KYUNGPOOK NATIONAL UNIVERSITY

COMP311-1 Logic Circuit Design Chap. 1

Instructor: Taigon Song (송대건)

2019 Fall

Evolution of Computer-aided Digital Design

- Early digital circuits with vacuum tubes
- IC: Integrated circuits in
  - SSI: Small Scale Integration
  - MSI: Medium Scale Integration (hundreds of gates)
  - LSI: Large Scale Integration (> 1000 gates)
  - VLSI: Very-large Scale Integration (> 100,000 transistors)
    - Measurement and verification were impossible to perform by hand
    - Design automation?...
- EDA: Electronic Design Automation
  - Before the era of EDA, everything was done by hand
    - Designing, testing…etc.

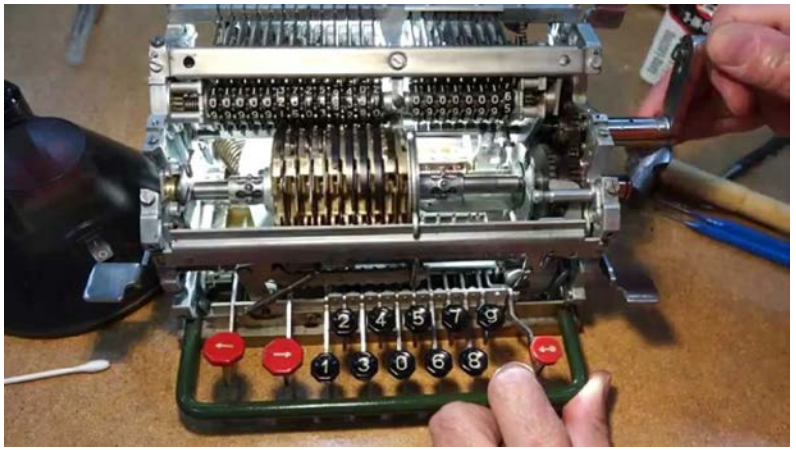
Processing Information

Mechanical? Electronical?

- Mechanical
  - https://www.youtube.com/watch?v=OFJUYFlSYsM

- Electronical

vs.



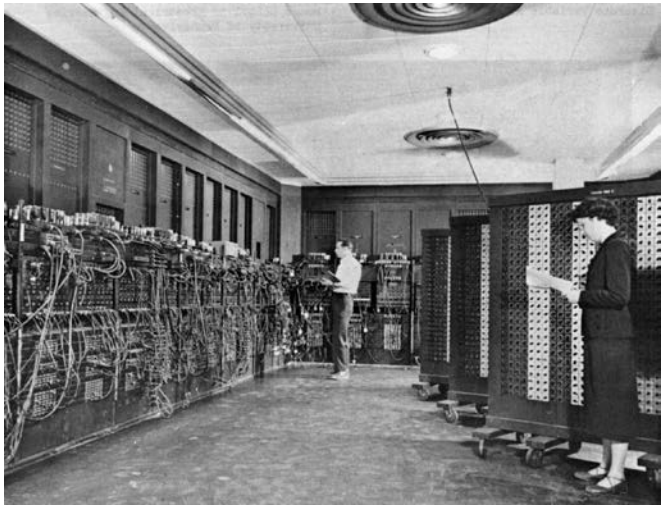
https://www.amazon.com/Calculator-Helect-Standard-Function-Desktop/dp/B01B5MU6JG/ref=sr_1_4?keywords=calculator&qid=1551576516&s=gateway&sr=8-4

https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKEwi7kfHogJvkAhVIQN4KHR9hAR0QjRx6BAgBEAQ&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D5utHg7tG10c&psig=AOvVaw3wgw0SEPoOJw9tWzt7NCs2&ust=1566718877232446

Note. This lecture is not connected to the product by any means and is describing this product solely for educational purpose.

ENIAC – The First Computer

- **Specs**
  - **Width :** 1m
  - **Height:** 2.5m
  - **Length :** 25m
  - **Weight :** ~ 30t
  - **# of Vacuum tubes :** ~18,000
  - **Power :** 1,500kWh

