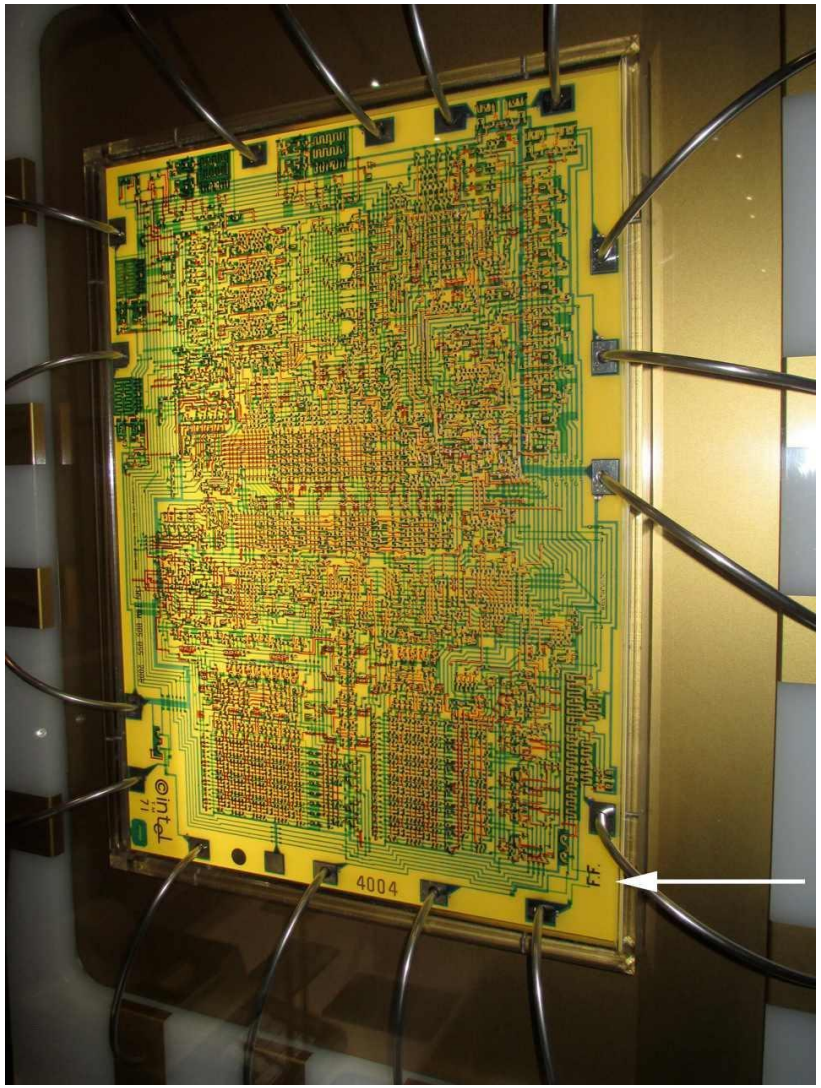
https://semiwiki.com/eda/2036-magic-no-its-computational-lithography/

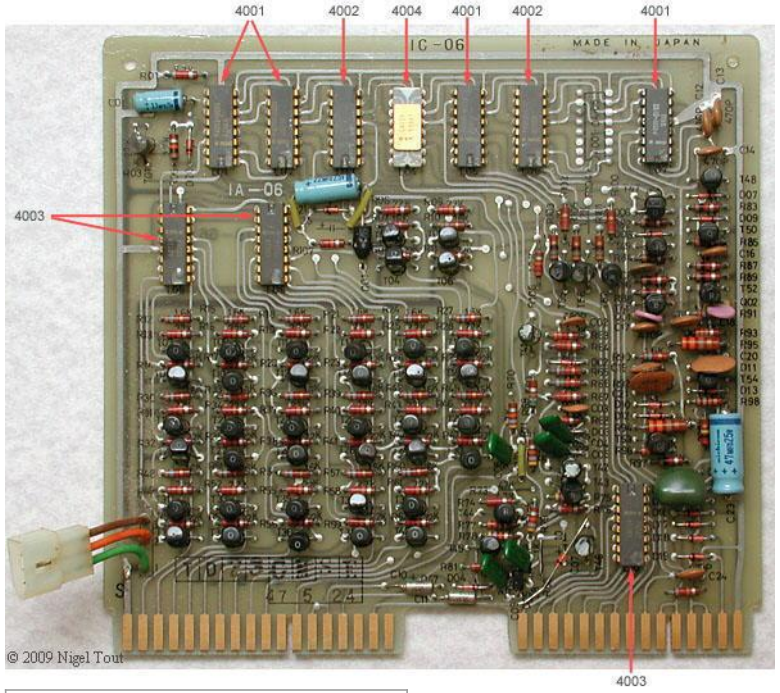


https://bits-chips.com/article/euv-for-dummies/

Chip and product

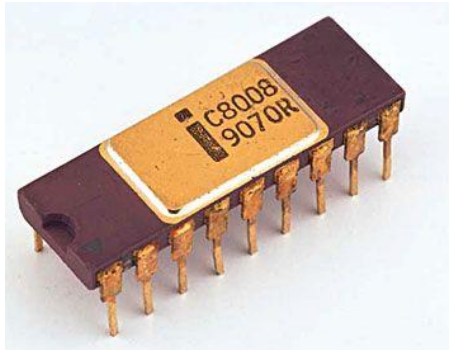
Intel 4000 Family:
4001 - RAM
4002 - ROM
4003 - Shift Reg
4004 - CPU





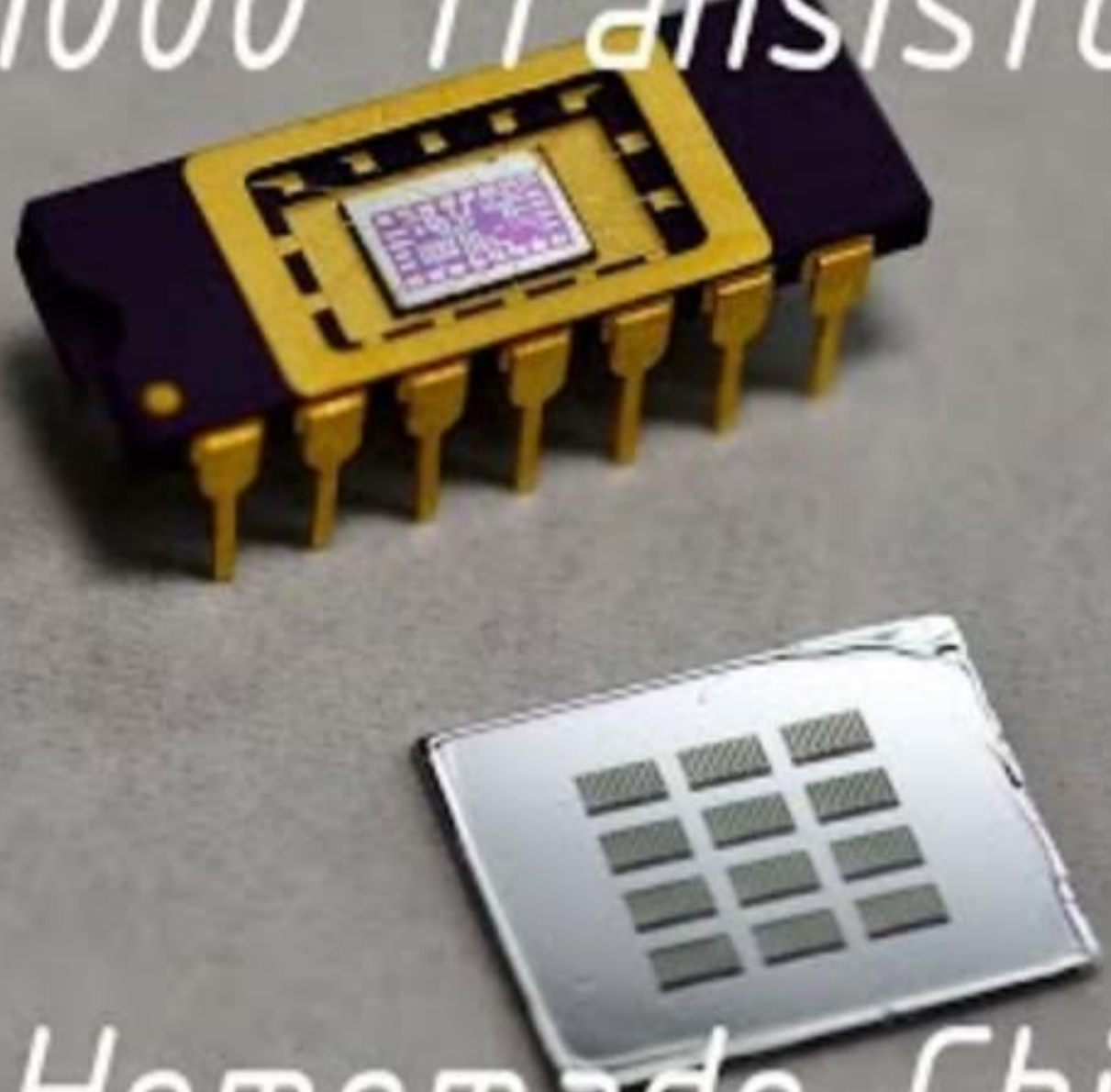
Intel 4004
1971
10 um
2250 transistors
740 kHz
92,000 instructions/s
Simulator





