



**DO NOT SHARE  
SLIDES AND CLASS MATERIALS  
ON ONLINE SITES**

Course Home

# Advanced Logic Design

## Lecture 1. Introduction

Mingoo Seok  
Columbia University

# About the Course

# About this Course

- An introduction to modern digital system design. Advanced topics in digital logic: controller synthesis (Mealy and Moore machines); adders and multipliers; structured logic blocks (PLDs, PALs, ROMs); iterative circuits. Modern design methodology: register transfer level modelling (RTL); algorithmic state machines (ASMs); introduction to hardware description languages (Verilog); design project.

# Course Goals

- Lecture/theories:
  - Logic (combinational & sequential) hardware design
  - Computer arithmetic hardware design
- Labs/skills
  - Verilog
  - Scripting languages (e.g., Perl, Python, Bash)
  - CAD tools: Verilog functional simulator, logic synthesis, static timing and power analysis

# About this Course

- Instructor
  - Mingoo Seok
  - 1012 CEPSR, 212-854-1701, ms4415@columbia.edu
  - Office Hours: by appointment
- Teaching assistants and graders
  - TA: Daniel Jang (dj2563)
  - Grader: Tianyu Qin (tq2155), Zhengping Zhu (zz2989)
- Webpage
  - <http://courseworks.columbia.edu>

# About the Course

- Lectures
  - TT 2:40pm-3:55pm
  - Location: 750 CEPSR
  - Mid exam: 25% & Final exam: 25% vs only Final 50%
- Homework
  - Total five, but we will grade **only the first two**: 15%
- Pre-project labs
  - Lab session starts in two or three weeks (Fridays)
  - No report submission
- Project-related labs
  - Each milestone has a report & the final report
  - Total five reports: 25%
- Class attendance:
  - 10%

# Text Books

- Recommended: Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, Third Edition
- Optional
  - Randy H. Katz, Contemporary Logic Design, Benjamin/Cummings Publishing Company, Inc., Redwood City, CA (1994).
  - Israel Koren, Computer Arithmetic, A K Peters, (2002)



# Additional Materials

- We will use the slides like today
- We will upload/distribute the handouts and papers, if needed
- Feel free to search the Internets; tons of materials regarding logic design and related CAD tools, though they are not organized






# Lecture List

- L1: Introduction
- L2: Combinational logic circuits refresher
- L3: Verilog
- L4: Sequential circuits
- L5: FSM
- L6: Arithmetic, CORDIC, DA
- L7: Project overview / buffer for lectures

# Lecture List

- L8: ASM
- L9: Microprogram
- L10: Low power design
- L11: Fault-tolerance and error correction
- L12: Synchronous & asynchronous
- L13: Testing
- L14: FPGA  $\mu$ A (if time allows)

# Homework

Homework	
	<b>Homework #1</b> Available until Sep 29 at 11:59pm   Due Sep 29 at 11:59pm   7.5 pts
	<b>Homework #2</b> Available until Oct 13 at 11:59pm   Due Oct 13 at 11:59pm   7.5 pts
	<b>Homework #3</b> Available until Nov 10 at 11:59pm   Due Nov 10 at 11:59pm
	<b>Homework #4</b> Available until Nov 28 at 11:59pm   Due Nov 28 at 11:59pm
	<b>Homework #5</b> Available until Dec 13 at 11:59pm   Due Dec 13 at 11:59pm

- Roughly every two weeks from Sep/29
- Submission, grading via Coursework
- #1 and #2 will be graded

# Lab & Design Project

- We will apply what learn in lectures onto actual hardware design
- This year's project target is: FIR processor
- It involves:
  - System design (Matlab)
  - RTL design
  - Logic synthesis
  - Timing, power, and accuracy characterization

# Pre-Project Lab Sessions

- Overview of the design flow
  - Verilog basics
  - Logic synthesis
  - Post-synthesis Verilog simulation
  - Static timing and power analysis (STA)
  - SRAM memory compiler
- 
- Led by the TAs
  - A reference design (LFSR: Linear Feedback Shift Register) is provided

# Pre-Project Labs

- LB1: Design flow overview
- LB2: Verilog
  - LA1: write the Verilog codes of a 32-bit ripple carry adder and a carry look-ahead adder and the testbenches
- LB3: Logic synthesis
  - LA2: 32-bit adder synthesis and verification for 10-MHz, 100-MHz, and 1-GHz clock targets
- LB4: Static performance, power, area sim
  - LA3: PPA analysis of the adders for the above clock frequency targets
- LB5: Memory complier
  - LA4: Create the Verilog codes of X kB memory and testbench to write a set of values from a file

**No submission and grading**

# Design Project

- A team of  $\leq$  **two** students
- This year's design target is:
  - **An FIR filter design**
  - But if you want to explore other architectures, talk to me
- Metrics
  - Accuracy (% for 10-k randomly generated sample)
  - Performance (MSample/s)
  - Energy efficiency (pJ/Sample)
  - Area (mm<sup>2</sup>)



# Project-Related Labs

LB5: Golden block in Matlab; Architect the FIR core;

- LA5: Given coefficient and input patterns, produce the output; architect the FIR core

LB6: Develop ALU: RTL coding, logic synthesis, func. test, PPA analysis

- LA6: Report on the developed ALU

LB7: Develop IMEM & CMEM & Register file: RTL coding, logic synthesis, func. test, PPA analysis;

- LA7: Report on the developed memories

LB8: Develop FIFO: RTL coding, logic synthesis, func. test, PPA analysis

- LA8: Report on the developed FIFO

LB9: Develop the FIR core: RTL coding, logic synthesis, func. test, PPA analysis

- LA9: Submit the final report

**Submit, and we will grade them**

# Project Presentation

- We will ask a few teams to present their progress in the week of Nov/16 to Nov/30 (TBD)
- The presentation is a great chance to get the instructors' feedback on the architecture
- We will use some of the existing lecture time, and it is likely to have only a few (1-3) slots
- Ask for the slot early!

# Sign up for labs and projects

- We need
  - Your name
  - UNI
  - Card number (for swipe access to the computer labs)
  - Availability info on Friday (to set up the lab time)
- Please use the link to the shared doc:
  - TBD
  - <https://docs.google.com/spreadsheets/d/1eNYniym1AmTHjL2SuTlfCyE4yS3OdtkPJgSBtjLj3dQ/edit#gid=0>

# Signing NDA

- In lab session and design project, you are using 130nm CMOS technology
- Need to sign the MOSIS NDA form today
- You should submit it ASAP via Coursework
  - The form is at Files @ Courseworks

# Submit a Signed NDA

≡ CSEEW4823\_001\_2020\_3 - ADVANCED LOGIC DESIGN > Assignments > Submit a signed NDA

Fall 2020

[Home](#)

[Announcements](#) 

[Course Info](#)

[Syllabus](#)

[Files](#)

[People](#)

[Attendance](#)

**[Assignments](#)**

[Quizzes](#)

[Grades](#)

[Submit Grades to  
SSOL](#)

[Photo Roster](#)

[Library Reserves](#)

[Textbooks](#)

This assignment does not count toward the final grade.

## Submit a signed NDA

✓ Published

 Edit



Please use scan the signed NDA and submit the pdf/jpg/jpeg file. The NDA is:  
-http://www.mosis.com/forms/mosis\_forms/academic\_nda\_non-liaison.pdf

Points **3**

Submitting a file upload

File Types pdf, jpg, and jpeg

Due	For	Available from	Until
Sep 18	Everyone	-	-

+ Rubric

# Project Group; Self Sign-Up

**Left Sidebar:**

- Account
- Dashboard
- My Courses
- Courses
- Calendar
- Inbox
- Help

**Top Navigation:**

- CSEEW4823\_001\_2019\_3 - ADVANCED LOGIC DESIGN > People > Groups

**Project groups**

Self sign-up is enabled for these groups. Groups are limited to 2 members.

**Unassigned Students (52)**

Search users
Pratyush Agrawal
Fabio Andre Cam...
Jamison Bunge
Weihan Chen
Allan Delarosa
Austin Ebel
Shayel Encaoua
Xiaohan Feng
Pooja Ganesh
Daniel Garces
Mae Graham
Shanglin Guo
Jino Haro
Pushan Hinduja
Yuchan Hsueh
Zhili Huang

**Groups (30)**

Group Name	Students
Project group 1	0 / 2 students
Project group 2	0 / 2 students
Project group 3	0 / 2 students
Project group 4	0 / 2 students
Project group 5	0 / 2 students
Project group 6	0 / 2 students
Project group 7	0 / 2 students
Project group 8	0 / 2 students
Project group 9	0 / 2 students

# References

- Lecture slides will be uploaded
  - Slides will only cover summary points
- Design project
  - Access them from IEEE Xplore
  - <http://ieeexplore.ieee.org>
  - Check background references from library
- Opencores.org
  - It has some Verilog code for implementing FFTs and other hardware blocks



# Remote Access to the Lab Computers

## EE Computing Lab Remote Access

In addition to [remote01.ee.columbia.edu](#)-[remote02.ee.columbia.edu](#), lab computers [cadpc01.ee.columbia.edu](#)-[cadpc42.ee.columbia.edu](#) and [micro01.ee.columbia.edu](#)-[micro35.ee.columbia.edu](#) can currently be accessed using the method below. You can use the [remote lab workstation status page](#) to help you identify open machines.