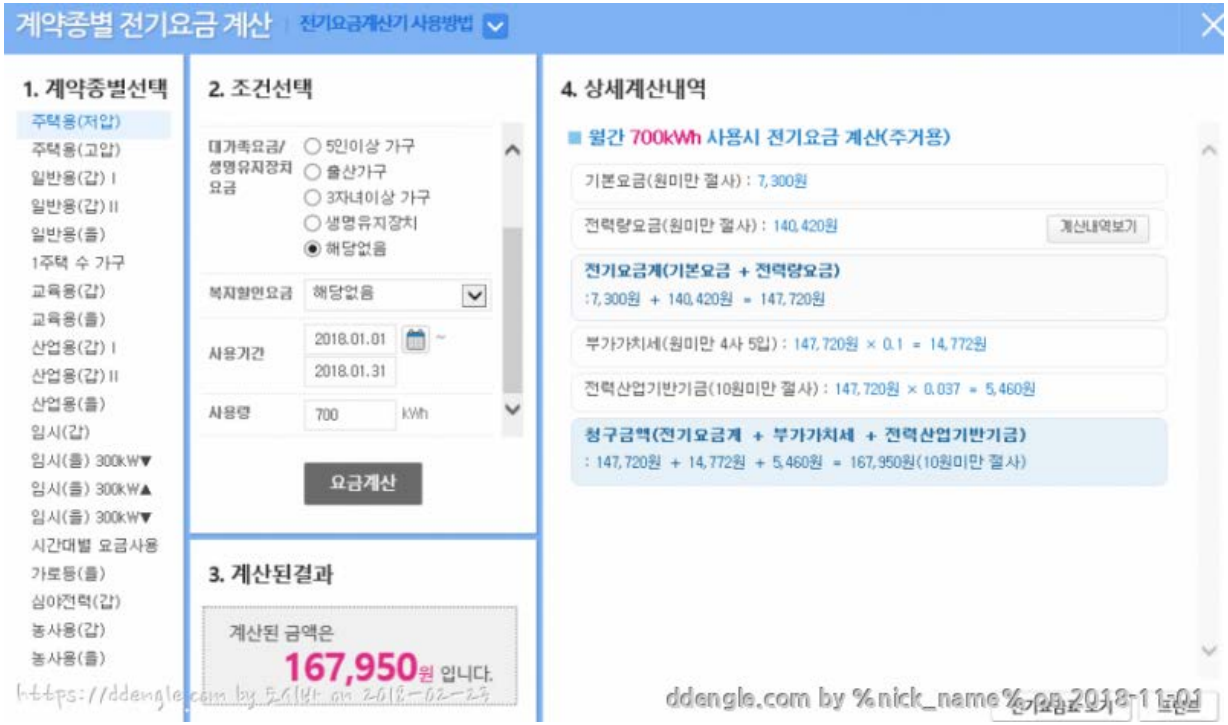


- LG Gram 14Z990-GA5IK
  - Size: 323x212x16.5mm
  - Weight: ~995g
  - Power: 10.7kWh

Sources:
https://ko.wikipedia.org/wiki/%EC%97%90%EB%8B%88%EC%95%85
http://prod.danawa.com/info/?pcode=7971001&cate=112758

How Expensive is Electricity?

- Note: ENIAC's power consumption: ***1500kWh***
  - At least 300k won per hour



Source:
https://www.ddengle.com/mining/6285565

CPU Design in the Early Days

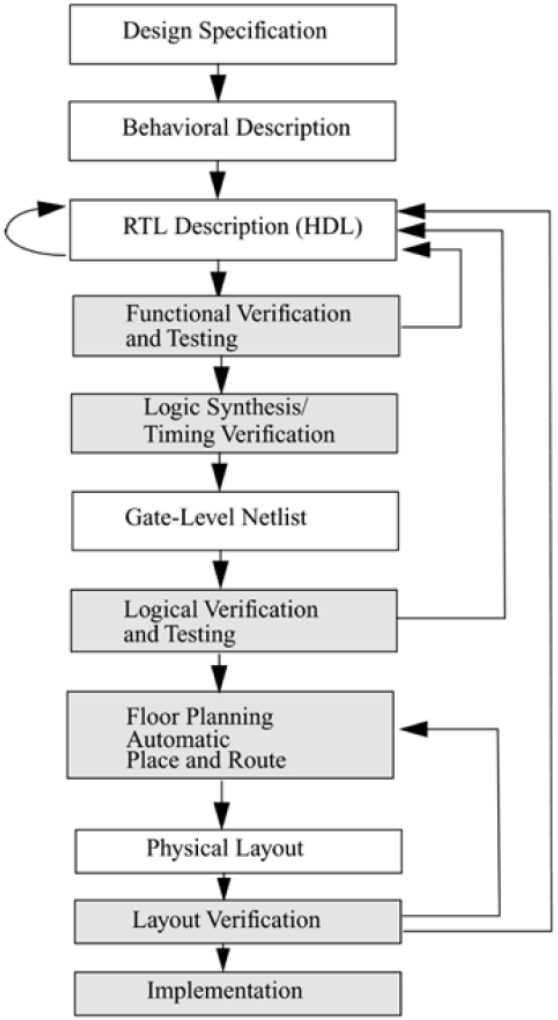
- An Intel 4004 processor example
  - http://www.intel4004.com/4004_original_schematics.htm