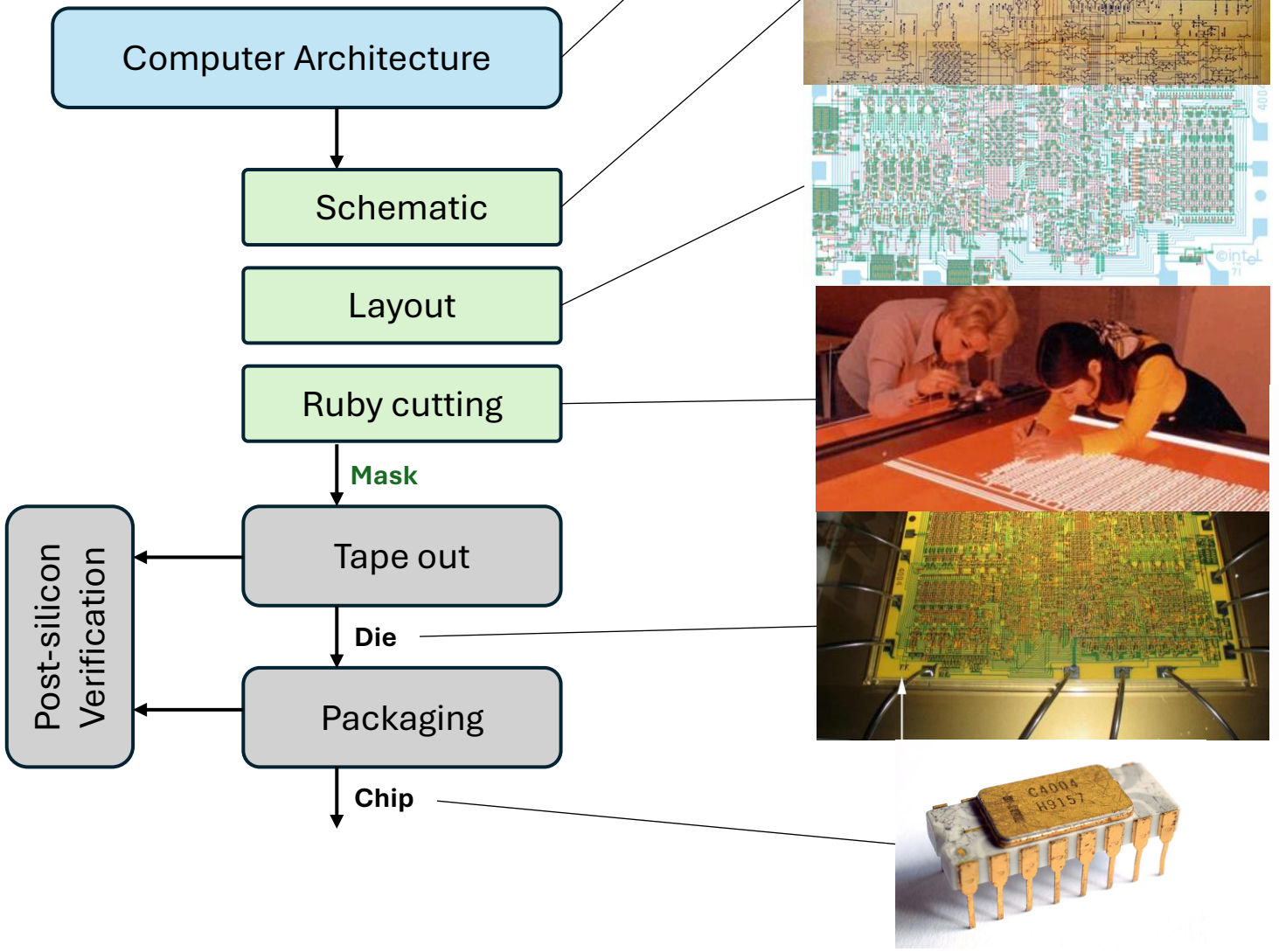
Intel 8008
8-bit
First x86

1000 Transistor



Homemade Chip

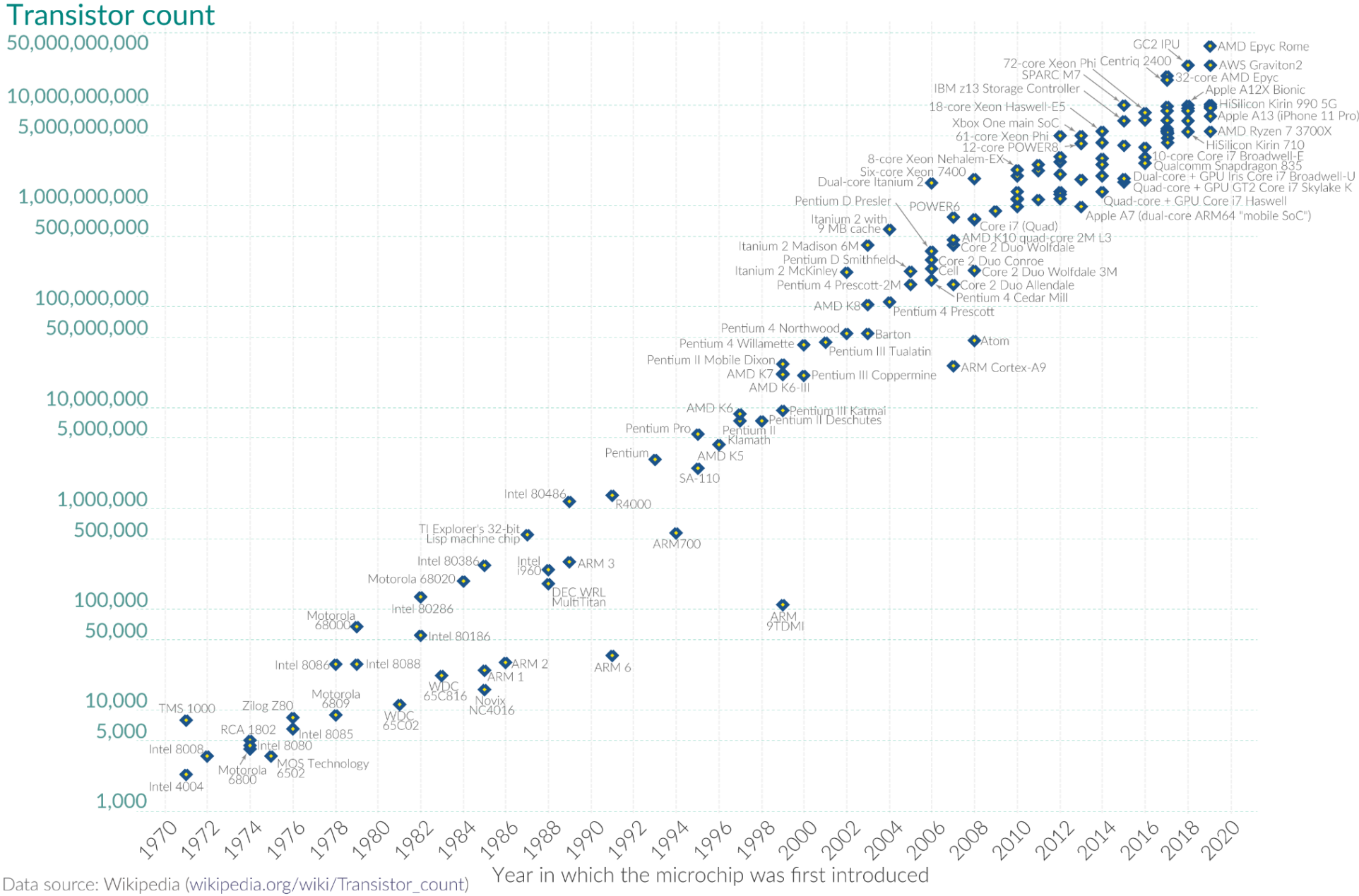
Manual ASIC Flow (1970)



Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World in Data



2,000 transistors in Intel 4004
to
208,000,000,000 transistors
in
NVIDIA Blackwell GPUs

**"Moore's law is dying"
for the past 20 years**

**Regardless, the industry
needs better designers**

Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

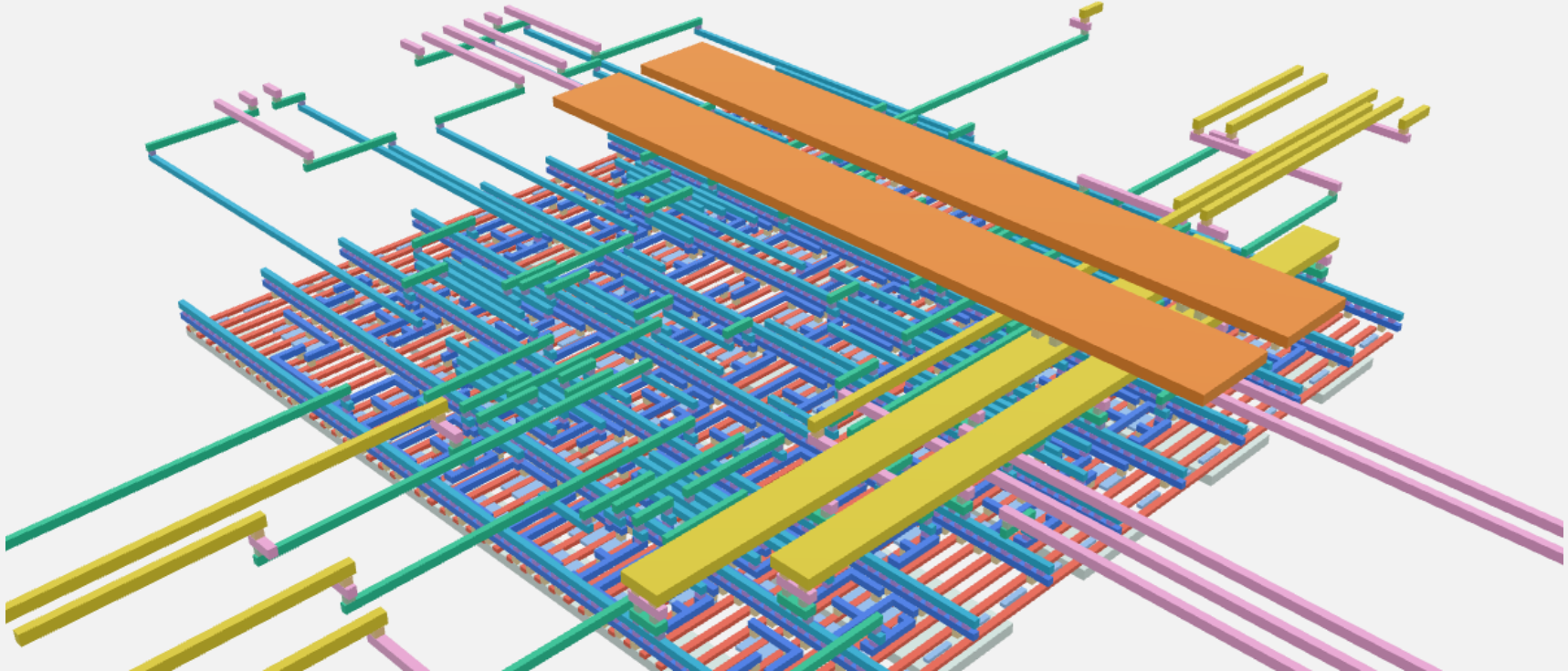
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

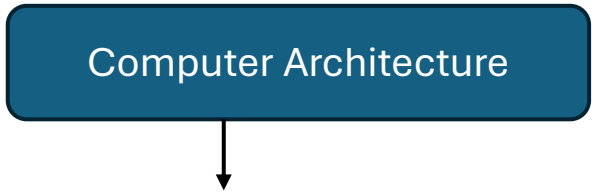
100 million times growth in 50 years!

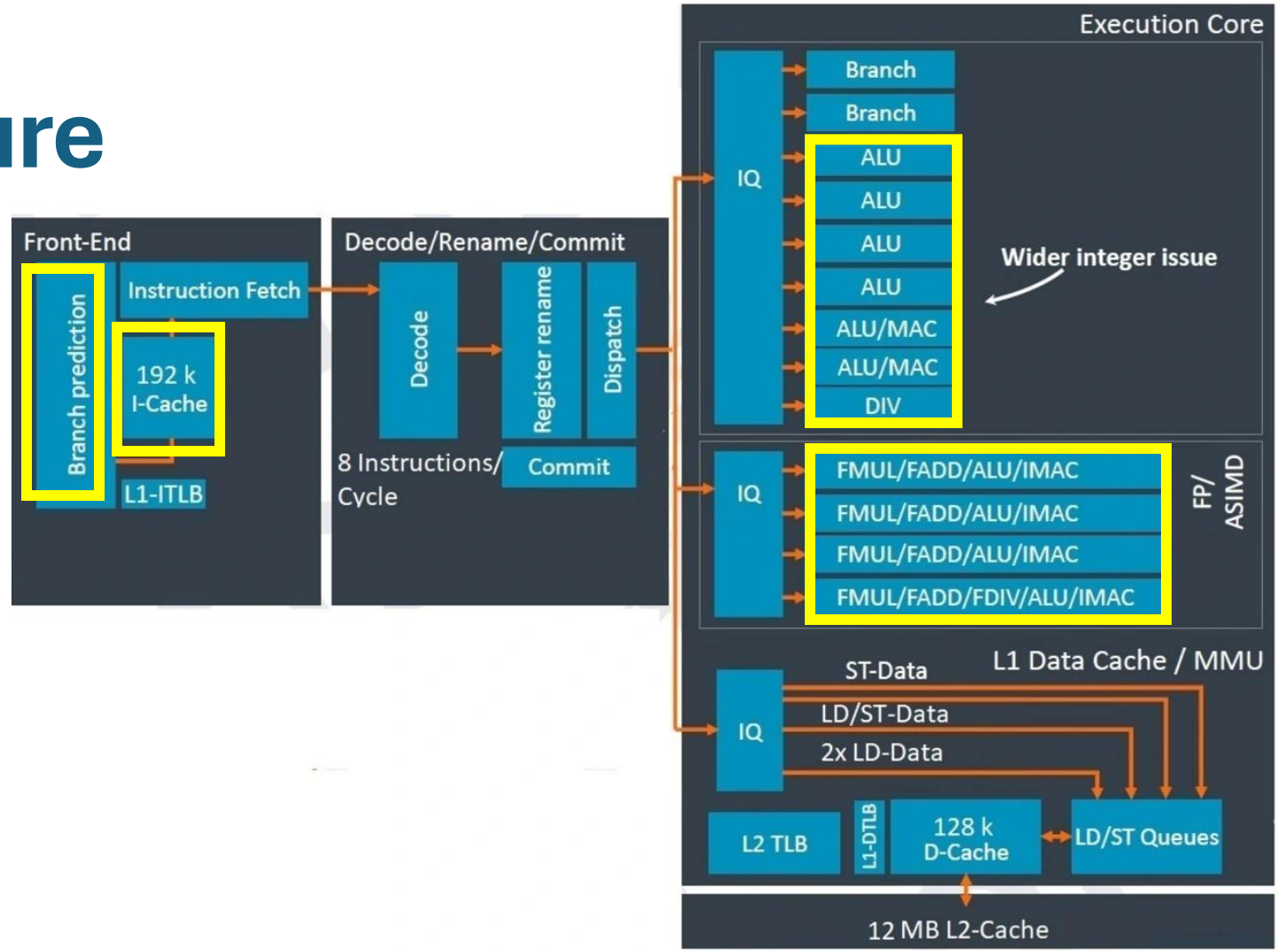
- **If 4004 was a house, today's chips are as big as a country**
- Every year, new chip, millions sold. How?
    1. Abstraction
        - Nobody needs to know everything
        - Can reuse parts of designs on every level
    2. CAD Tools
        - Smart enough to do optimize high level logic
        - Expensive - Since they need to be accurate
    3. Industry Standards
        - SystemVerilog, GDS2, lib, lef…etc