In order to access the labs remotely, you'll make use of a VNC client and an SSH client. [TigerVNC](#) is a VNC client that works on Windows, Mac OS X, and Linux. SSH clients are normally installed by default in Mac OS X and Linux. On Windows, two options are [PuTTY](#) and [MobaXterm](#).

The machines [remote01.ee.columbia.edu](#) and [remote02.ee.columbia.edu](#) are workstations that have a VNC server installed on them. You'll need to start a VNC session on one of those workstations. In order to do so, ssh into one of those machines and run the command 'vncserver'. The first time you run the command, it will prompt you to create a password. The particular password you choose is not crucial, but you'll need it at a later step. The command will indicate that it has created a new desktop. You'll need to take note of the number after the colon. That number affects what port the VNC server is available on. You can now end this initial SSH session.

Because VNC doesn't have good security built into it, you'll use SSH to tunnel the traffic to the VNC server. If you connected to [remote02](#), your desktop number from before was 3, and your UNI was [aa9999](#) you could run a command like the following from a local terminal:

```
ssh -C -L 5910:localhost:5903 aa9999@remote02.ee.columbia.edu
```

Then, while that connection is open, you can connect a VNC client on your local computer to [localhost:5910](#). You're welcome to choose any open port instead of 5910. It just has to be the same port that you forward to the machine running the VNC server using SSH. At that point you'll be prompted for the password you created earlier, which should let you connect to the persistent desktop being run by the VNC server. You can ignore warnings about security here because you're tunneling the traffic over SSH.

If you disconnect without logging out, the desktop will stick around. If you log out, you'll need to start a new instance of vncserver before connecting again. The command 'vncserver -list' run on [remote01](#) or [remote02](#) will show you any existing sessions you have running on that machine. You can create an instance with a particular desktop number by passing that as a parameter when you start vncserver. You could run, for example:

```
vncserver :12
```

Then the connection you create for your tunnel would use a command such as:

```
ssh -C -L 5910:localhost:5912 aa9999@remote02.ee.columbia.edu
```

MobaXterm also has a feature for saving tunnels, and PuTTY can remember your sessions including the particular port being forwarded. One advantage of selecting a particular desktop number when you create the VNC server instance is that you can always use the same port for the tunnel.

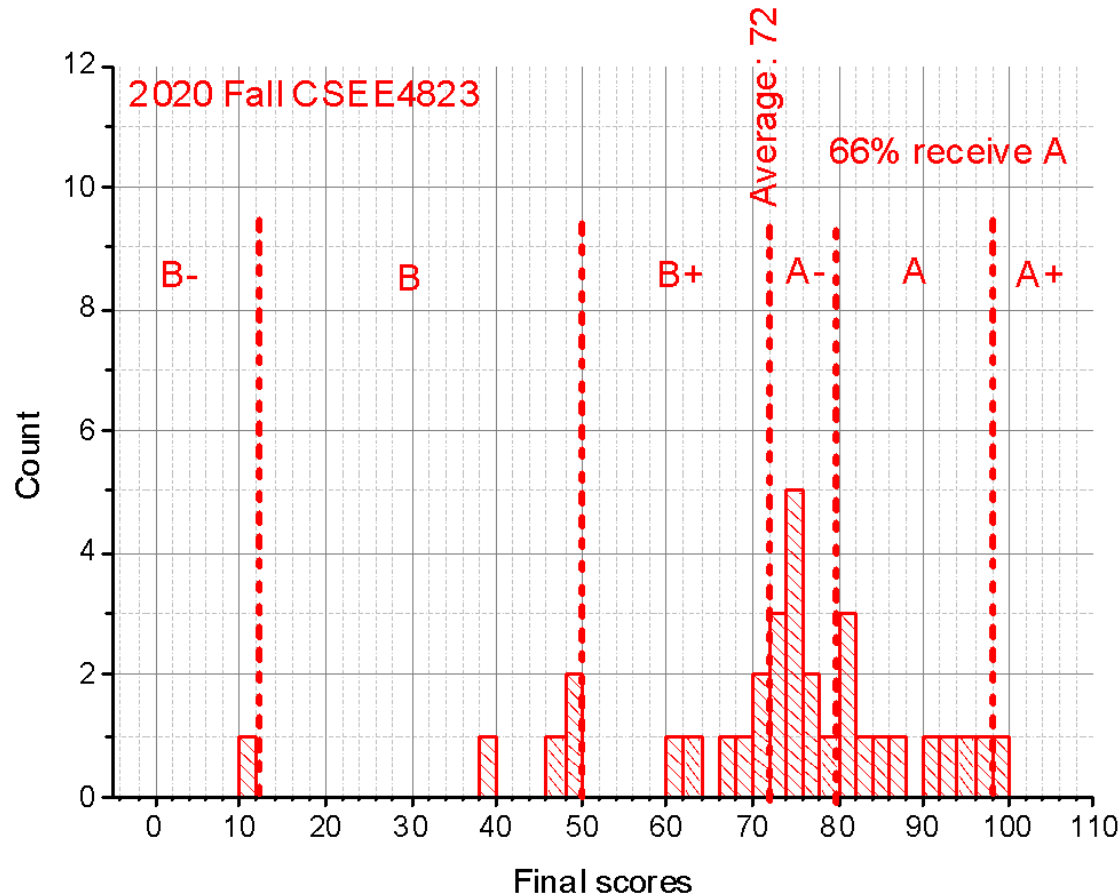
In any case, if you still have an existing VNC server instance running, you can just create the appropriate tunnel and then connect the VNC client. New server instances only need to be started after logging out. In general, though, it's good to actually log out of your sessions if you're not trying to save progress of some particular piece of work. Programs that you leave open can cause problematic behavior when you're logged in at another machine since your home directory is shared across all the workstations.

- Please find the instruction

<https://www.ee.columbia.edu/content/ee-computing-lab-remote-access>



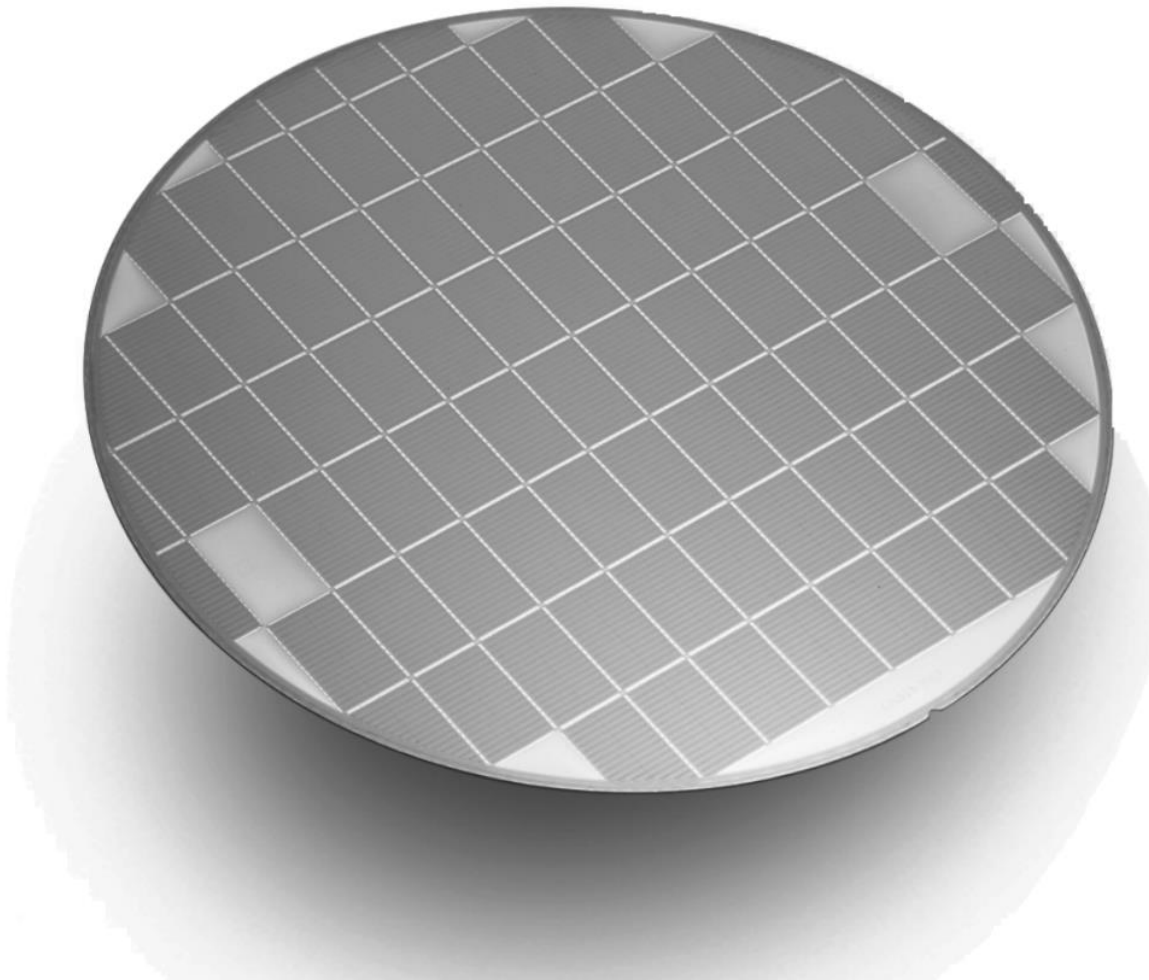
# Grades



- While we ask you to do a lot of things, the final grade would be, I think, quite generous. If you decide to take it, I hope you not be too stressed about the grade