Emergence of HDLs

- Sequential languages to describe computer programs
- Hardware Description Languages (HDL) for concurrent hardware designs
  - Verilog HDL from Gateway Design Automation (1983)
    - Acquired by Cadence Design Systems (www.cadence.com)
- Concept of logic synthesis invented (1980s)
  - Design Compiler by Synopsys Inc. (www.synopsys.com)
  - Era of RTLs (register transfer level)
    - Source code for hardware design
- HDL now used for system-level design
  - FPGA (Field Programmable Gate Arrays), PAL (Programmable Array Logic) becomes useful for various purposes
- Current Verilog HDL standard
  - IEEE1364-2001

Typical Design Flow

- For what?
  - Think what makes "***$$$***"
    in the semiconductor business



Silicon Design and Verification

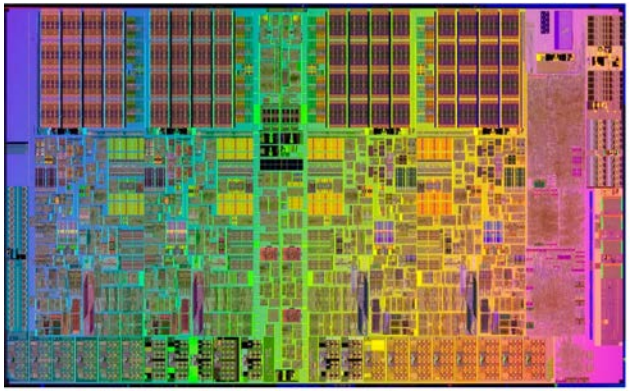
How IC designs are done

- An RTL-to-GDSII flow

```
Module intel_i7 (clk, input_a, input_b, … output_z);
  input clk, input_a;
  output output_a, output_b;
…

endmodule
```

RTL



GDSII

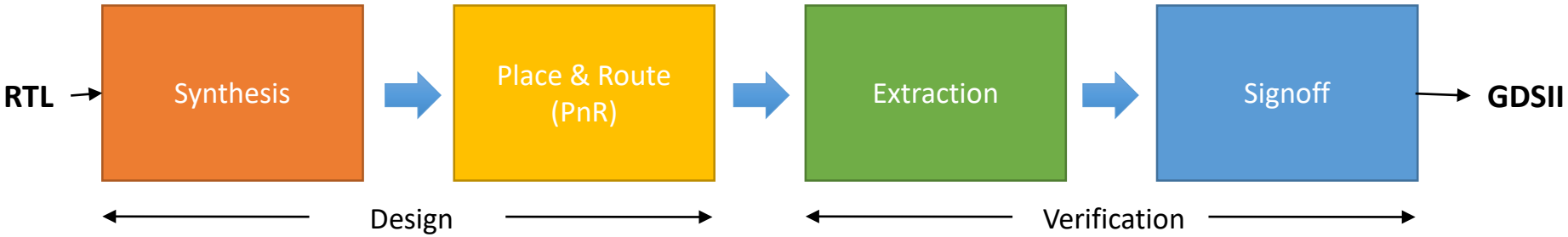
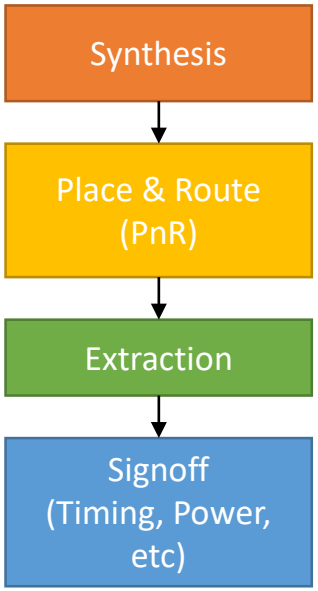