Modern ASIC Flow



Computer Architecture



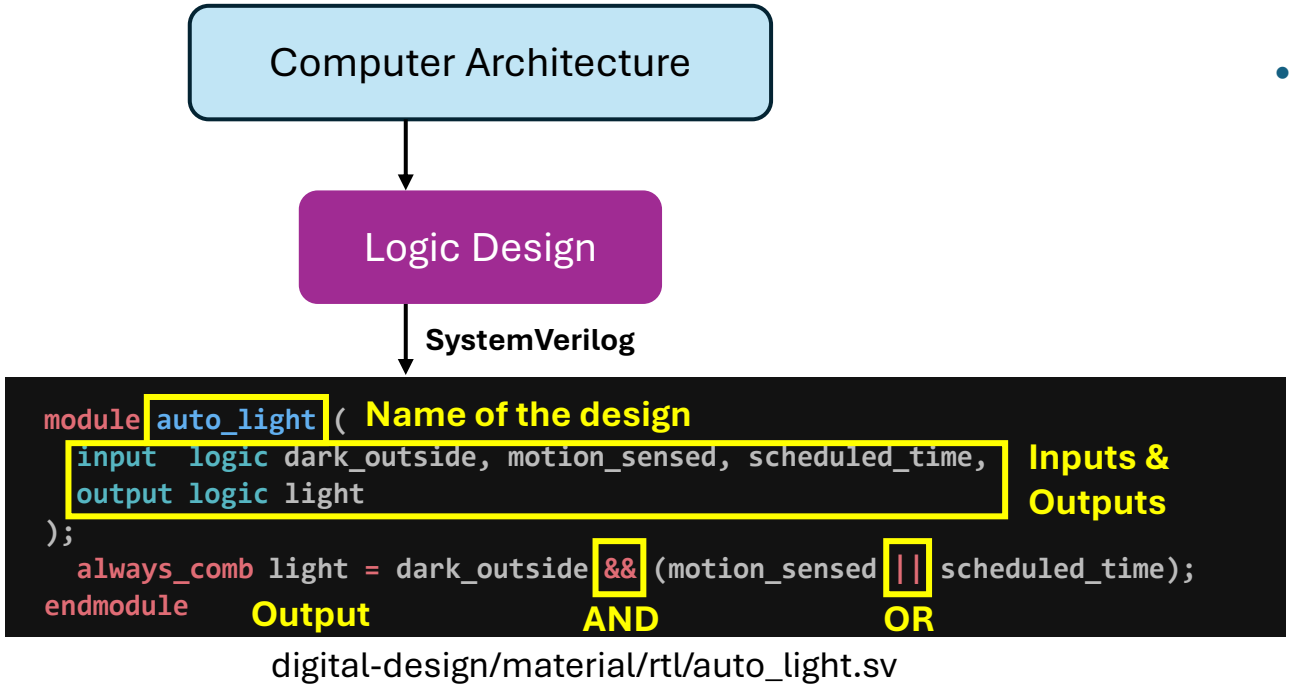


- Highest-level design
- C/C++ models to assess tradeoffs
- CSE 141

Rough representation of
Firestorm Performance core in Apple M1

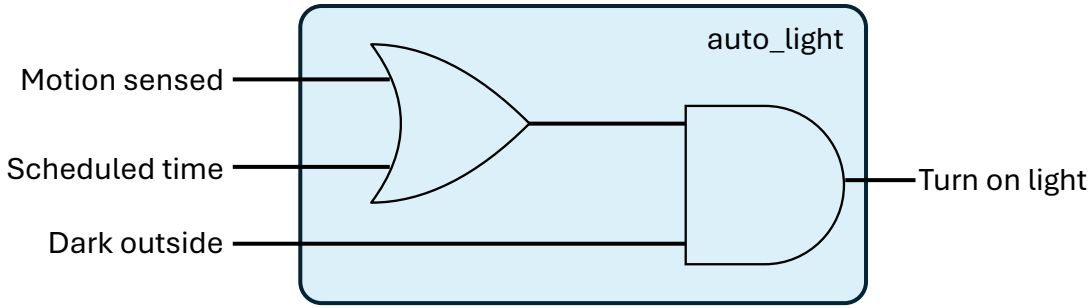
elektroniknet.de/halbleiter/prozessoren/apples-m1-soc-demuetigt-komplette-x86-konkurrenz.7c09a245-d133-4c55-bcc7-4a001c5c4ae9.html

Logic Design



digital-design/material/rtl/auto_light.sv

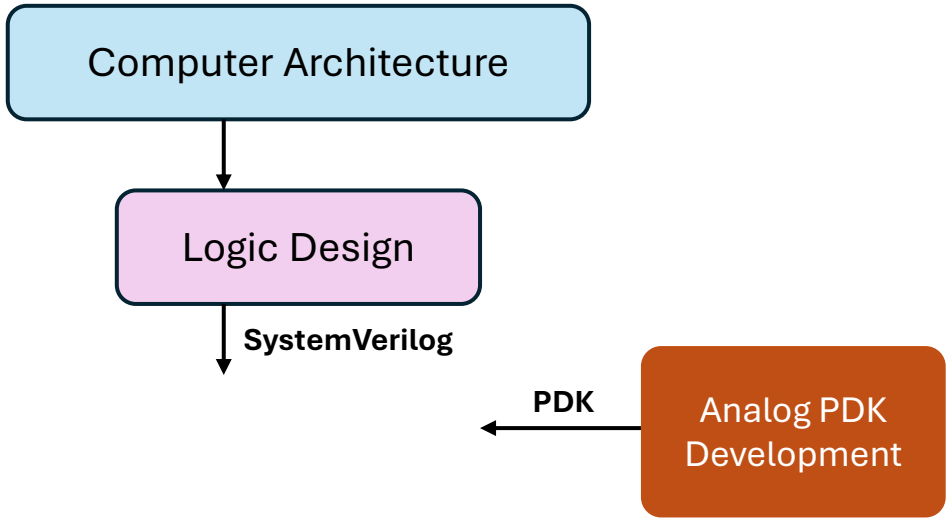
- Each submodule must be designed
- Circuit expressed in text using HDL
- HDL: Hardware Description Language: SystemVerilog
- E.g.: **A chip for an automatic backyard light:**
  - Turn on when dark outside, if any movement sensed, or if scheduled time is now



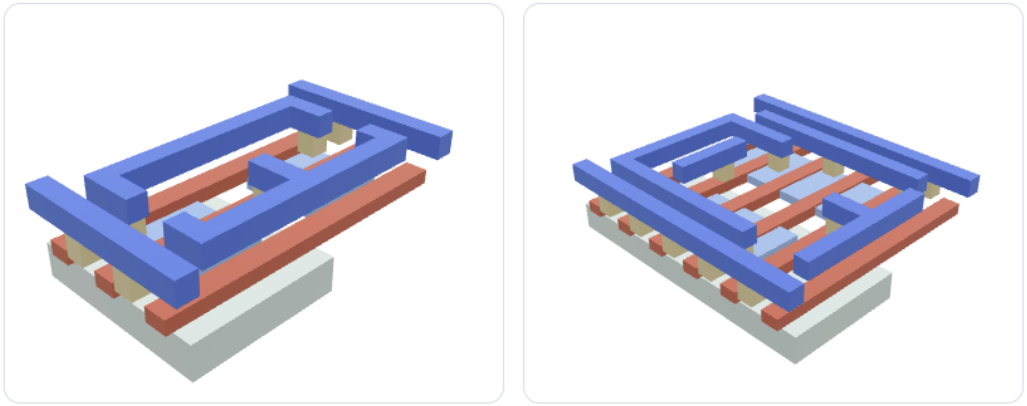
Arithmetic & Logic Unit (ALU) of our CPU

```
case (opcode)
  ADD : regs[i_rd ] <= reg_1 + reg_2;
  SUB : regs[i_rd ] <= reg_1 - reg_2;
  MUL : regs[i_rd ] <= reg_1 * reg_2;
endcase
```

Analog Design (PDK Development)



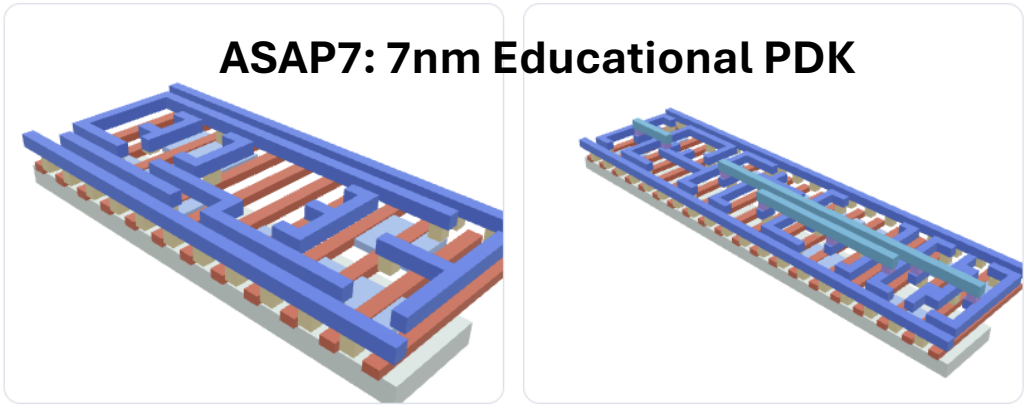
- **PDK**: Process Development Kit
- **Process**: e.g. M1 chip was in TSMC N5 (5nm)
- PDK is a collection of standard cells (Lego blocks)
- More: abapages.com/digital-design/3d-cells



NOT gate (INVx1)

NAND gate (NAND2x1)

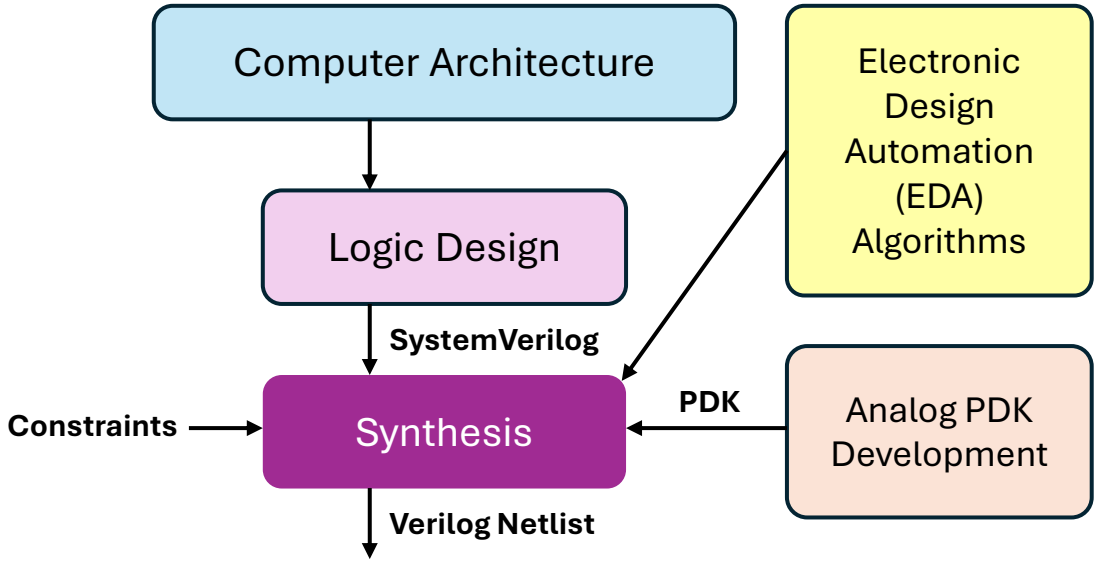
ASAP7: 7nm Educational PDK



AND-OR-INVERT (AOI211x1)

D Flip-Flop (DFFHQNx1)

Synthesis




```
module auto_light(dark_outside, motion_sensed, scheduled_time, light);
  input dark_outside; wire dark_outside;
  input motion_sensed; wire motion_sensed;
  input scheduled_time; wire scheduled_time;
  output light; wire light; wire _0_;
  OR2x2_ASAP7_75t_R _1_ (
    .A(motion_sensed),
    .B(scheduled_time),
    .Y(_0_));
  AND2x4_ASAP7_75t_R _2_ (
    .A(dark_outside),
    .B(_0_),
    .Y(light));
endmodule
```

2- input OR cell from ASAP7

2- input AND cell from ASAP7

digital-design/material/openroad/work/
results/asap7/auto_light/base/1_2_yosys.v

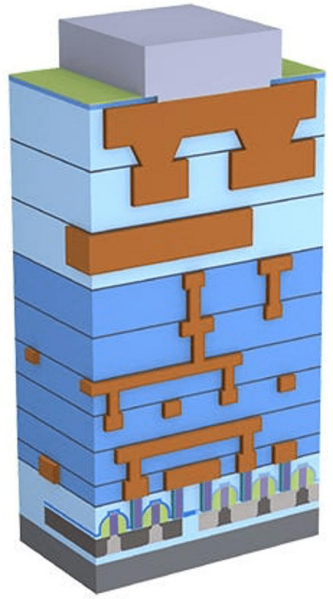
- Synthesis tool (EDA software)
  - Synopsys Design Compiler
  - Cadence Genus
  - Siemens Aprisa
  - Yosys (open source)
- Takes
  - SystemVerilog RTL (high level)
  - PDK (cell library)
  - Constraints: Clock frequency, area…etc.
- **Performs logic minimization**
  - We'll learn on Thursday!
- Maps to the available cells
- Generates a Verilog Netlist

```
make start
make enter
```

```
make DESIGN=auto_light USE_BASIC_GATES=1 gds
make DESIGN=auto_light show_syn_netlist
```