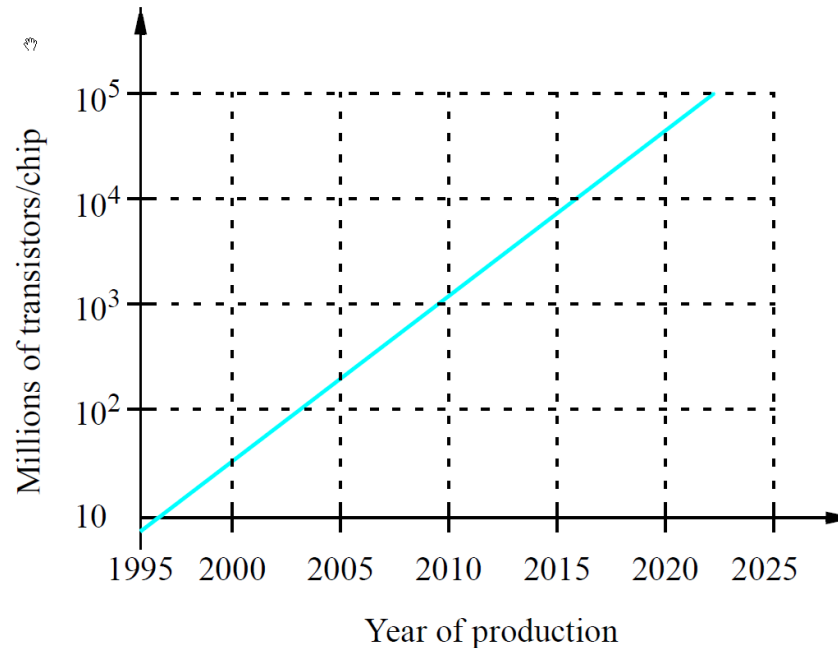
# IC Design

# Integrated Circuits (IC)



- Until the 1960s logic circuits were constructed with bulky components such as transistors and resistors
- By 1970 it was possible to implement almost all circuitry needed to realize a microprocessor on a single chip

# Moore's Law



- Early in the 1950's Gordon Moore made a bold prediction that is the number of transistors on a chip would be doubled every two years
- The Moore's law is still going on. In 2015, the # of transistors is roughly 10 billion

# Critical IC Design Technologies

- To answer this question, let's first answer to another question: why semiconductor is important?
- Three reasons
  - Essential in many other products (car, cell tower, electric grids. Pretty much everything)
  - Show me the money: Global sales are 600B USD in 2022
  - Military & defense! - In WW2, the USA

# What Semiconductor Chips are Important?

## Logic – Memory – Analog – Discrete

- Logic: microprocessors, GPU, FPGA
  - Need cutting-edge fabrication technology (5nm, 3nm, etc)
  - Complex to design
  - Need to consider other technologies: CAD, computer architecture, software
- Memory: DRAM, FLASH
  - Also need cutting-edge fabrication technology
  - Less complex to design
  - Less worry about CAD, computer architecture, software
- Analog: imagers, ADC, DAC, PMIC
  - Don't need cutting-edge fabrication technology
  - Know-how is critical (difficult to mimic)
  - High variety and low volume
  - Less worry about CAD, computer architecture, software
- Discrete: transistor, capacitor, inductor, resistor, etc

# What Semiconductor Chips are Important?

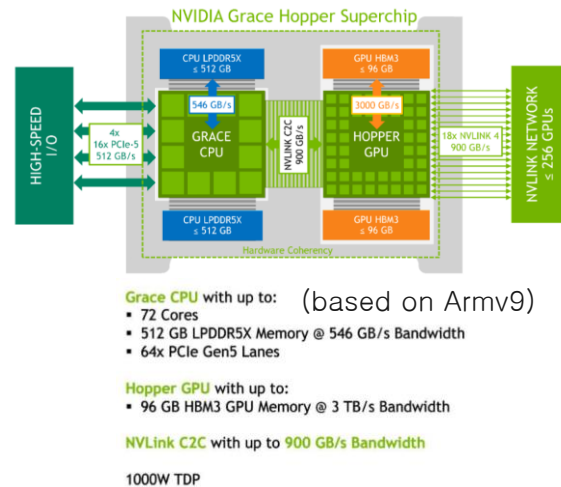
## Logic – Memory – Analog – Discrete

- Logic: microprocessors, GPU, FPGA
  - Need cutting-edge fabrication technology (5nm, 3nm, etc)
  - Complex to design
  - Need to consider other technologies: CAD, computer architecture, software
- Memory: DRAM, FLASH
  - Also need cutting-edge fabrication technology
  - Less complex to design
  - Less worry about CAD, computer architecture, software
- Analog: imagers, ADC, DAC, PMIC
  - Don't need cutting-edge fabrication technology
  - Know-how is critical (difficult to mimic)
  - High variety and low volume
  - Less worry about CAD, computer architecture, software

# Microprocessor and GPUs

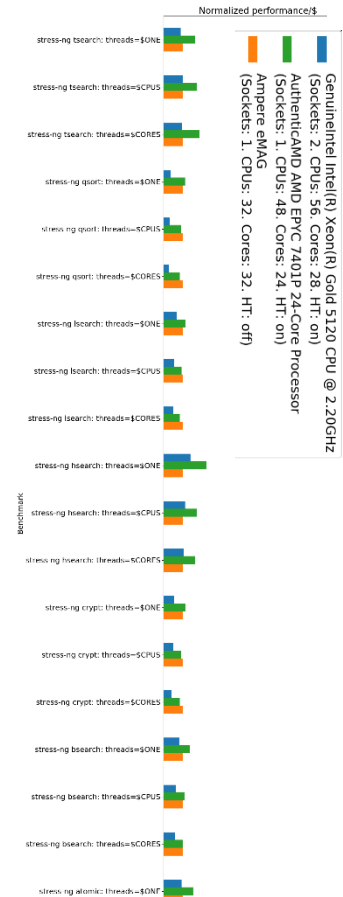
- Data center, cloud, HPC, and PC
  - CPU: USD 85B in 2020, CAGR of 4.1%
  - GPU: USD 25B in 2020, CAGR of **32.8%** (estimated 250B in 2028)
- Key players: Intel, AMD, NVIDIA
  - Loongson (China): LS3C5000 : LA464 (MIPS based) 2.2GHz, 16 cores, 32 MB L3, DDR4-3200, 150W,
- R&D directions:
  - Breeding edge CMOS technology
  - Packaging
  - More cores
  - More memory
  - Faster off-chip links
  - Thermal, reliability, power delivery
  - ARM-based (low power, low thermal)

## Grace Hopper Superchip



## GTX 4080, \$1200

NVIDIA CUDA Cores	9728
Boost Clock (GHz)	2.51
Memory Size	16 GB
Memory Type	GDDR6X
Max Display Resolution	4K at 240Hz or 8K at 60Hz, with DSC





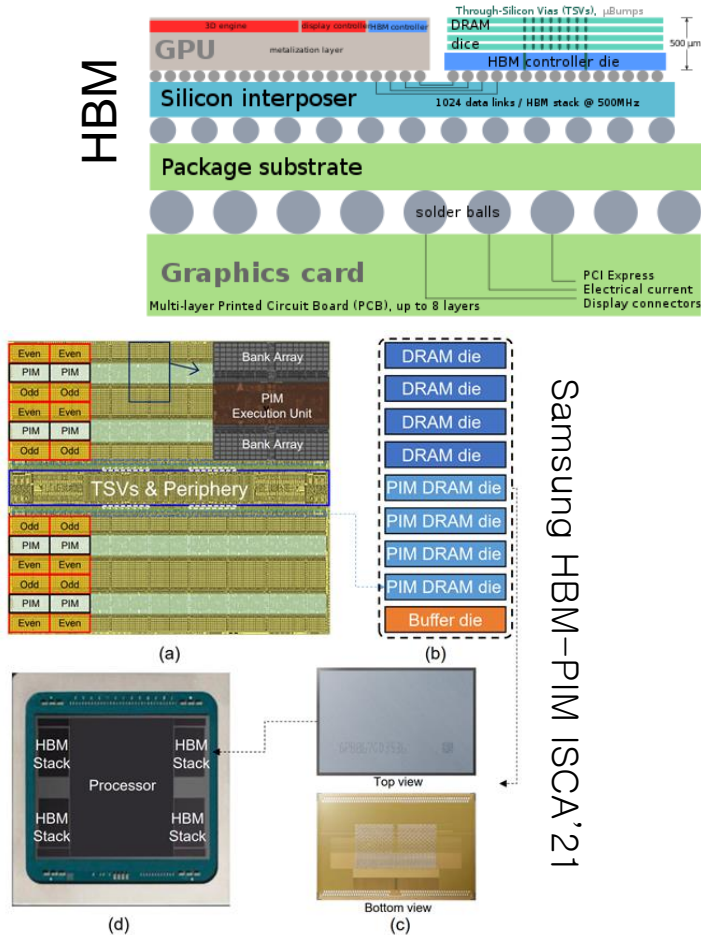
# Application Processor (AP)