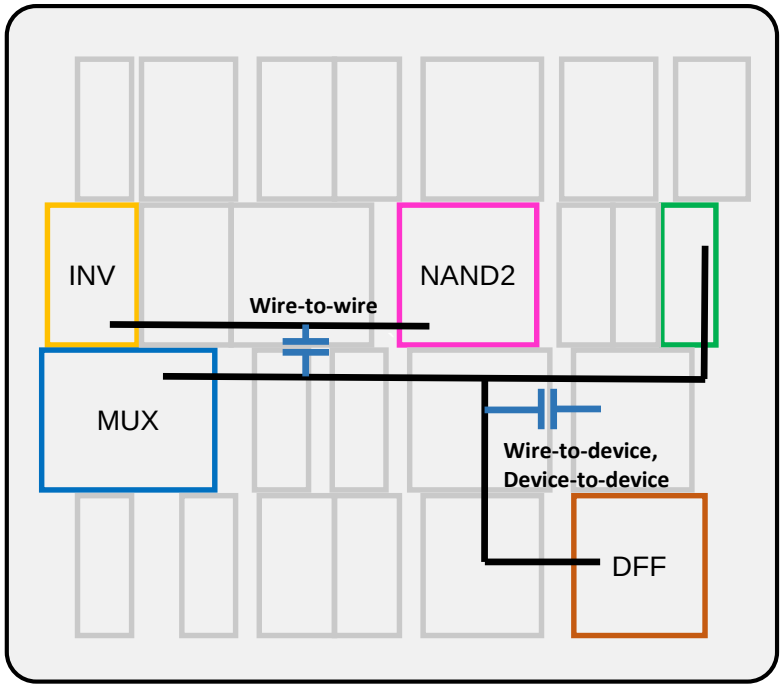
Image source : Intel

Silicon Design and Verification

Details of IC design and verification



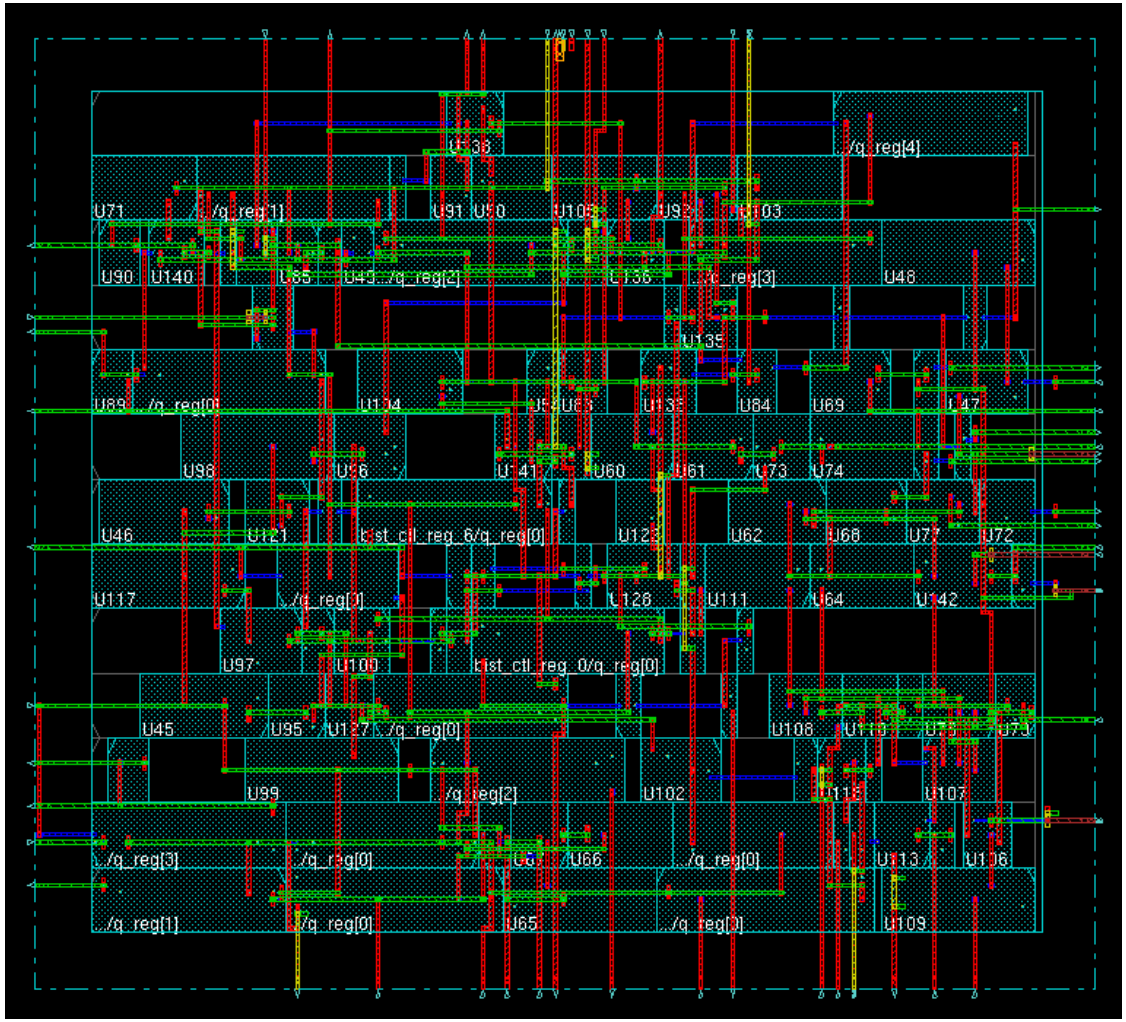