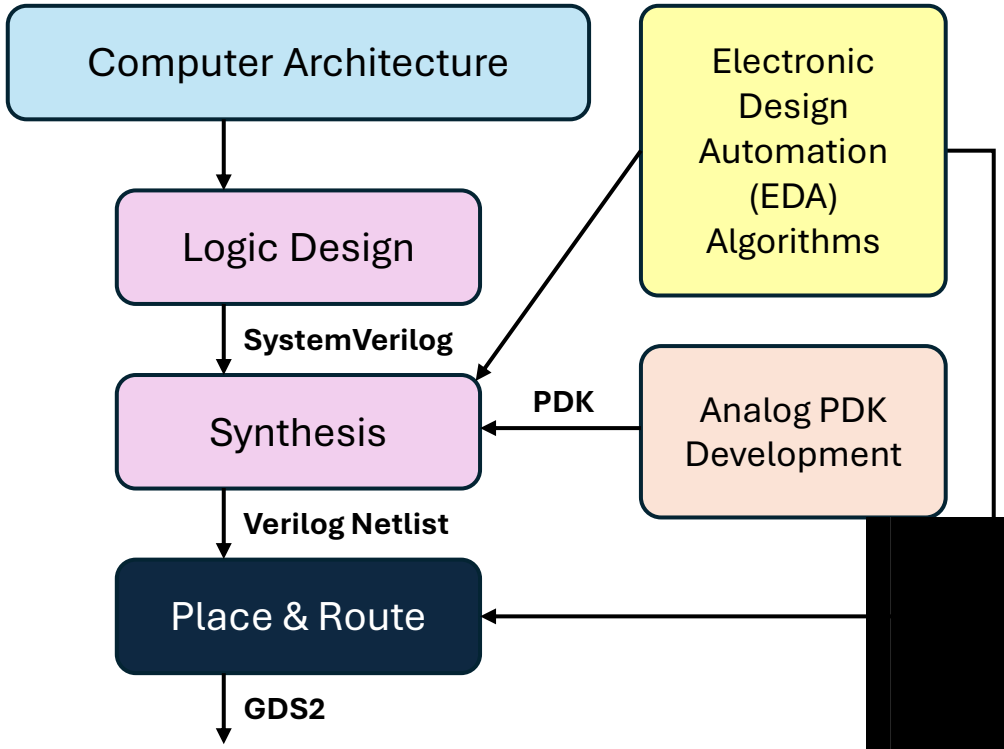
Place & Route

- Decide where to place the std cells
- Connect them with wires in many layers
- Fix any design-rule errors, repeat
- Generate GDS2 – used to make the mask

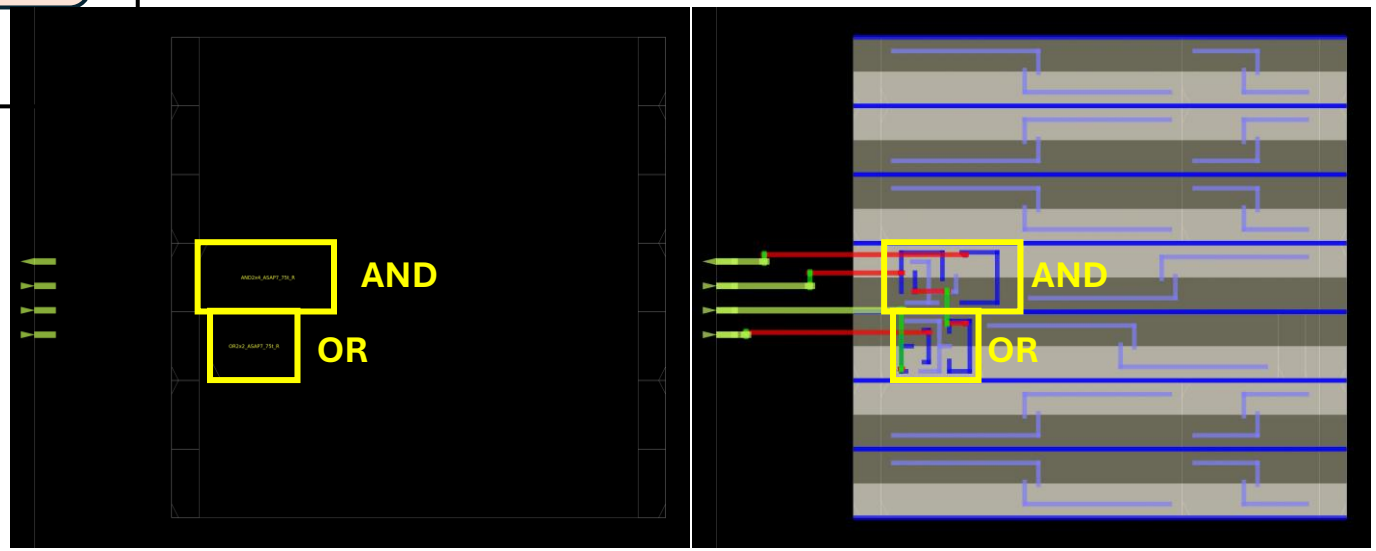


semiengineering.com/interconnects-approach-tipping-point/

Front-end

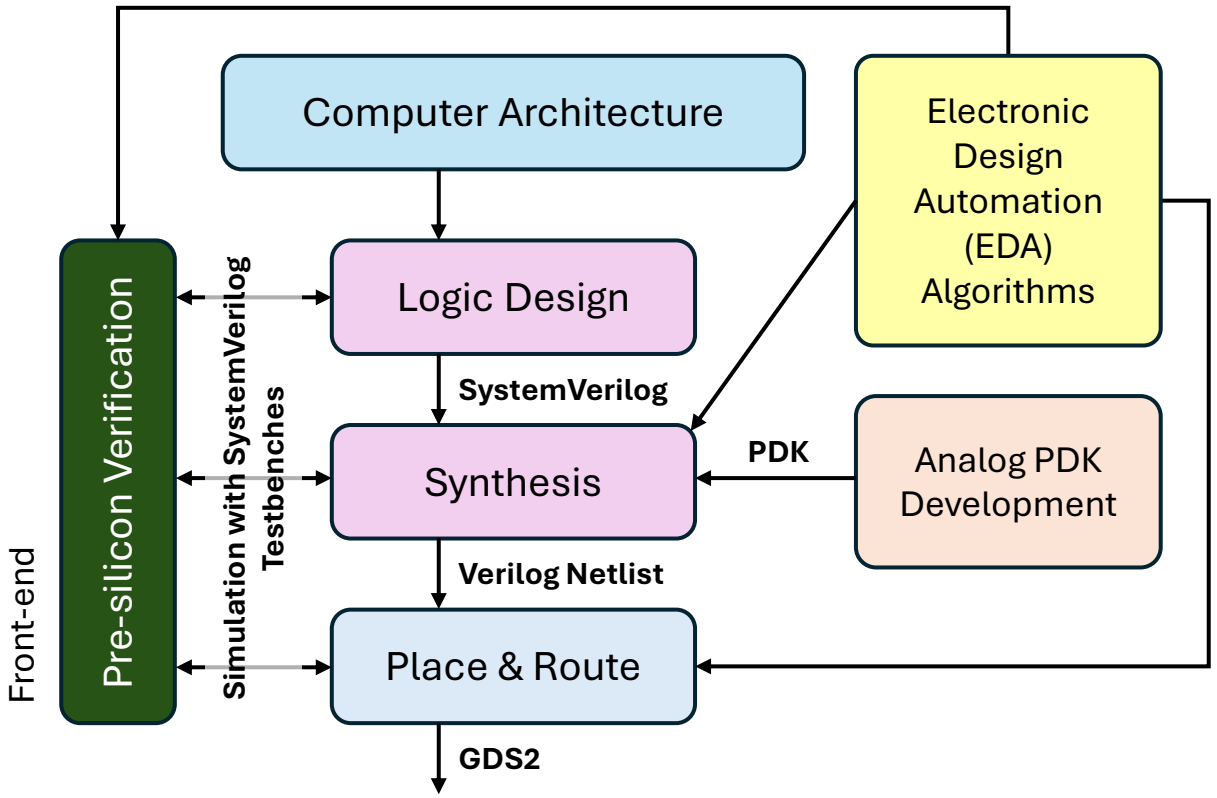


digital-design/material/openroad/work/results/asap7/auto_light/base/6_final.gds



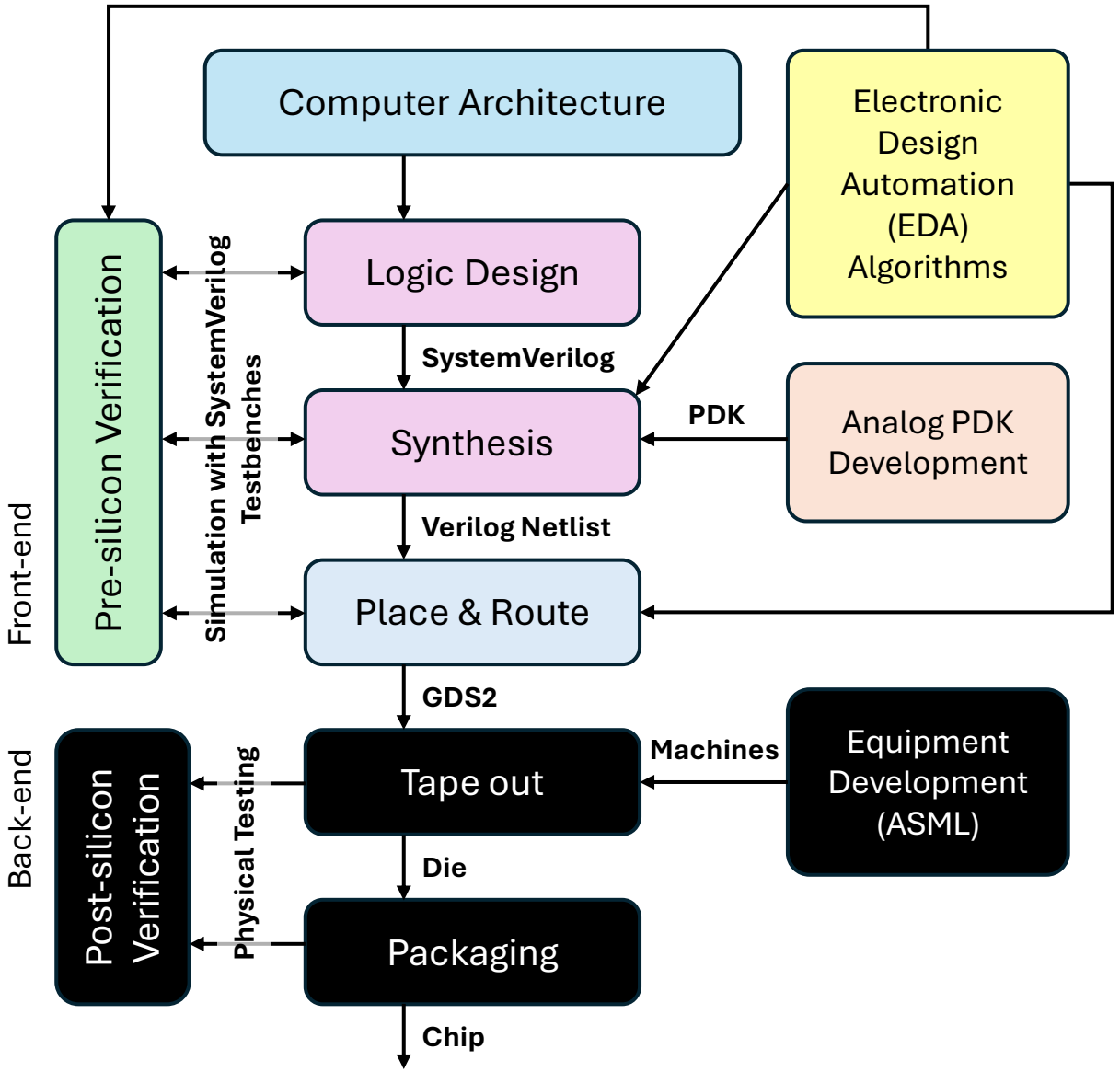
digital-design/material/openroad/work/reports/asap7/auto_light/base/final_routing.webp

Pre-Silicon Verification

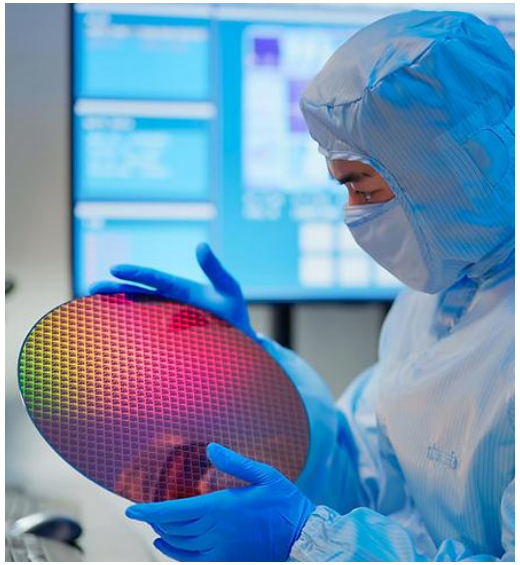


- Testbench – software that
  - Generates randomized inputs
  - Monitors outputs of the design
  - Checks if outputs match expected
- Testbench software is also in SystemVerilog!!
- Design is tested in simulation, at every level
- 5 verification engineers per 1 design engineer

Fabrication, Packaging & Testing



- ASML makes the machines: $150M each
- Masks are made from GDS2 and used as stencil
- Tape-out is done in massive clean rooms (fabs)
  - TSMC, Intel, Global Foundries