Global Smartphone Chipset Market Share (Q2 2021 – Q3 2022)						
Brands	Q2 2021	Q3 2021	Q4 2021	Q1 2022	Q2 2022	Q3 2022
Mediatek	42%	40%	35%	36%	38%	35%
Qualcomm	26%	27%	29%	33%	29%	31%
Apple	14%	15%	20%	14%	13%	16%
UNISOC	9%	10%	11%	11%	11%	10%
Samsung	5%	5%	4%	5%	8%	7%
HiSilicon (Huawei)	3%	2%	1%	1%	0%	0%

Source: Global Smartphone AP-SoC Shipments & Forecast Tracker by Model – Q3 2022

- Heart of the mobile/embedded computing (smart phones, tablets, and drones)
- USD 26B in 2021, CAGR of 6.7%
- Key players: MediaTek (Taiwan), Qualcomm, Apple, UNISOC (China), Samsung, Huawei/HiSilicon (China)
- R&D directions: fast, low power, better support AI workload, implemented in a cutting-edge technology node

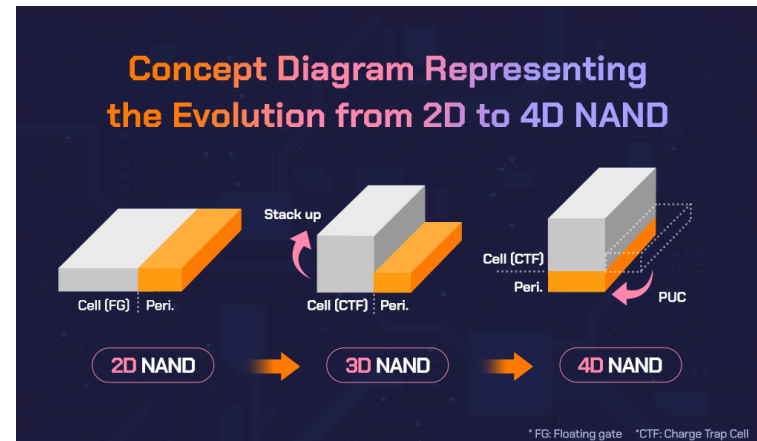
# DRAM



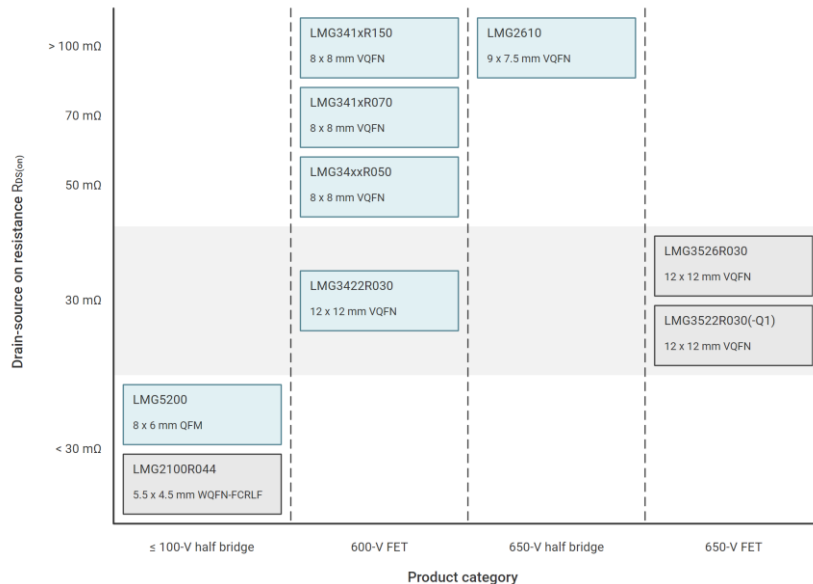
- Main memory in almost all computing systems (mobile and cloud)
- USD 105B in 2021, CAGR of 8.8%
- Key players: Samsung, SK Hynix, Micron
- R&D directions:
  - Speed: DDRX, LPDDR, GDDR
  - Dense (price): Extreme ultraviolet (EUV) (Samsung announced the mass production of the 14nm EUV DDR5 in Oct/2021)
  - 3D: High bandwidth memory (HBM)
  - Processing in memory (PIM)

# NAND

- Key data storage in almost all computing systems
- USD 66.52B in 2021, CAGR of 5.33%
- Key players: Samsung, KIOXIA (Toshiba), Micron, SK Hynix, YMTC (China)
- R&D directions:
  - Dense (price): 236 layer (July 2022), 2D→3D→4D



# PMIC (Power Management Integrated Circuits)



- Heart of power supply/management of computing systems
  - Voltage regulator, battery management, motor control IC, etc
  - Discrete (power FET, capacitor, inductor)
- USD 33B in 2022, CAGR of 6.3%
- **Many** players: Renesas, Texas Instruments, Analog Device, NXP, ON Semi, Dialog Semi, Maxim, Nordic, Qualcomm, ROHM, Mitsubishi, ABB, Allegro
- R&D directions:
  - Support higher voltage, more current, larger conversion ratio, efficiency, size reduction, reliability
  - Support new applications (e.g., EV)

# Chip and SW Vertical Integration

---

- Past: Horizontal integration & no end-product
- Apple manufactures its own
  - Application processor (M1, M2)
  - It plans the 5G modem and the WiFi/BT connectivity chip
- Benefits:
  - Performance and energy efficiency
  - Custom functionality: video calling on the Macbook: Apple could add custom image processor cores for the built-in camera
  - Financial and supply-chain benefits: it reduces the cost and better control chip supply
  - Other non-silicon companies follow the similar practice

# Chip and SW Vertical Integration

- Google
  - Tensor processing unit (TPU, 2016), optimized for low-precision (8b) neural network
- Amazon
  - Amazon Web Services (AWS) records \$45B in 2020.

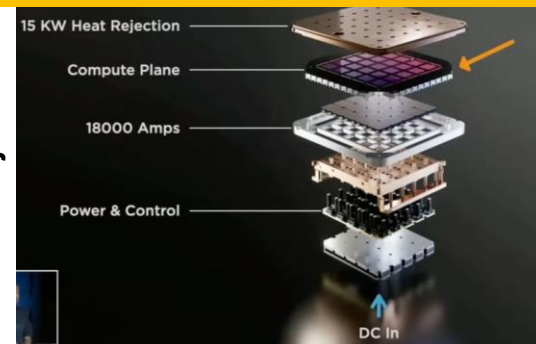
Tensor Processing Unit products<sup>[13][14]</sup>

	TPUv1	TPUv2	TPUv3	TPUv4 <sup>[14][15]</sup>	Edge v1
Date introduced	2016	2017	2018	2021	2018
Process node	28 nm	16 nm	16 nm	7 nm	
Die size (mm <sup>2</sup> )	331	< 625	< 700	< 400	
On-chip memory (MiB)	28	32	32	144	
Clock speed (MHz)	700	700	940	1050	
Memory	8 GiB DDR3	16 GiB HBM	32 GiB HBM	32 GiB HBM	
Memory bandwidth			900 GB/s	1200 GB/s	
TPUv4	75	280	280	170	0

**IC design used to be relevant for the so-called chip company**  
**But, to be competitive,**  
**more companies will and should do IC design**

- DOJO (2022): HPC (super computer), **System-on-wafer**, "Each D1 die is integrated onto a tile with 25 dies at 15kW. Beyond the 25 D1 dies, there are also 40 smaller I/O dies"
- FDS (2019): 12 ARM Cortex-A72 at 2.2GHZ, Mali GPU at 1GHz, and NPU at 2GHz (36 TOPS), LPDDR4-4266, 260 mm<sup>2</sup>

Tesla Dojo training



# Long-Term Research Effort

# Non Von-Neumann Computer Architecture

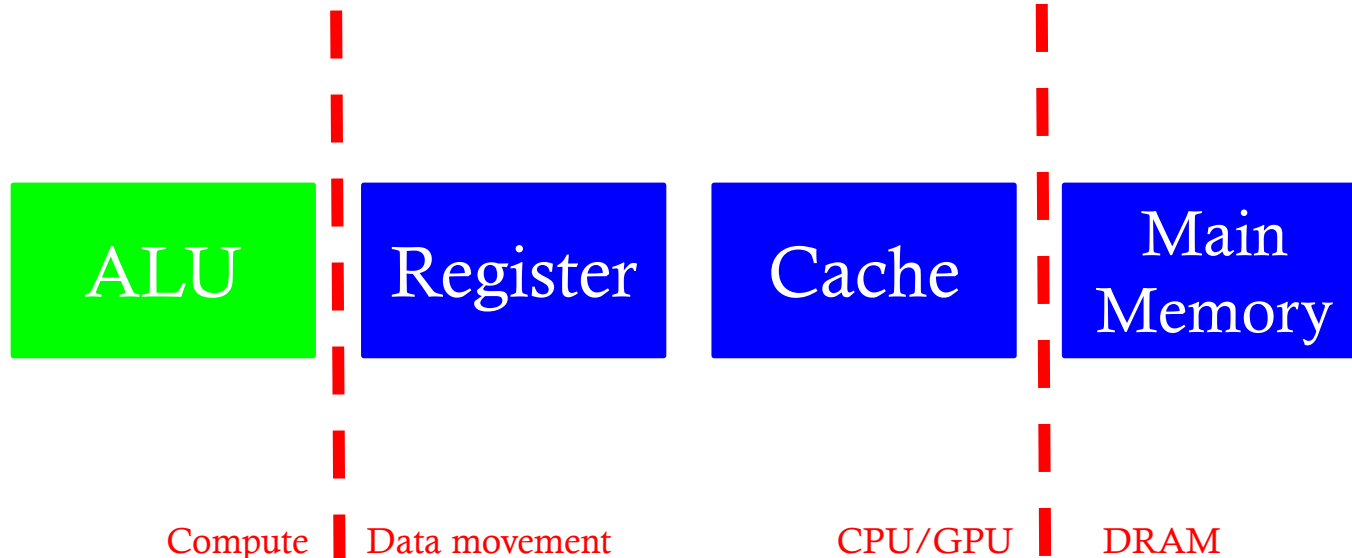
- What is Von-Neumann's computer architecture?
  - Key concept: **Stored Program** → **General purpose**
  - Before VNCA, computers were **hard-wired** to do one task. If the computer had to perform a different task, it had to be **rewired by hand**, which was a tedious process.