Design & Verification Flow



A Sample Layout

Actual Layout

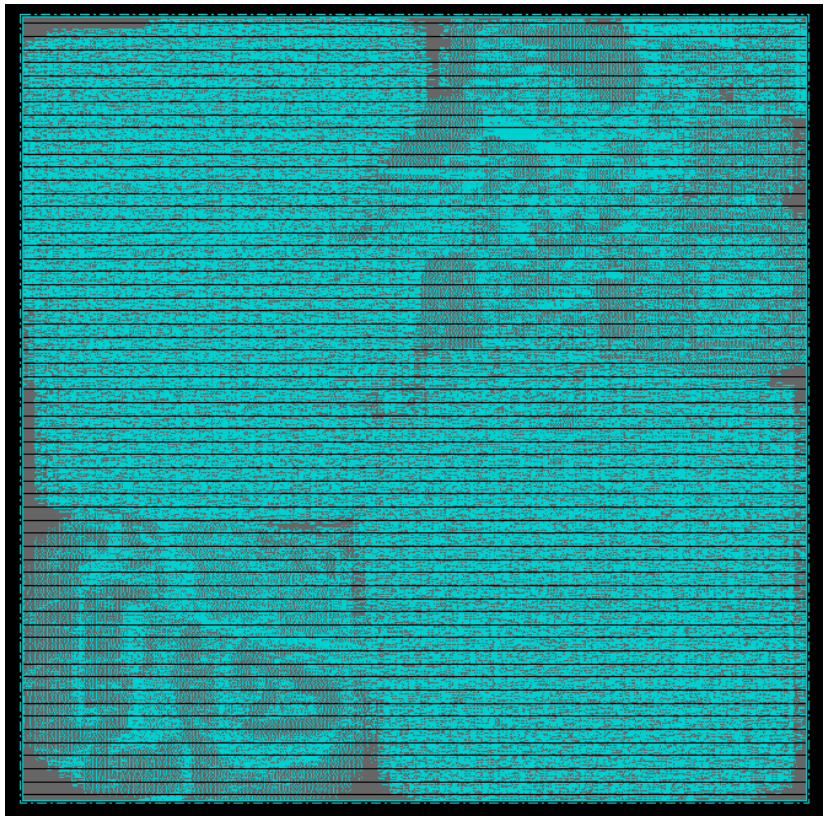
test_stub_bist (in OpenSPARC T1)