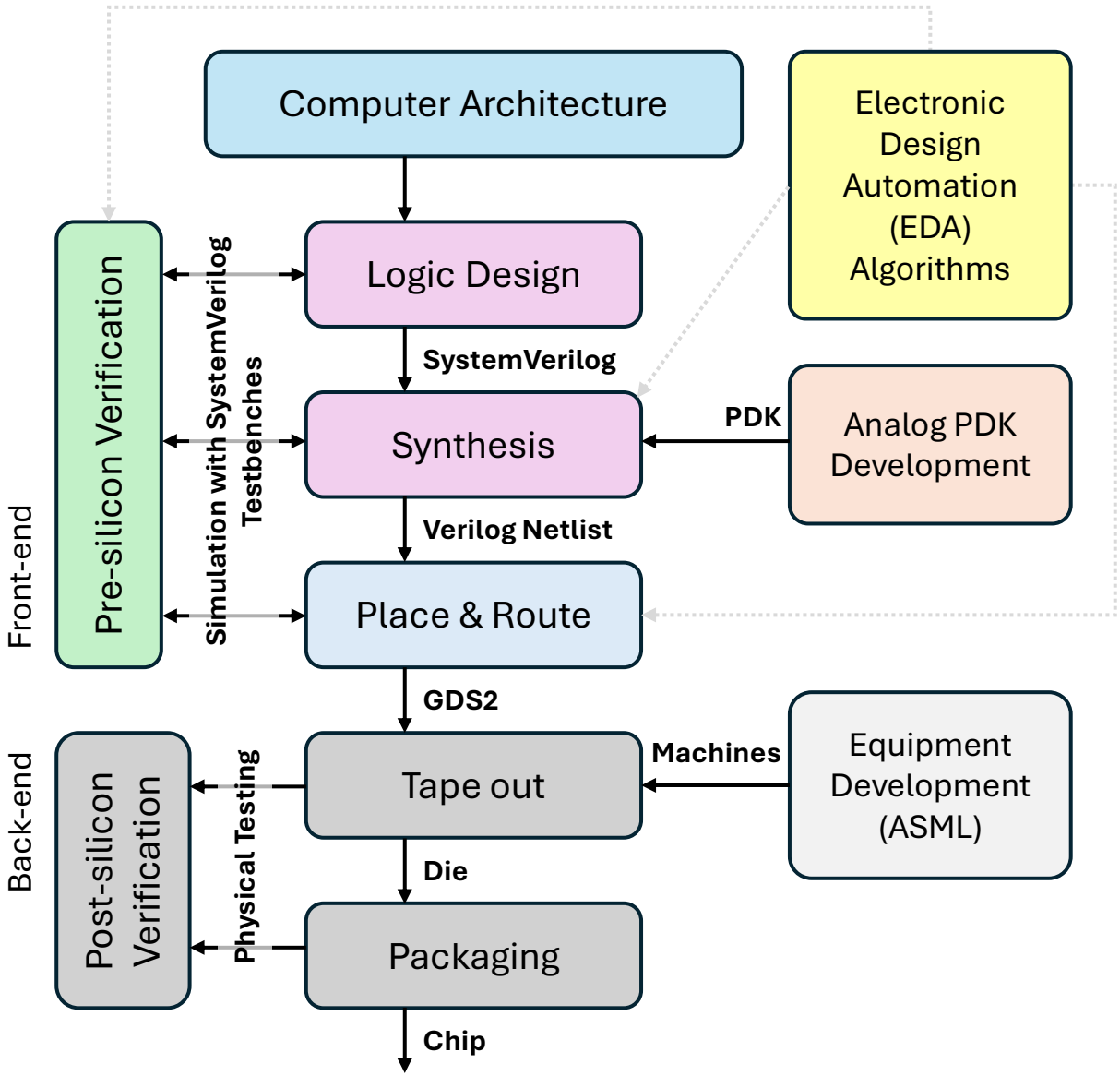


nymi.com/post/enhancing-security-in-semiconductor-wafer-fabrication-a-passwordless-approach-with-the-nymi-band

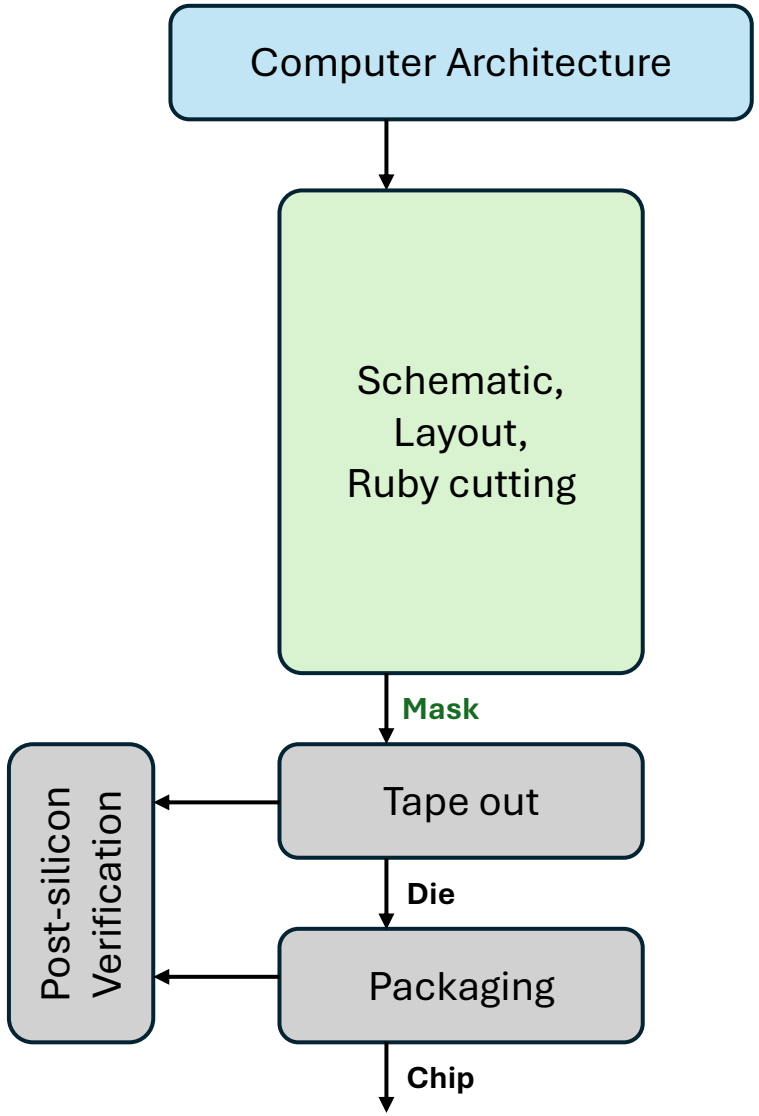


wired.com/story/asml-extreme-ultraviolet-lithography-chips-moores-law/

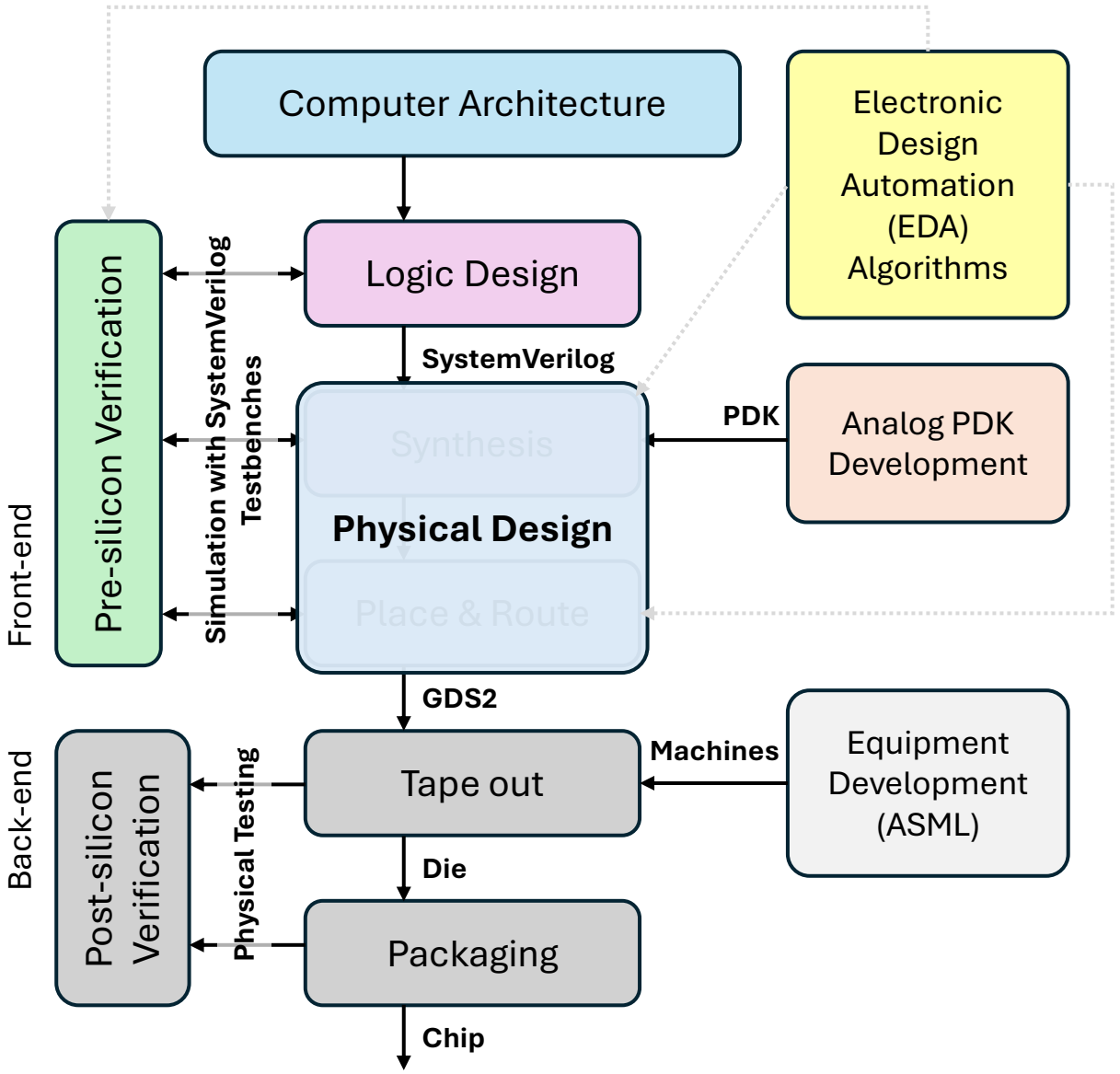
Modern ASIC Flow (simplified)



Manual ASIC Flow (1970s)



A Map of Job Opportunities

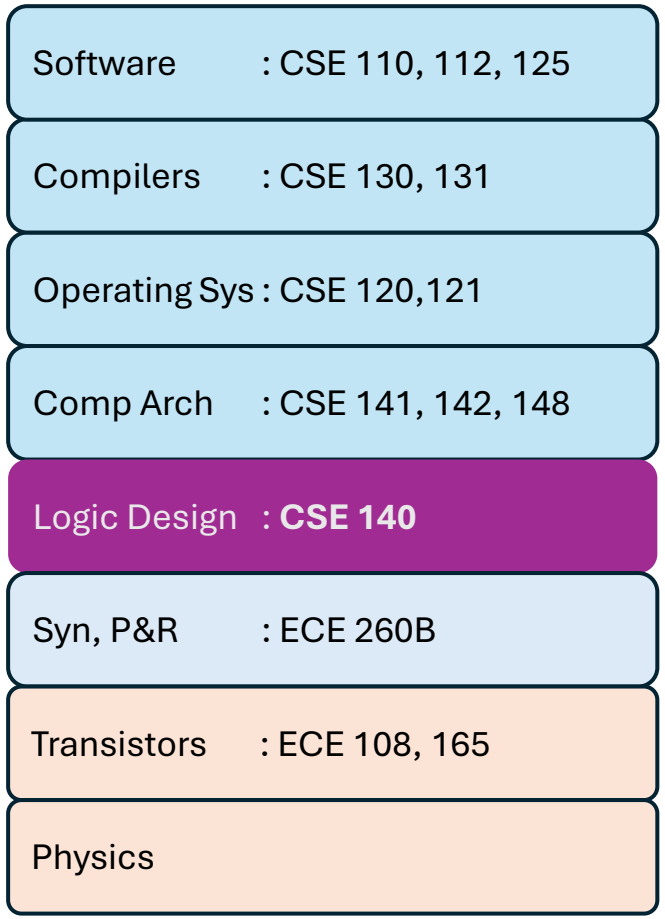


Base salary: 25th–75th percentile range

- Computer architect : $153K-187k
- Logic designer : $149k-190k
- Physical designer : $154k-193k
- Verification engineer : $152k-210k
- PDK designer : $129k-166k
- EDA software developer : $145k-183k
- Packaging engineer : $130k-169k
- Post-silicon testing : $131k-194k
- ASML engineer : $113k-145k