Female technicians connecting the wiring of the ENIAC, circa 1943-46



# Characteristics of Von-Neumann



- The most important characteristics is a separation between logic and memory
- The registers and the cache require billions of row-by-row accesses and 10-100X more costly than on-chip computation
- The physical gap between CPU and DRAM makes the off-chip communication 100-1000X more costly than on-chip computation

# Challenge: Cost of Data Movement

Table 2. Energy consumption of multiply-accumulations (Horowitz, 2014)

Operation	MUL	ADD
8bit Integer	0.2pJ	0.03pJ
32bit Integer	3.1pJ	0.1pJ
16bit Floating Point	1.1pJ	0.4pJ
32bit Floating Point	3.7pJ	0.9pJ

**4x energy / 2x precision  
in integer multiplication**

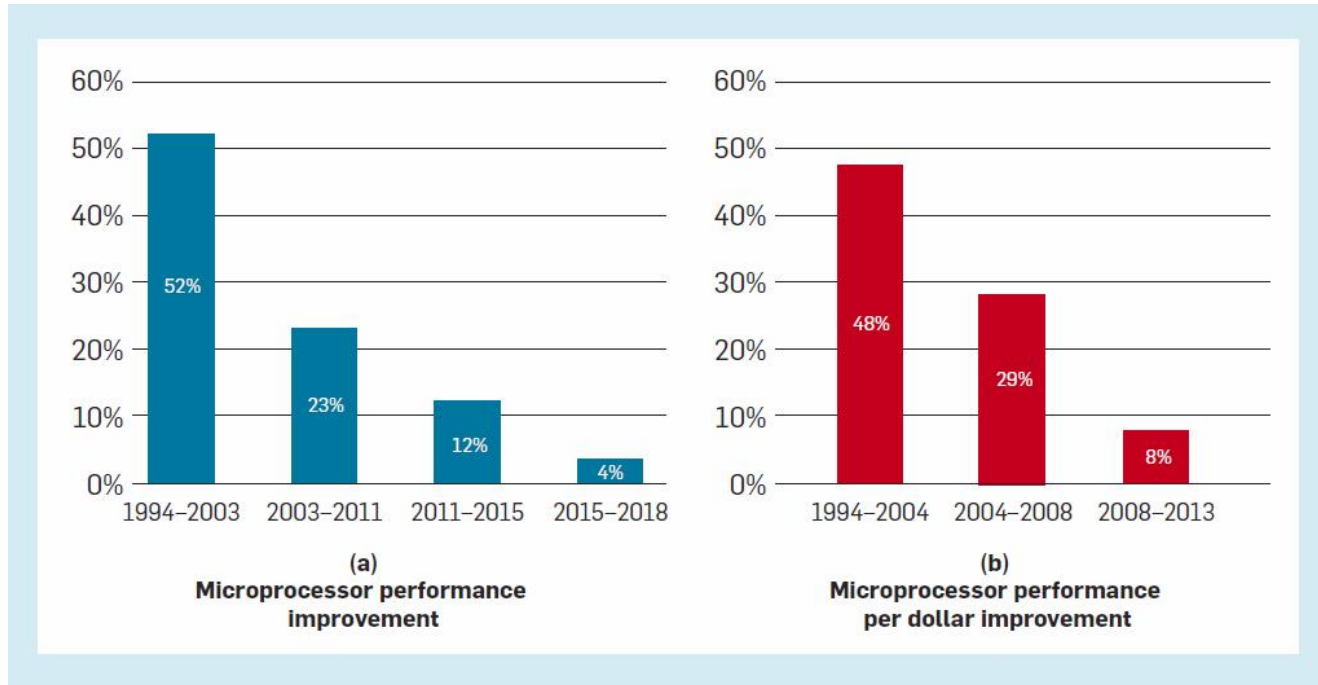
Table 3. Energy consumption of memory accesses (Horowitz, 2014)

Memory size	64-bit memory access
8K	10pJ
32K	20pJ
1M	100pJ
DRAM	1.3-2.6nJ

} On-chip  
→ Off-chip

[arXiv:1602.02830v3]

# Cost of General-Purpose Hardware



Neil C. Thompson, Svenja Spanuth, Communications of the ACM, March 2021

- The second important characteristic is targeting general-purpose microprocessors
- The transistor improves slowly and so does a microprocessor w/o architectural change

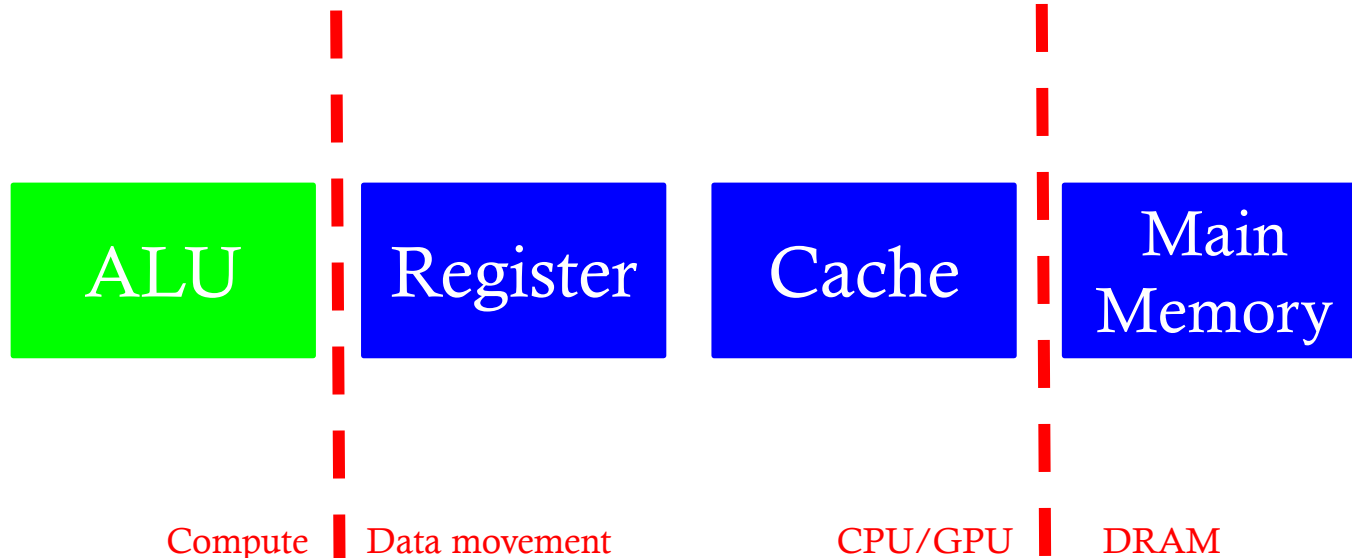
# Non Von-Neumann Computer Architecture

- What is Von-Neumann's computer architecture?
  - Key concept: **Stored Program** → **General purpose**
  - Before VNCA, computers were **hard-wired** to do one task. If the computer had to perform a different task, it had to be **rewired by hand**, which was a tedious process.