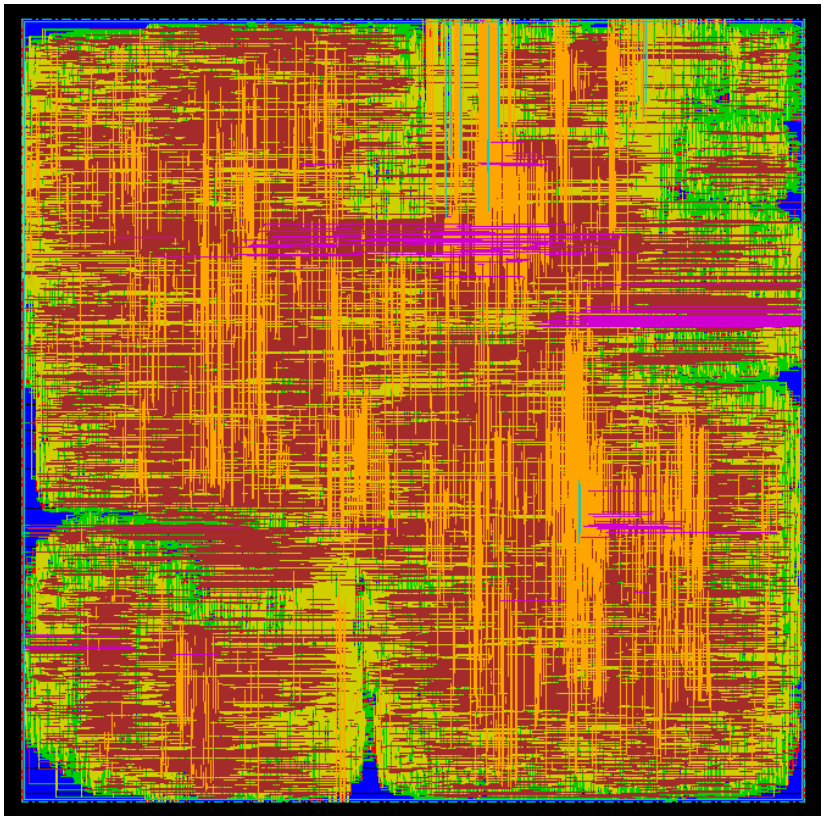
Layout of FGU

A module in OpenSPARC T1

- In 90nm Technology
  - Cell count: 111k

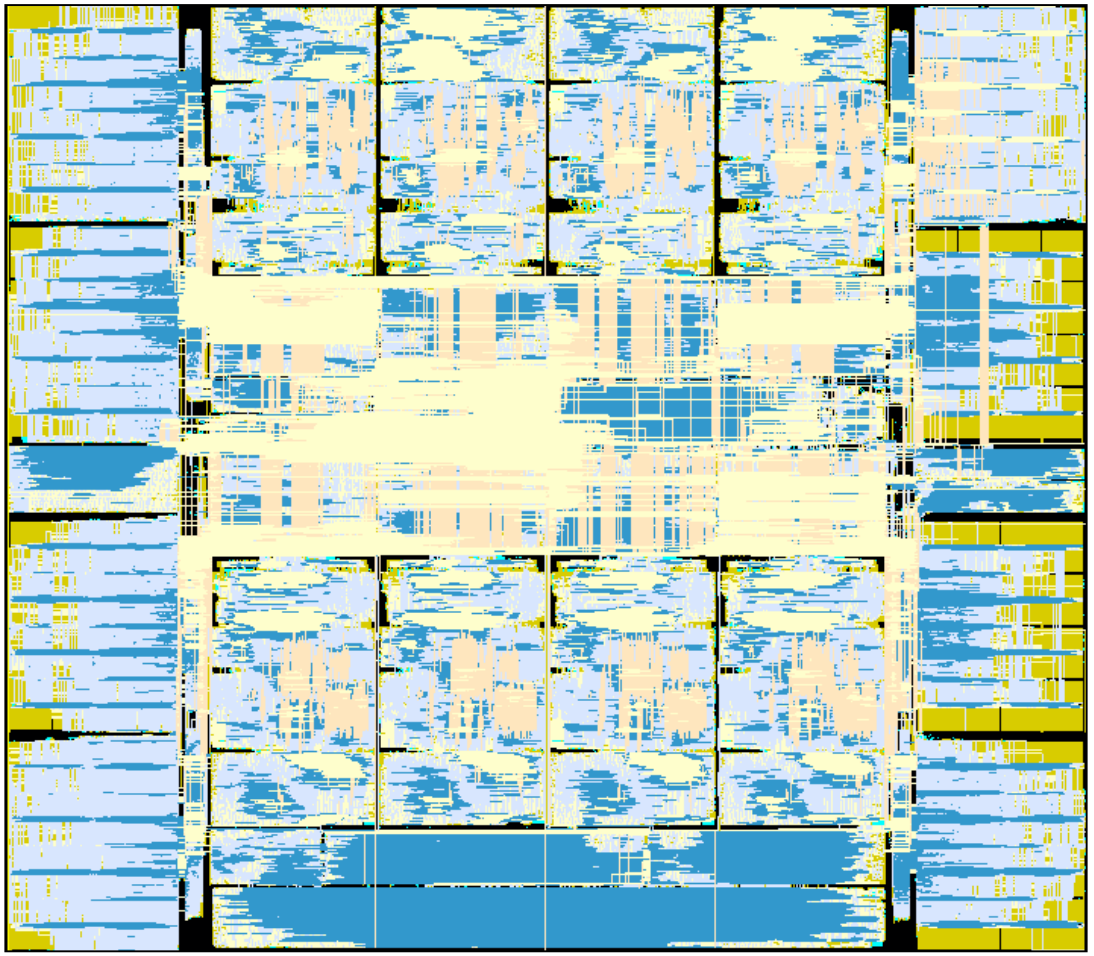


1676um



GDSII Layout of a Commercial CPU

- OpenSPARC T2 in 28nm Technology
  - 9mm x 8mm
  - 7.41M cells

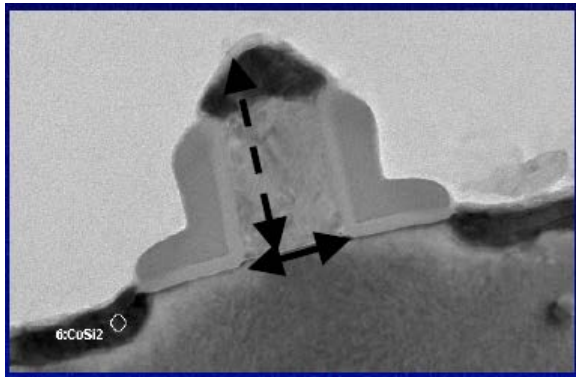


Importance of HDLs

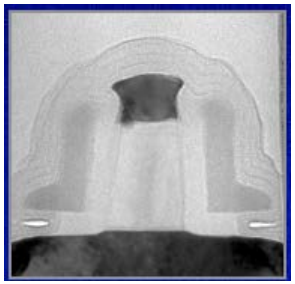
- A very abstract level design is possible by HDLs
  - Immune to fabrication technology
  - By Synthesis magic!
- Functional verification can be done early in the design cycle
  - Reducing $$$
- Easier to develop and debug

50+ Years of Transistor Scaling

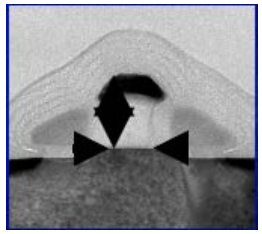
Accelerated scaling of planar transistors from 2001



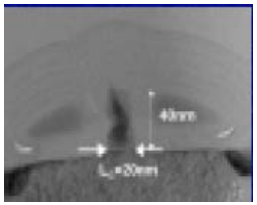
130nm Node
(Production 2001)