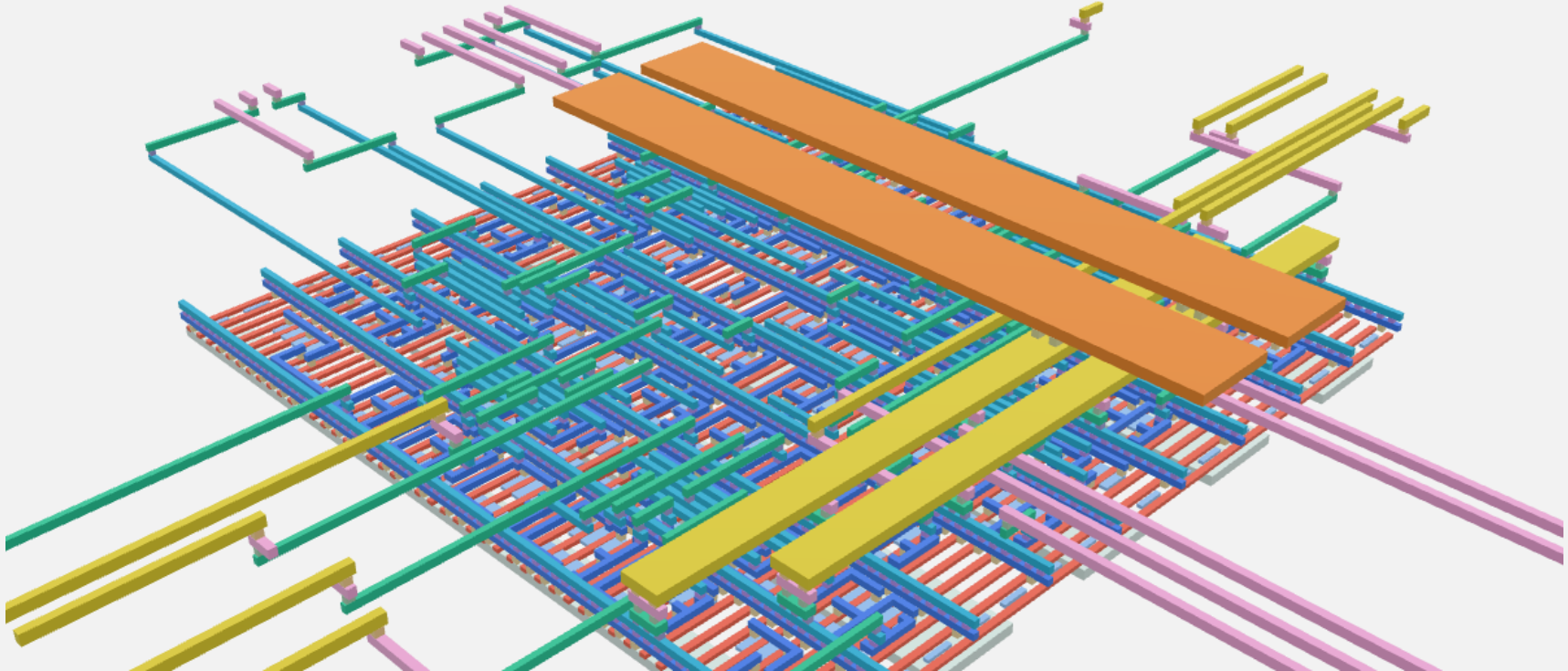
Roughly based on: dol.gov/agencies/eta/foreign-labor/performance

Where CSE140 Fits



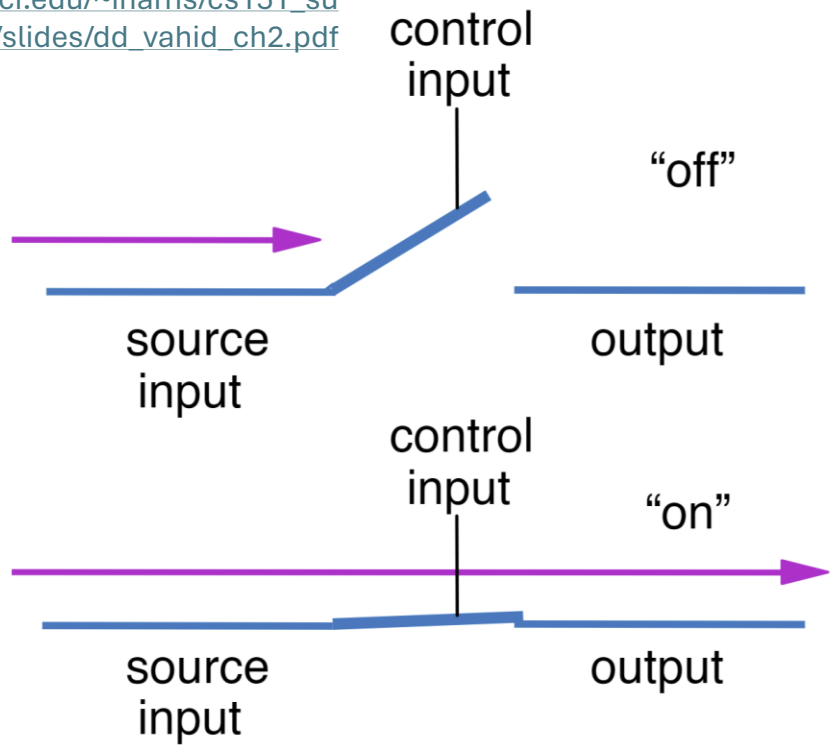
- **Two objectives:**
  - Logic Minimization
    - Given a logic function/ circuit, how to simplify it to use the minimum transistors?
    - Basics of Synthesis
  - RTL Design – Register Transfer Level
    - Designing complex circuits using SystemVerilog
- **We start with AND, OR, NOT gates**
- **End with:**
    - Neural Network Accelerator
    - Tiny CPU
- **Next course:** CSE 141 – Computer Architecture of real CPUs

Transistors to Gates



ics.uci.edu/~iharris/cs151_sum07/slides/dd_vahid_ch2.pdf

Switches



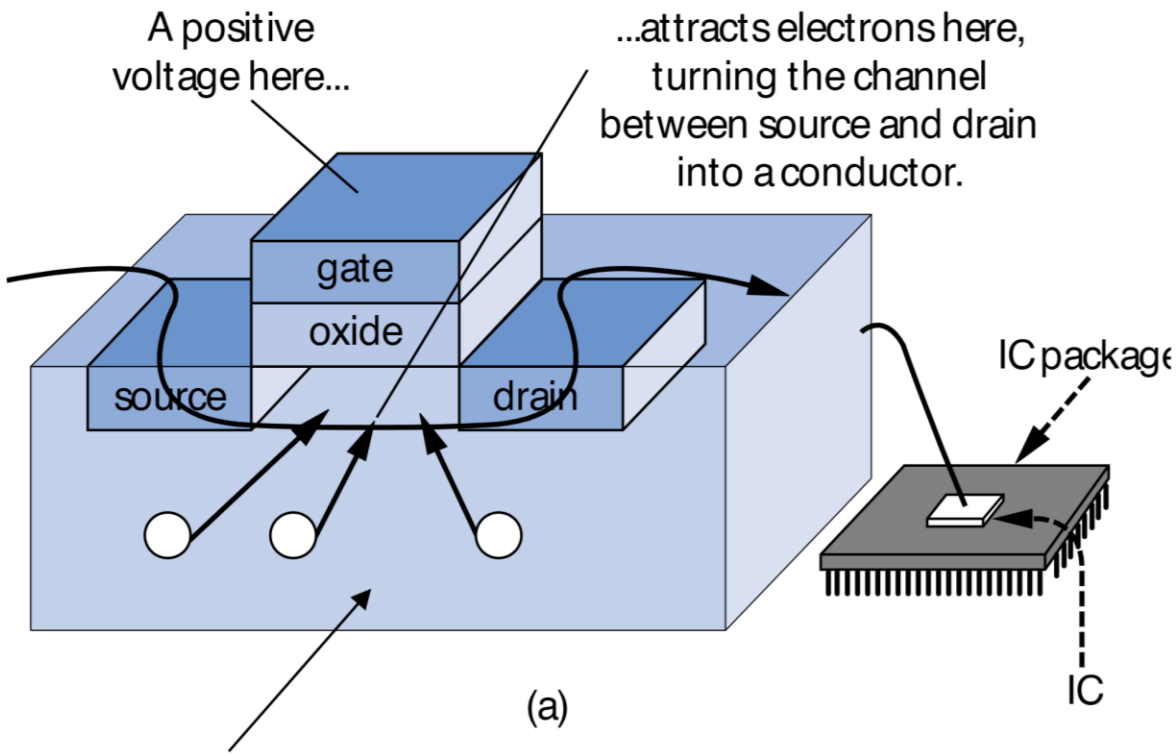
- **A switch has three parts:** source input, control input and output
  - Current wants to flow from **source input** to **output**
  - Voltage at **control input** controls whether that current can flow

- **The amazing shrinking switch**
  - 1930s: Relays
  - 1940s: Vacuum tubes
  - 1950s: Discrete transistor
  - 1960s: Integrated circuits (ICs)
    - Initially just a few transistors on IC
    - Then tens, hundreds, thousands, billions

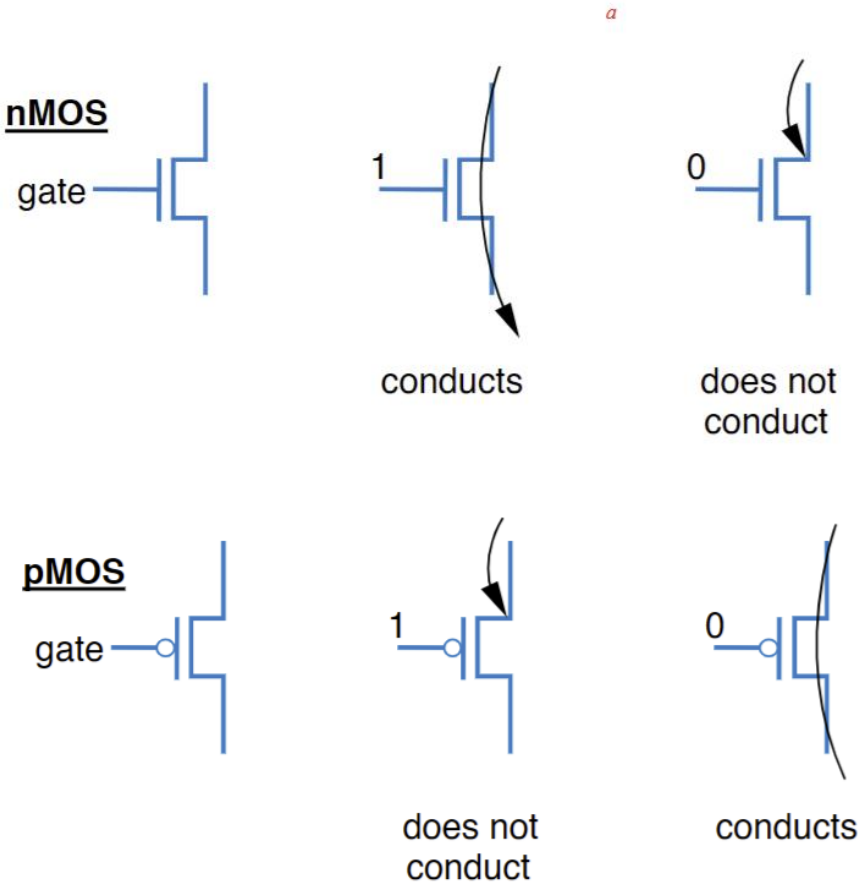


darlingevil.com/hist101-transistors/

CMOS Transistors: The Basic Switches of modern ICs



Silicon -- not quite a conductor or insulator:
Semiconductor



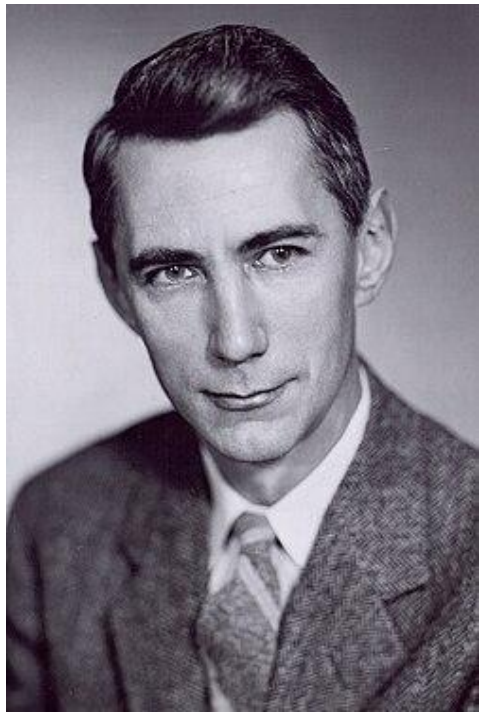
ics.uci.edu/~iharris/cs151_sum07/slides/dd_vahid_ch2.pdf

ics.uci.edu/~iharris/cs151_sum07/slides/dd_vahid_ch2.pdf

Transistors to Gates

- Making circuits with transistors/switches is hard & error prone
- Transistors → Gates → Circuits is much easier
- Why? Boolean Algebra! (Next lecture)
- Claude Shannon's MS thesis, 1937 at MIT

yes!!



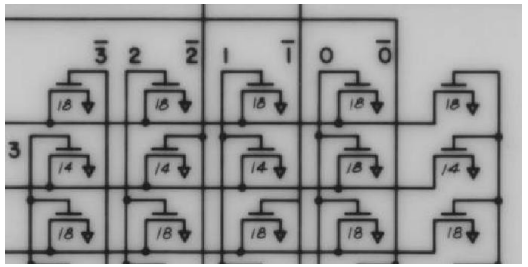


These blocks...



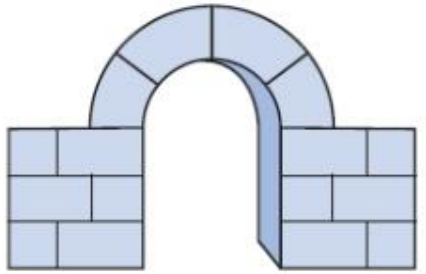
...are hard to work with.