Female technicians connecting the wiring of the ENIAC, circa 1943-46

# Characteristics of Von-Neumann



- The most important characteristics is a separation between logic and memory
- The registers and the cache require billions of row-by-row accesses and 10-100X more costly than on-chip computation
- The physical gap between CPU and DRAM makes the off-chip communication 100-1000X more costly than on-chip computation

# Challenge: Cost of Data Movement

Table 2. Energy consumption of multiply-accumulations (Horowitz, 2014)

Operation	MUL	ADD
8bit Integer	0.2pJ	0.03pJ
32bit Integer	3.1pJ	0.1pJ
16bit Floating Point	1.1pJ	0.4pJ
32bit Floating Point	3.7pJ	0.9pJ

**4x energy / 2x precision  
in integer multiplication**

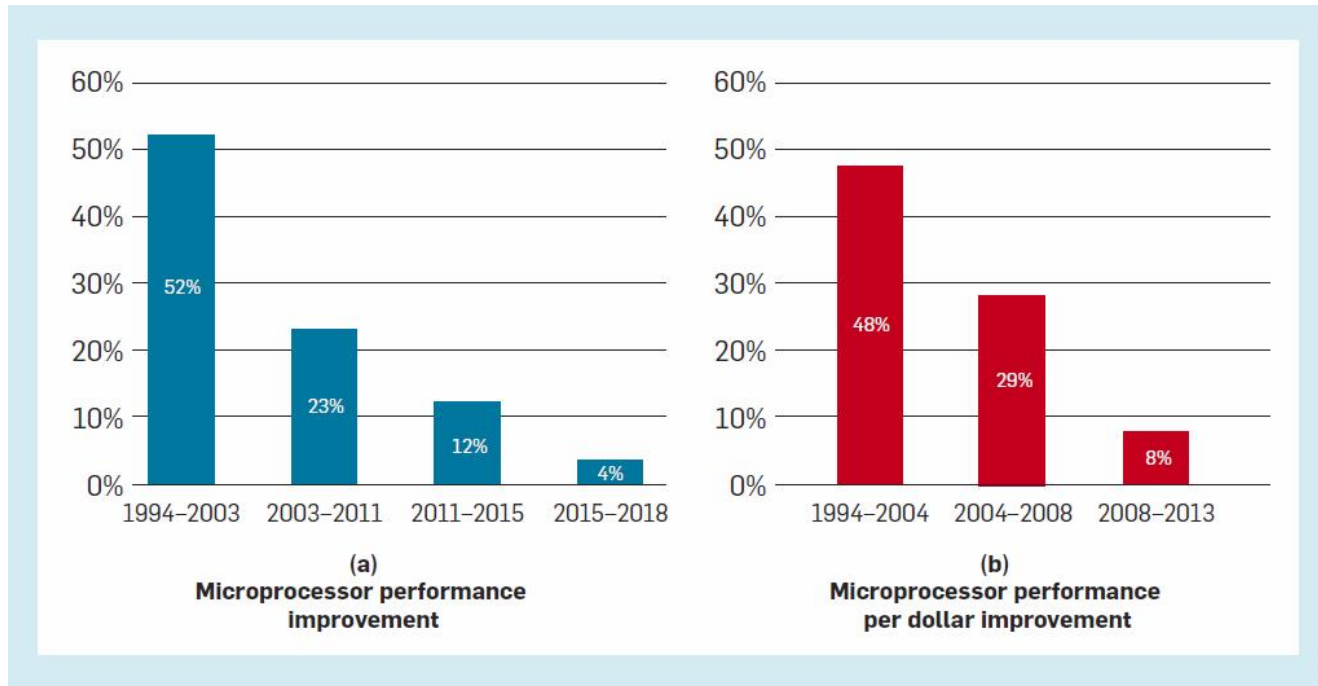
Table 3. Energy consumption of memory accesses (Horowitz, 2014)

Memory size	64-bit memory access
8K	10pJ
32K	20pJ
1M	100pJ
DRAM	1.3-2.6nJ

} On-chip  
→ Off-chip

[arXiv:1602.02830v3]

# Cost of General-Purpose Hardware



Neil C. Thompson, Svenja Spanuth, Communications of the ACM, March 2021

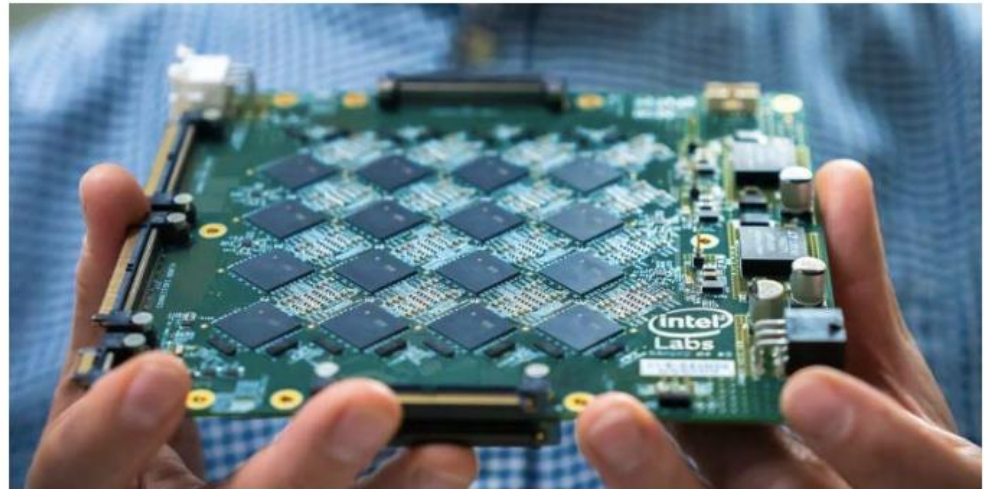
- The second important characteristic is targeting general-purpose microprocessors
- The transistor improves slowly and so does a microprocessor w/o architectural change

# Non Von-Neumann Architecture

- Specialized to a target workload; no more general-purpose hardware
- Uses mostly on-chip memory
- Limited programmability. Sometimes, no program and rely on an FSM
- Distributed data storage on a chip, embedded between logic circuits, to reduce data travel distance

Non von-Neumann architecture  
is also known as

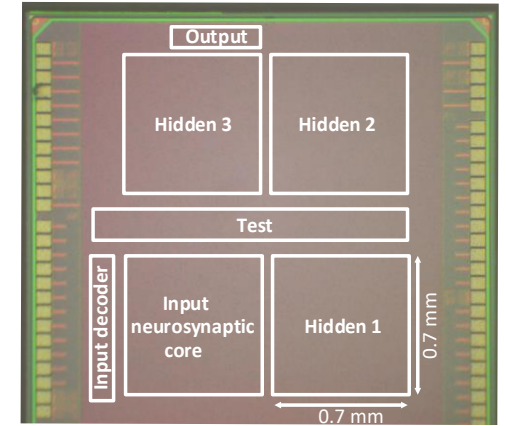
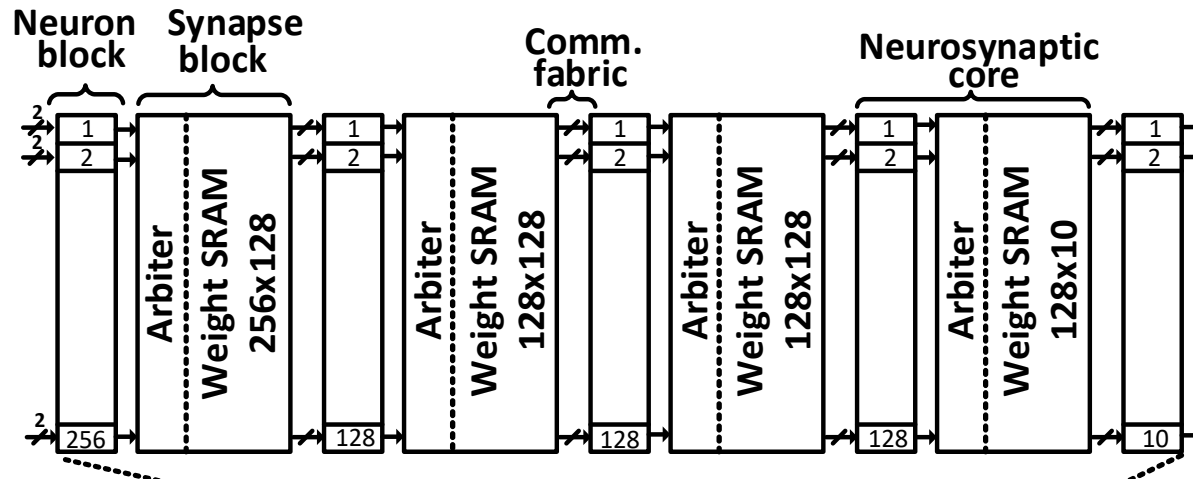
- **Accelerator**
- **ASIC**



[Intel's Nahuku board (8-32 Loihi chips)]



# Spiking Neural Network (SNN) Processor



[Chundi, FiN'21]

- Specialized to support *ONLY* spiking neural network models
- No general purpose
- Uses mostly on-chip memory
- Distributed data storage on a chip, embedded between logic circuits
- It also employs a customized sequencing technique titled spike-event-driven (neither conventional synchronous nor asynchronous)

# Comparison to Prior Works

	This work	Park ISSCC19[5]	Chen VLSI18[15]	TrueNorth[2]
Technology [nm]	65	65	10	28
Neuron count	650	410*	4096	1M
Synapse count	67K	N/A	1M	256M
Area[mm <sup>2</sup> ]	1.99	10.08	1.72	430
Clock frequency	70KHz@0.5V	20MHz	105MHz@0.5V	N/A
MNIST Classification				
Power	305nW	23.6mW	9.42mW**	63mW
Accuracy[%]	97.6	97.8	97.9	97.6***
Throughput [infs/s]	2	100K	N/A	N/A
Energy per inference [nj]	195	236	1,700	N/A
Energy per SOP [pj]	1.5	N/A	3.8	26

\* Input layer not included; \*\* Estimated from neuron's power dissipation

\*\*\* Estimated from Hsin-Pai Cheng et al, IEEE DATE 2017

**[Chundi, FiN'21]**

- Our chip contains ~650 neurons and consumes 300 nW
- A human brain has 86B neurons and consumes about 20W