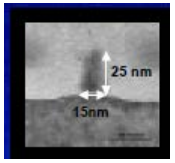
90nm Node
(Production 2003)



65nm Node
(Production 2005)



45nm Node
(Production 2007)

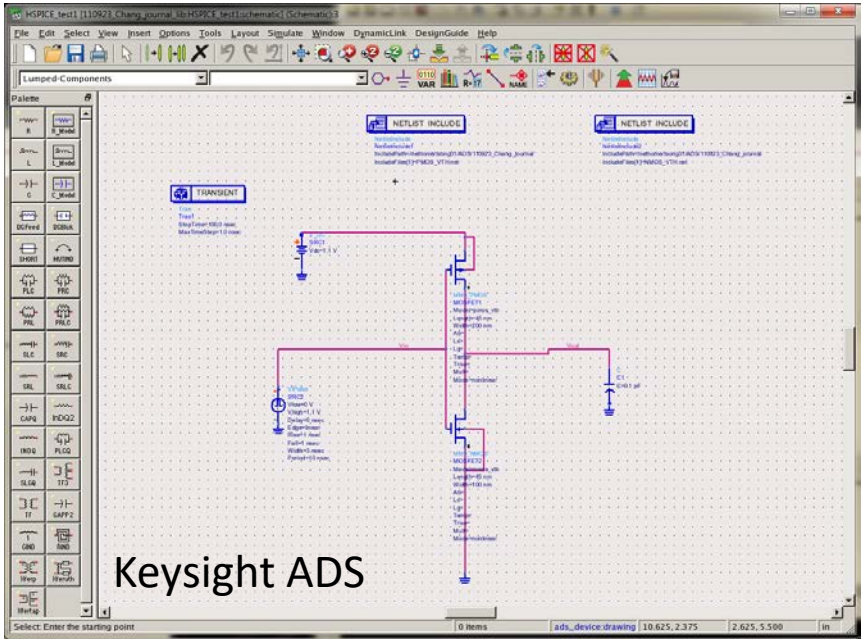


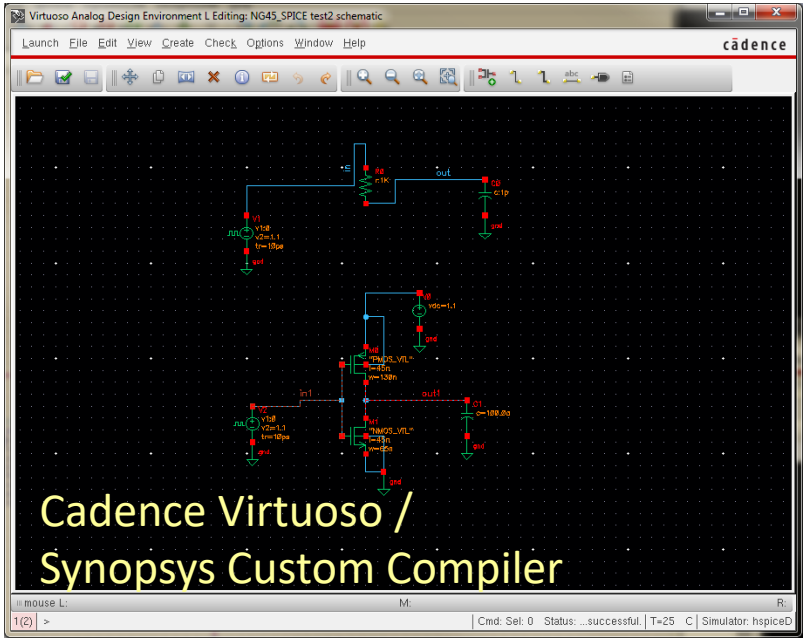
32nm Node
(Production 2009)

Source: Intel
Gerald Marcyk, "High Performance Non-planar Tri-gate Transistor Architecture"

Schematic vs. Verilog

Which is more simple?






```
1.  module inv(input A, output B);
2.
3.      //-- Both the input and the output are "wires"
4.      wire A;
5.      wire B;
6.
7.       //-- Assign the inverse of the input, to the output
8.       assign B = ~A;
9.
10. endmodule
```

Verilog HDL

Popularity of Verilog HDLs

- General-purpose hardware description language
  - Similar to C language (easy to use).
- Allows different levels of abstraction to be mixed
  - Switches, gates, RTL, or behavioral code
- Popular logic synthesis tools support Verilog HDL
  - Synopsys Design compiler, Cadence Genus … etc
- Abundant libraries
  - All fabrication vendors provide Verilog HDL libraries for post-logic synthesis simulation
- Programming Language Interface (PLI)
  - Other languages to interact with Verilog HDL