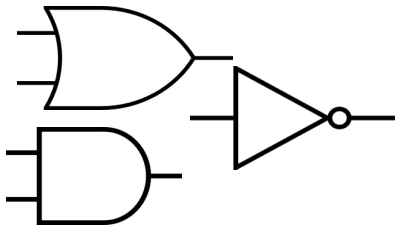


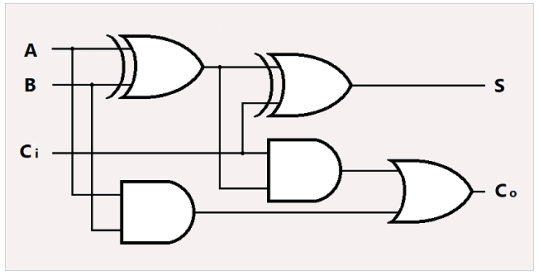


The right building blocks...

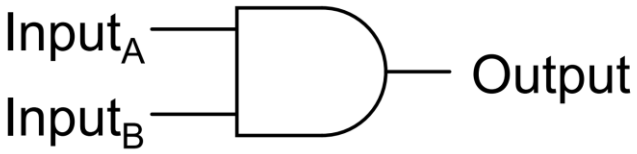


...enable greater designs.



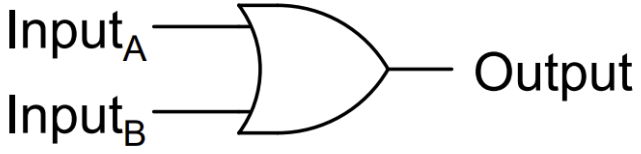


AND, OR, NOT Gates



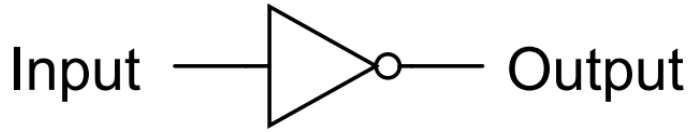
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Logical : **Y = A . B**
SystemVerilog : **Y = A && B**



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

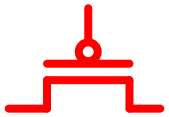
Logical : **Y = A + B**
SystemVerilog : **Y = A || B**



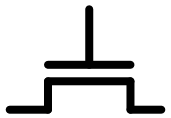
Input	Output
0	1
1	0

Logical : **Y = A'**
SystemVerilog : **Y = !A**

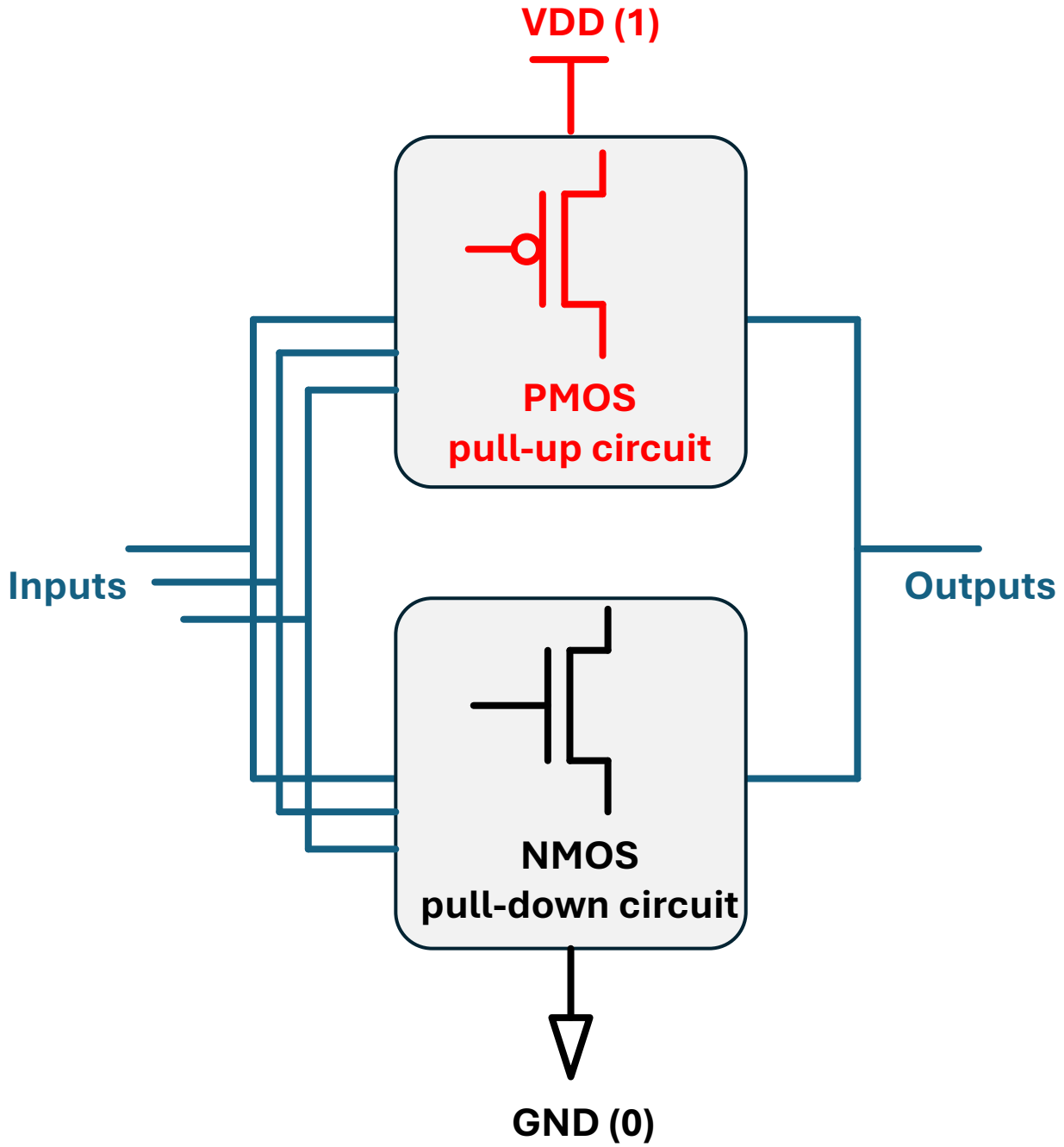
CMOS Circuits

PMOS 

- Turns on when GATE=0 (hence the bubble)
- Good at conducting 1s
- Used as pull-up circuit

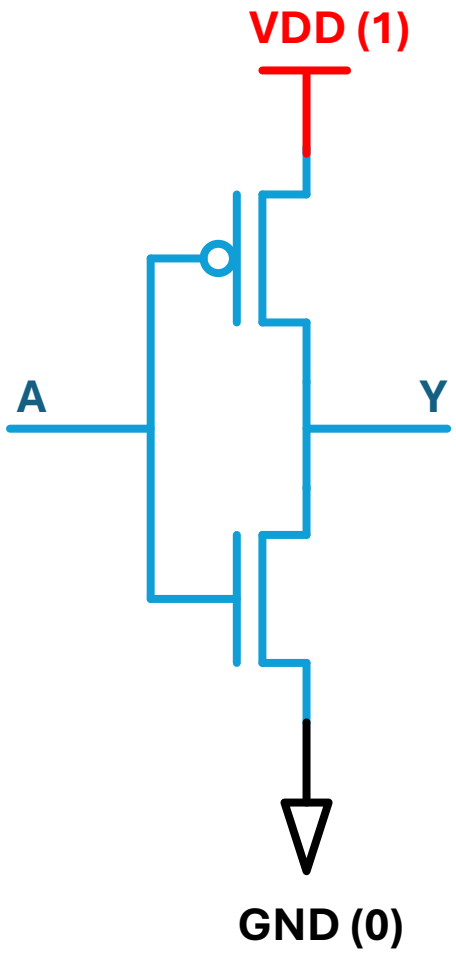
NMOS 

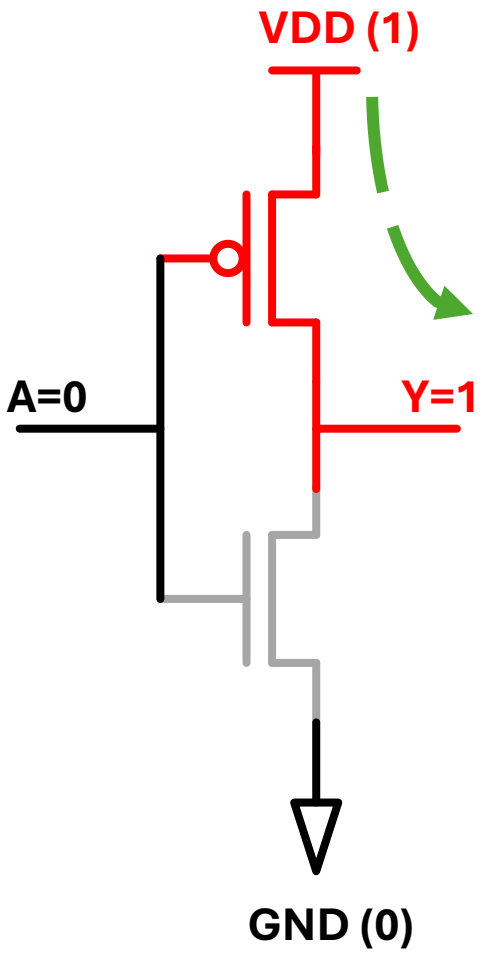
- Turns on when GATE=1
- Good at conducting 0s
- Used as pull-down circuit