→ 461 pW/neuron

→ 232 pW /neuron

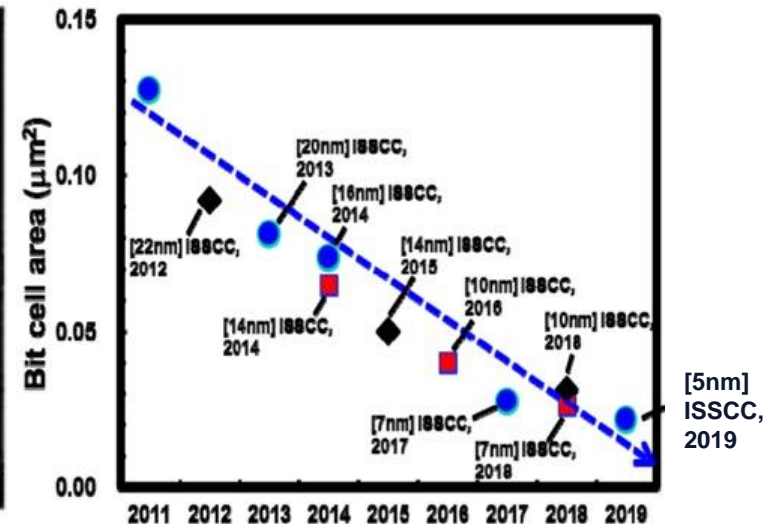
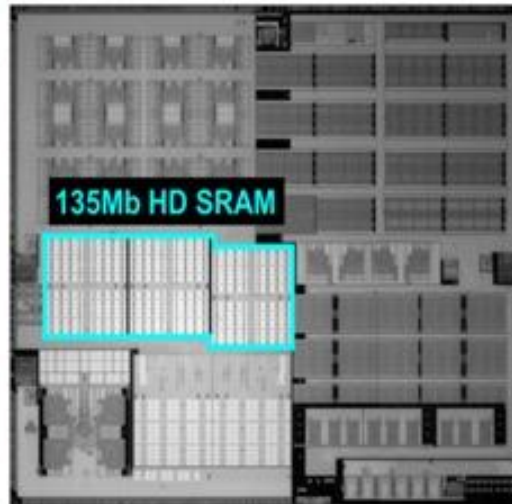
# How to Scale UP?

---

- But the challenge is how to **SCALE UP**
- Scaling up is often limited by the amount of memory we can have on a chip
- Because we can reuse ALUs over time (time-sharing), but we can't time-share data storage

# How Much Memory Can We Integrate on a Chip?

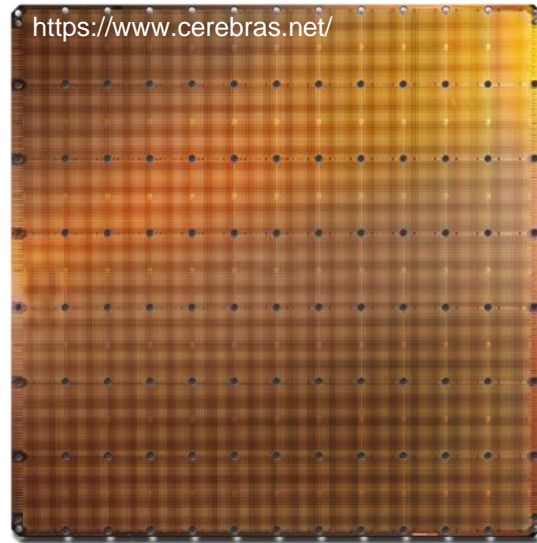
Technology	5nm HK-MG FinFET
Supply voltage	Core: 0.75V IO: 1.2V
Bit cell size	0.021 $\mu\text{m}^2$
SRAM macro configuration	1024x144 MUX4 256 bits/BL, 288 bits/WL
SRAM capacity	135Mb
Test Features	Column Redundancy Programmable E-fuse
Chip size	10mm x 7.98mm = 79.8mm <sup>2</sup>



## ■ Embedded SRAM

- Despite of the tremendous amount of difficulties in technology scaling, SRAM bitcell size has been scaled relatively well in the past ten years
- 22nm: 0.09  $\mu\text{m}^2$   $\rightarrow$  5nm: 0.021  $\mu\text{m}^2$  :: marks 4.2X reduction (linear scaling)
- For a 500-mm<sup>2</sup> chip, theoretically, we can integrate  $\sim 2$  GB of SRAM
- Note that a DRAM per-bit area is: 0.0016  $\mu\text{m}^2$ , which is  $\sim 13$ X smaller than 5nm SRAM
- Yet, for the manufacturing cost wise, 5-nm SRAM is way more expensive than 1X-nm DRAM

# Wafer-Scale Computing



Cerebras WSE

1.2 Trillion transistors  
46,225 mm<sup>2</sup> silicon

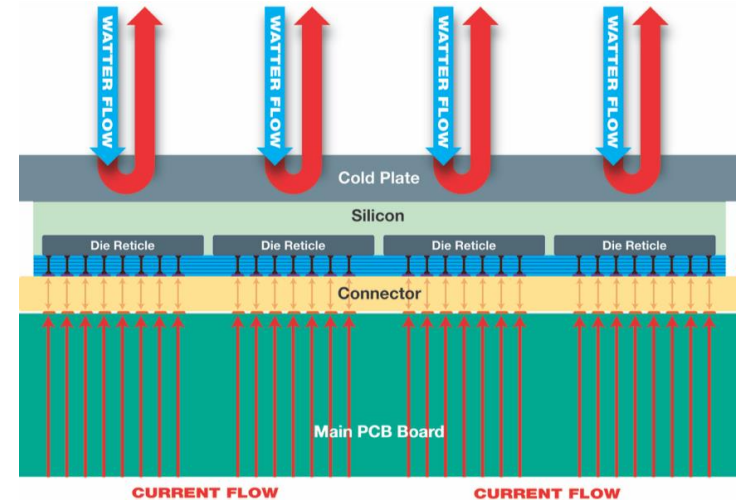
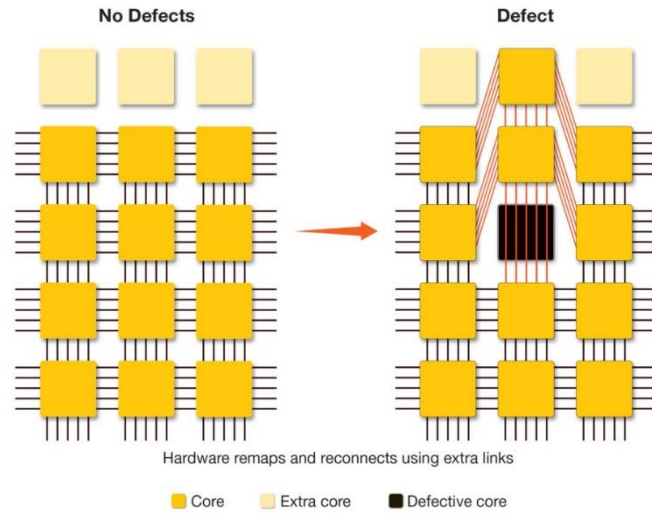


Largest GPU

21.1 Billion transistors  
815 mm<sup>2</sup> silicon

- Building very-large integrated circuit (commonly called a "chip") networks from an entire silicon wafer to produce a single "super-chip"
- 46,225 mm<sup>2</sup> in a 16nm process = 0.05 μm<sup>2</sup>/b X **115 GB SRAM**

# Wafer-Scale Computing: Challenges



<https://www.cerebras.net/>

- Yield
  - Redundancy
  - Post-silicon calibration
- Packaging & board design
  - Power delivery
  - Thermal control

# Emerging Non-Volatile Embedded Memory

Parameters	Typical memory technology			New memory technology				
	SRAM	DRAM	Flash (NAND)	FeRAM	ReRAM	PCRAM	STT-MRAM	SOT-MRAM
Non-volatility	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Cell size ( $F^2$ )	50-120	6-10	5	15-34	6-10	4-19	6-20	6-20
Read time (ns)	$\leq 2$	30	$10^3$	$\approx 5$	1-20	$\approx 2$	1-20	$\leq 10$
Write time (ns)	$\leq 2$	50	$10^6$	$\approx 10$	50	$10^2$	$\approx 10$	$\leq 10$
Write power	Low	Low	High	Low	Medium	Low	Low	Low
Endurance (cycles)	$10^{16}$	$10^{16}$	$10^5$	$10^{12}$	$10^6$	$10^{10}$	$10^{15}$	$10^{15}$
Future scalability	Good	Limited	Limited	Limited	Medium	Limited	Good	Good

Enlong Liu, PhD. Thesis, 2018

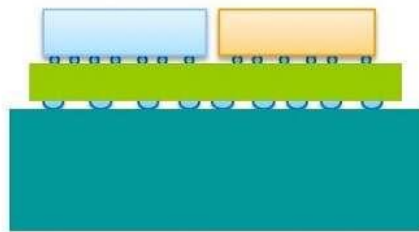
- ReRAM, PCRAM, STT-MRAM, and SOT-MRAM show the promising cell size

However:

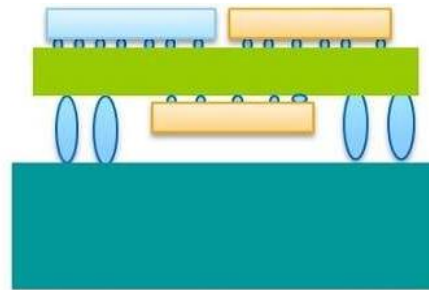
- Some of them are limited in reliability and robustness
- **F** may not be scaled very well
- Price is the other factor to consider