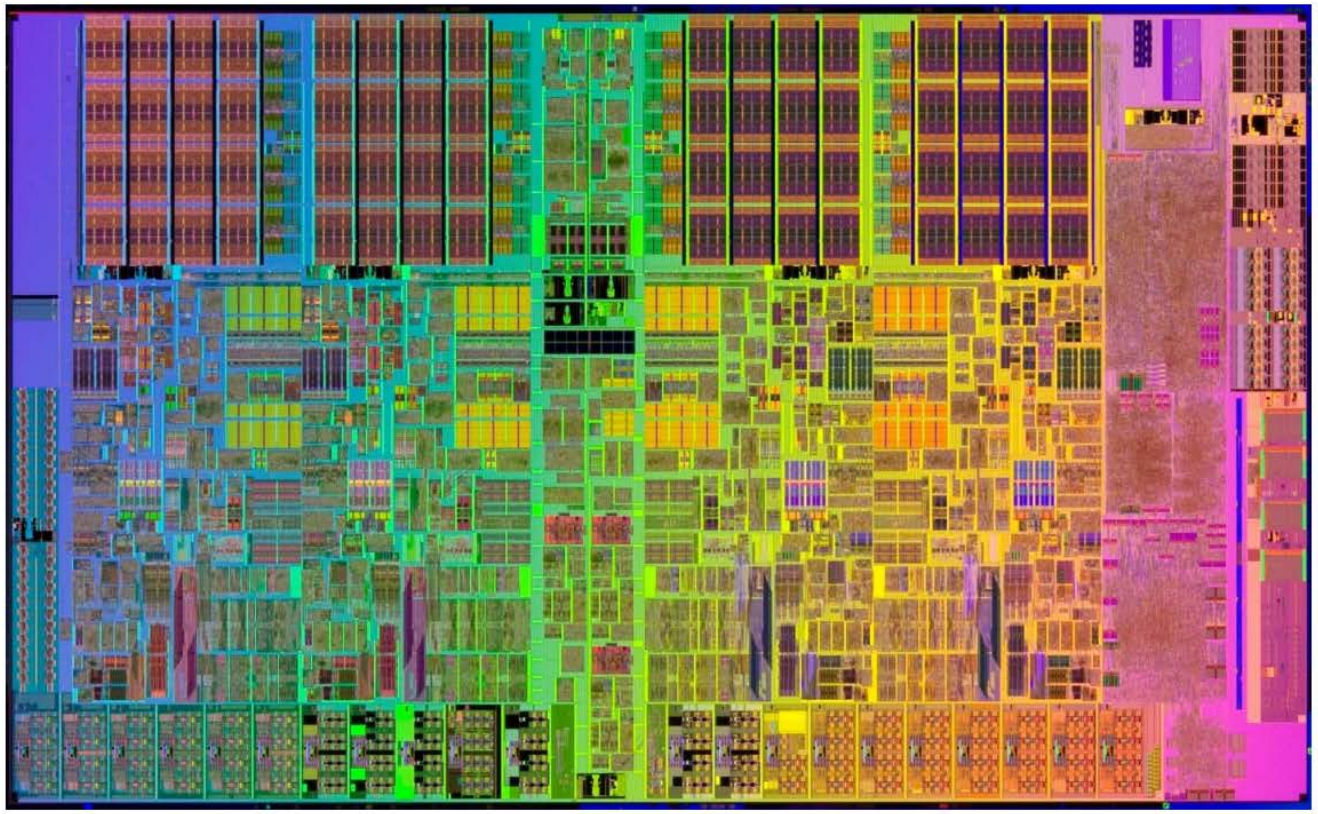
Trends in HDLs

- To design HDLs in an RTL level
  - From now on, let's use the term RTL for Verilog source coding
- Formal verification and assertion checking techniques emerged
- New verification languages
- For very high-speed and timing-critical circuits like microprocessors, gate-lv netlists by logic synthesis tools is not optimal.
  - Often gate-lv description directly mixed into RTL coding
- Mixed bottom-up methodology for system-lv design
  - A mix of RTL (e.g., CPU) and behavioral modules (e.g., I/O, GPU, bus …etc)

Why Not Design a Complex IC?

(Integrated Circuits)

- Can I make this using Verilog HDL?
  - Yes!!!



Source: Intel

So, Now What?

- In this class, let's target in designing a reasonable logic structure
  - Say, a calculator

Amazon's Choice

Calculator, Helect Standard Function Desktop Calculator - H1001

https://www.amazon.com/Calculator-Helect-Standard-Function-Desktop/dp/B01B5MU6JG/ref=sr_1_4?keywords=calculator&qid=1551576516&s=gateway&sr=8-4

Note. This lecture is not connected to the product by any means and is describing this product solely for educational purpose.

Let’s Design a Calculator

That’s all you need for this semester

- Primary function:
- Ingredients: VDD/VSS, logic gates
- Questions
  - How would I want to add numbers?
    - Numbers… how do I represent them?
    - How do I represent addition?
    - What more?