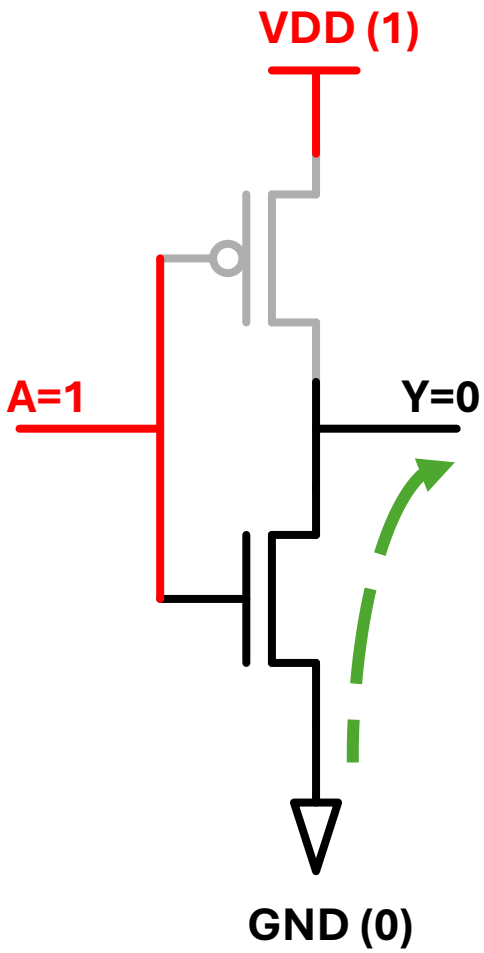


NOT Gate in CMOS

A	Y
0	1
1	0

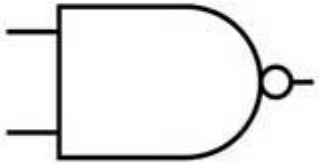




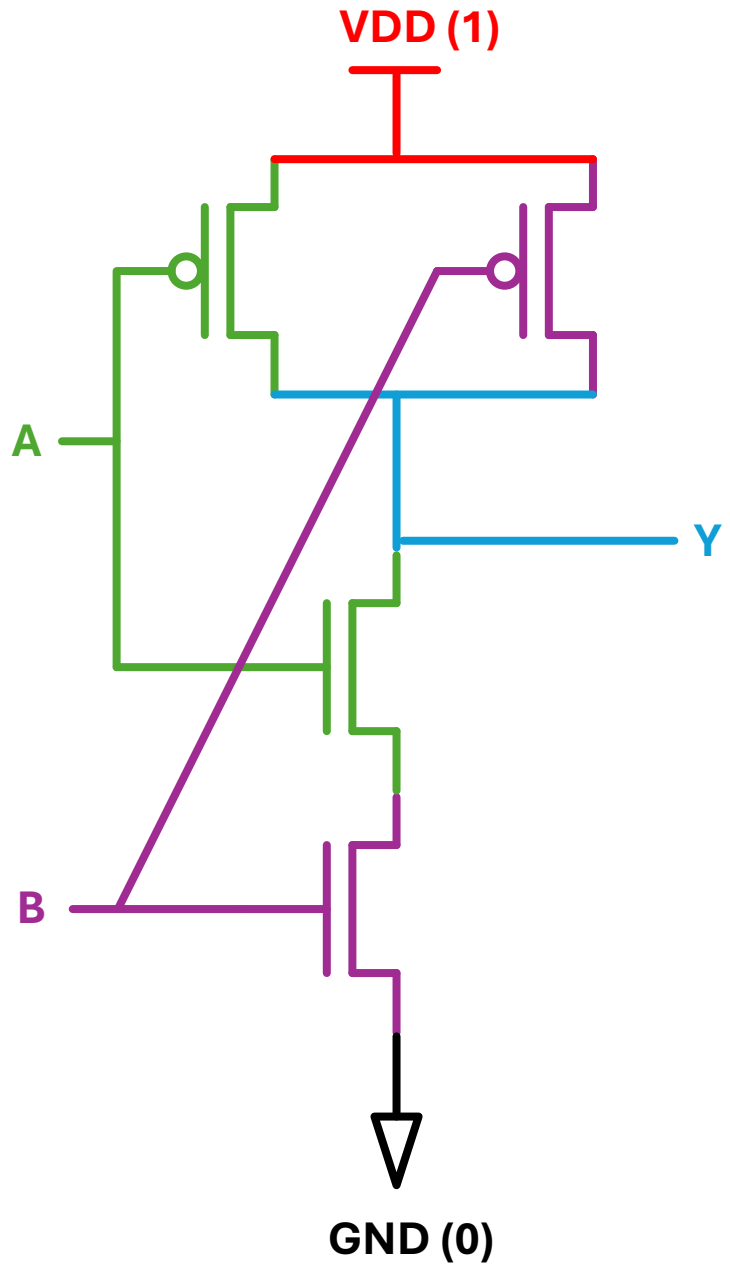


NAND Gate in CMOS

NAND

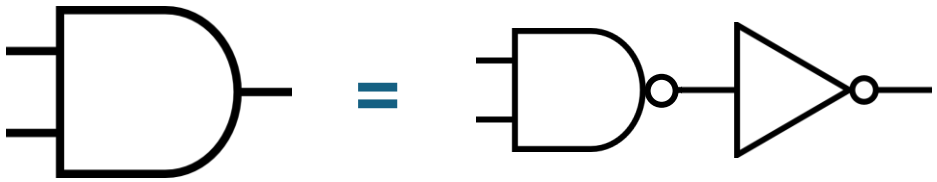


A	B	Output
0	0	1
1	0	1
0	1	1
1	1	0

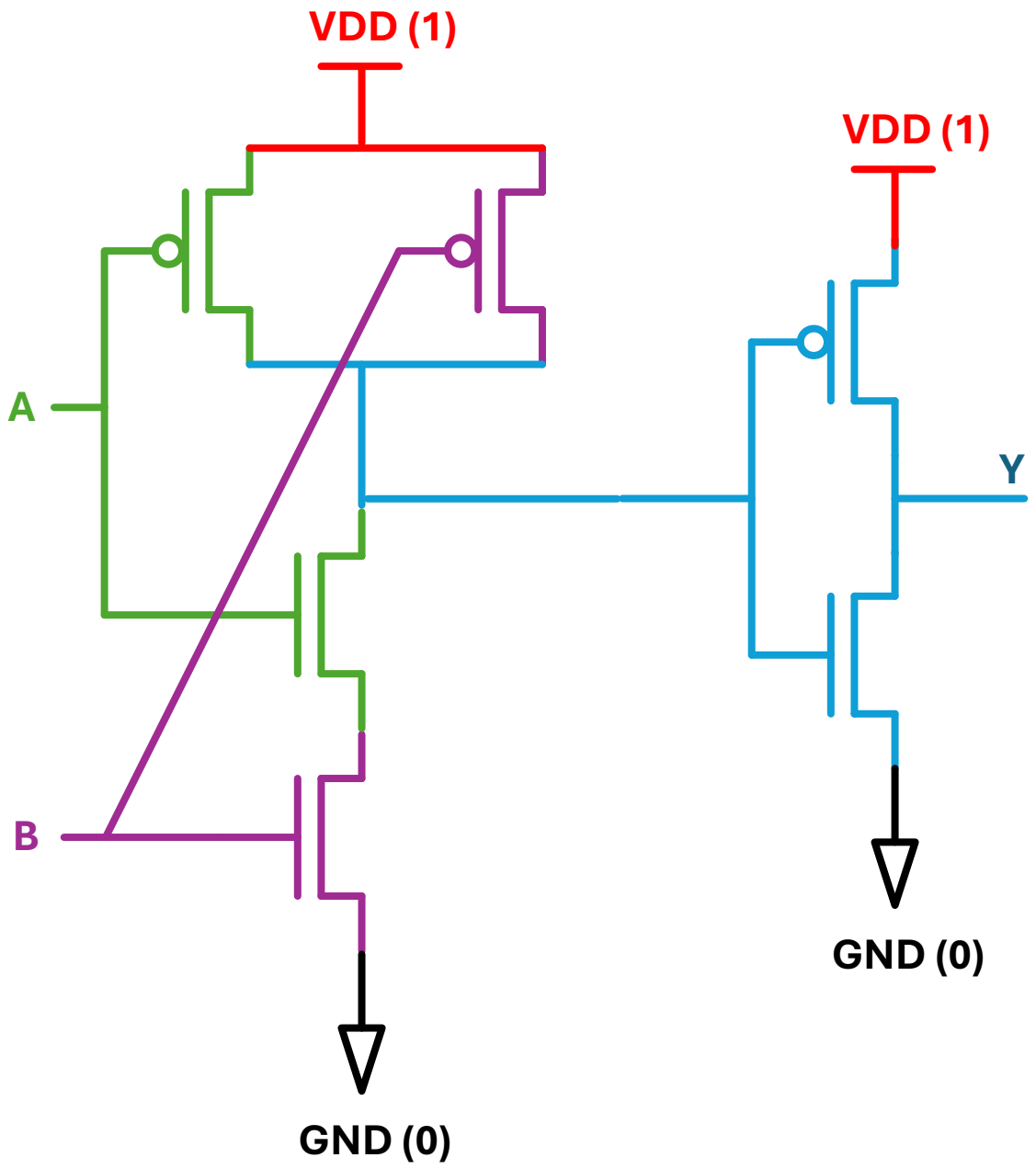


AND Gate in CMOS

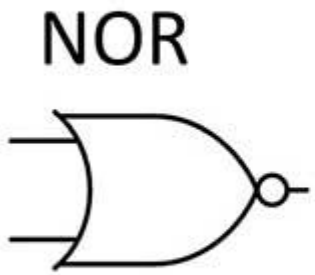
$$A.B = ((A.B)')'$$



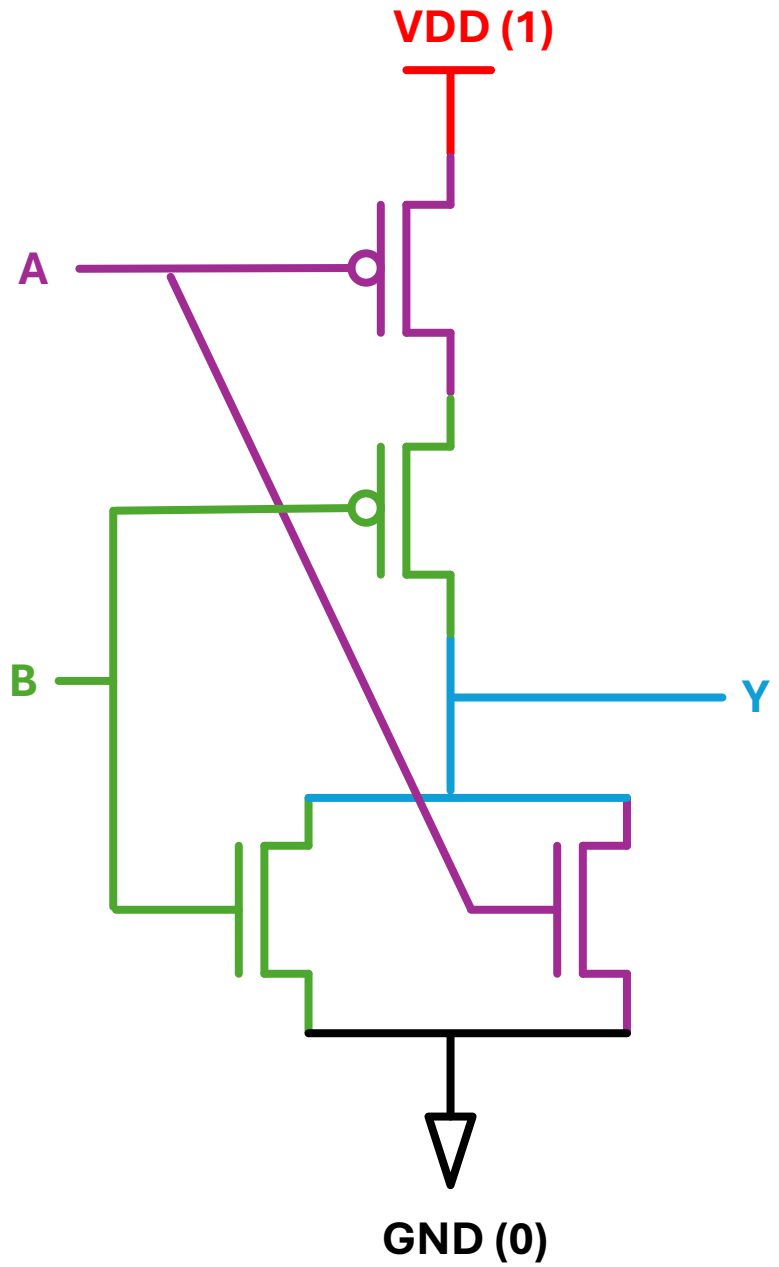
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



NOR Gate in CMOS



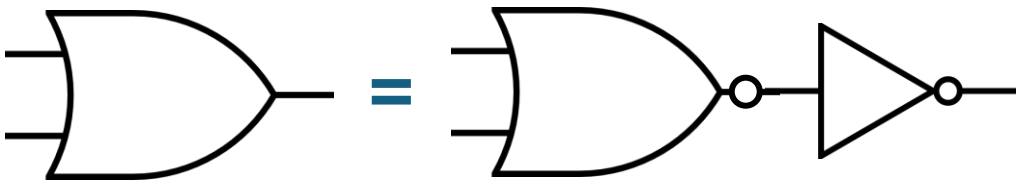
A	B	Output
0	0	1
1	0	0
0	1	0
1	1	0

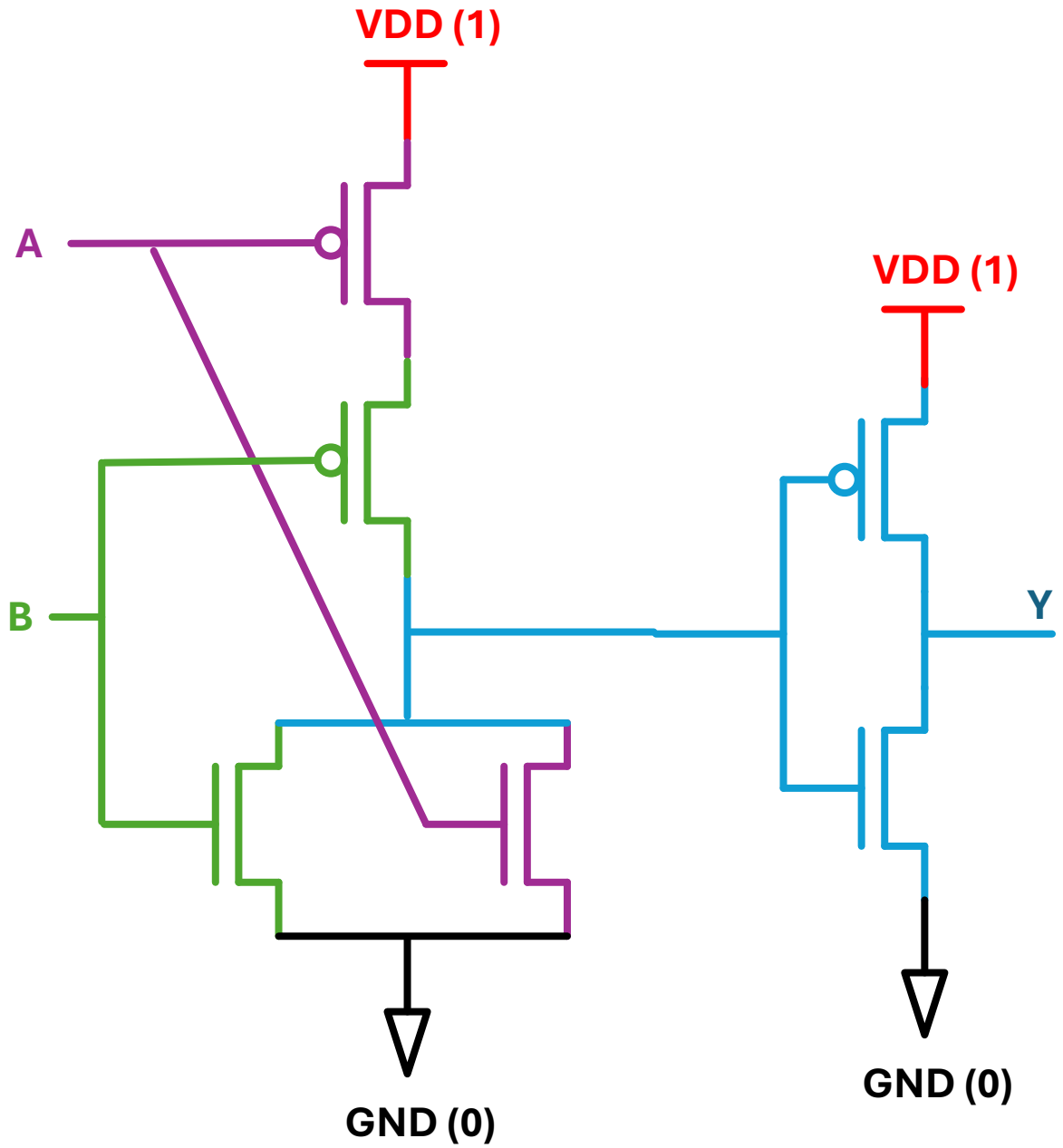


OR Gate in CMOS

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

A+B = ((A+B)')'





Day 1 Feedback

- One thing I liked about today’s class:
- One thing that could be improved:
- One place where I felt lost (if any):