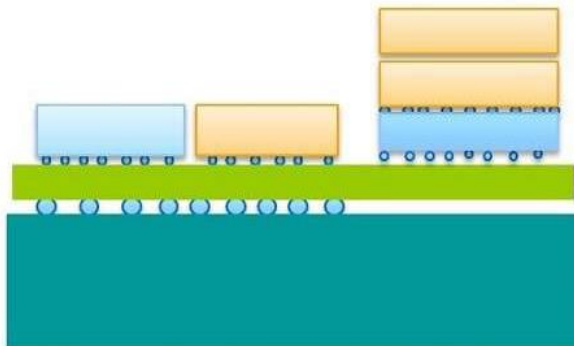
# 3D-IC: Memory on Logic



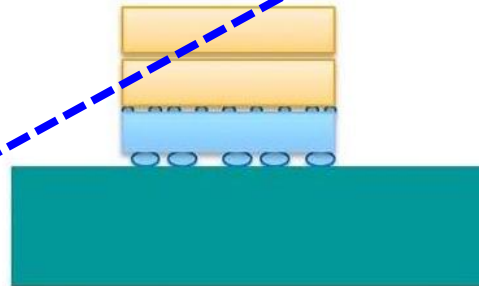
**2.5D:** Side-by-side die stacked on a passive interposer that includes TSVs



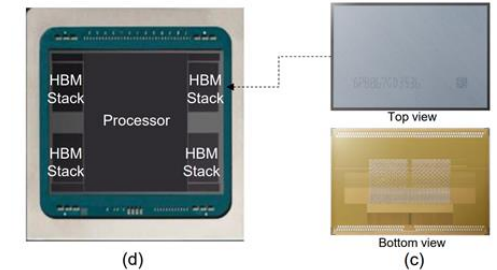
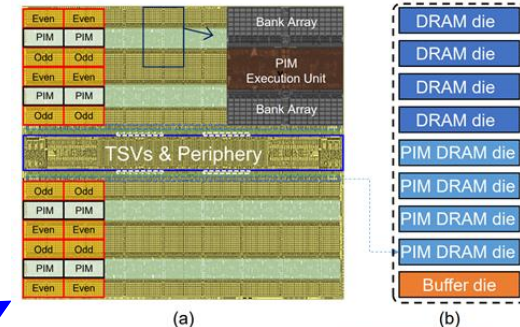
**2.5 or 3D:** Interposer with top and bottom connection



**3D + Interposer:** Mix of side-by-side and stacked implementations on an interposer



**3D Memory on Logic:** One or more DRAM die stacked directly on logic die



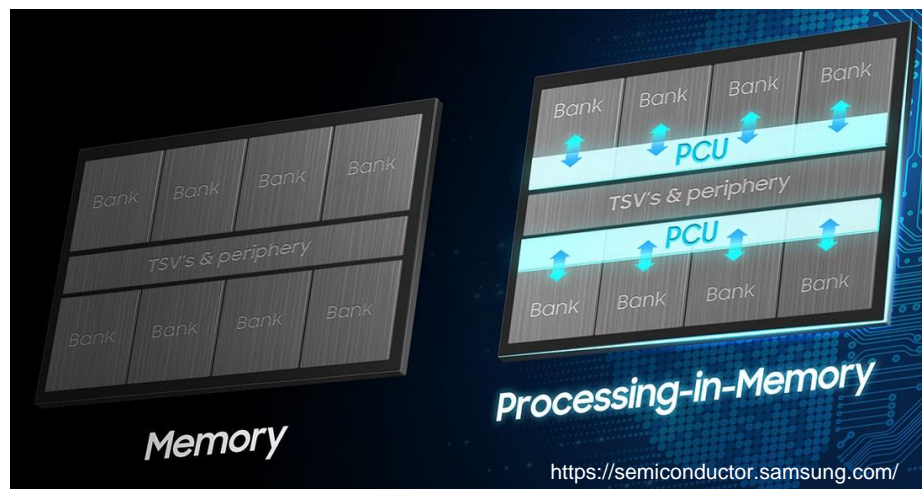
Samsung HBM-PIM  
ISCA'21

Holi grail  
Not commercialized,  
yet

<https://www.eetimes.com/is-3d-ic-the-next-big-profit-driver/>



# PIM – Processing In Memory



- Adding computing hardware on a DRAM die
  - Some of the computing tasks are off-loaded to a PIM
  - PIM sends pre-computed results to the CPU/GPU
- 2X performance improvement, 3X less energy efficiency consumption
- Good but not impressive (sorry)

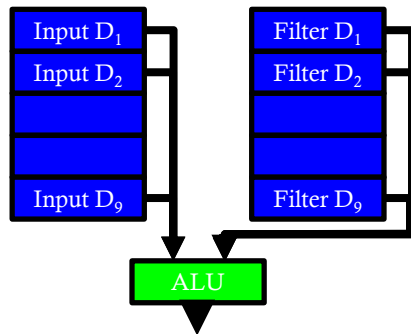
# PIM: Challenges

	Pre-PIM	PIM	Future?
CPU/GPU	100%	50%	0%
DRAM	0%	50%	100%

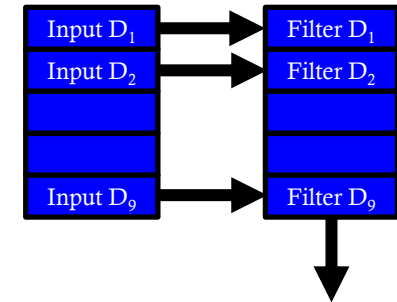
- Good but not impressive (sorry); WHY?
  - Dividing a task in a way that the divided parts do not communicate with each other much is not simple (probably impossible)
  - You end up still needing to move data, and it still looks like a significant amount
- Future PIM?
  - It should perform a single task for itself without talking to CPU/GPU
  - DRAM-based GPU??
  - Bigger chip or multiple chiplets (DRAM die size is  $\sim 25 \text{ mm}^2$ , 20X smaller than a CPU/GPU core)
  - Need to address the DRAM FET's performance

# SRAM-based In-Memory Computing

Multiply and Accumulations (MACs) =  $D_1 \cdot W_1 + D_2 \cdot W_2 + \dots D_9 \cdot W_9$

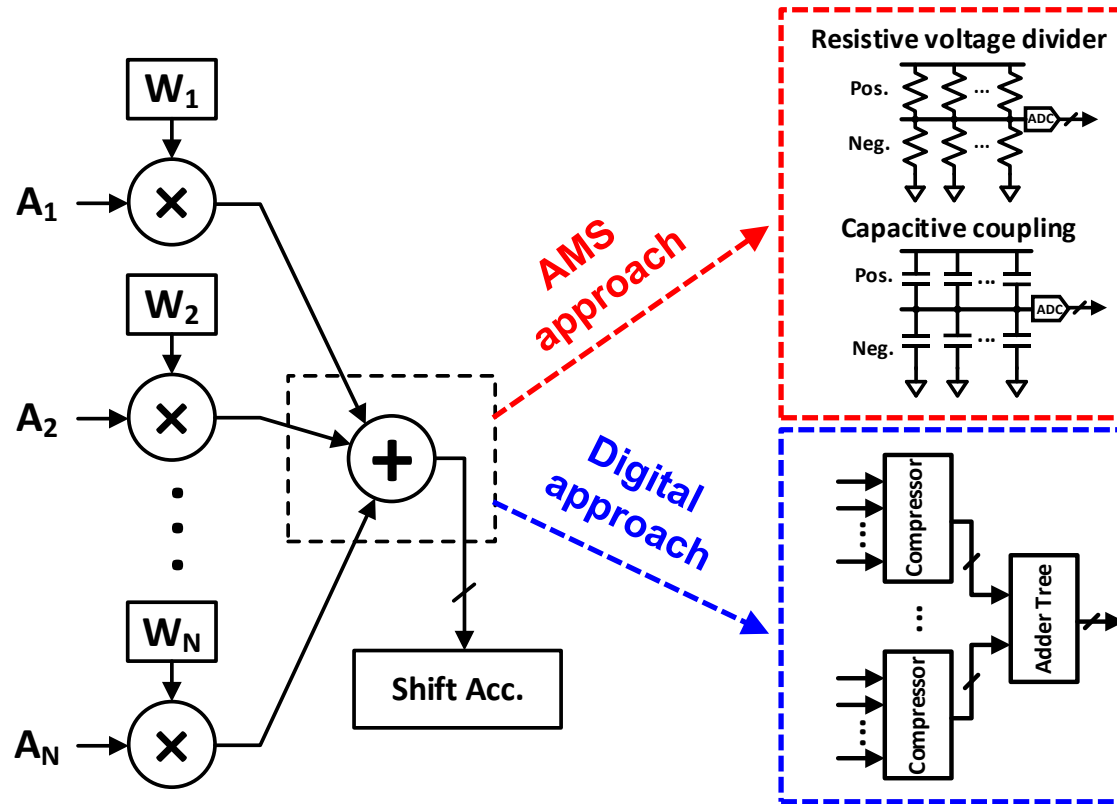


Row-by-row access: **9 cycles**



In-memory computing: **1 cycle**

# Logical View of a Column of SRAM-based IMC HW



# Logic Design

# Design Process

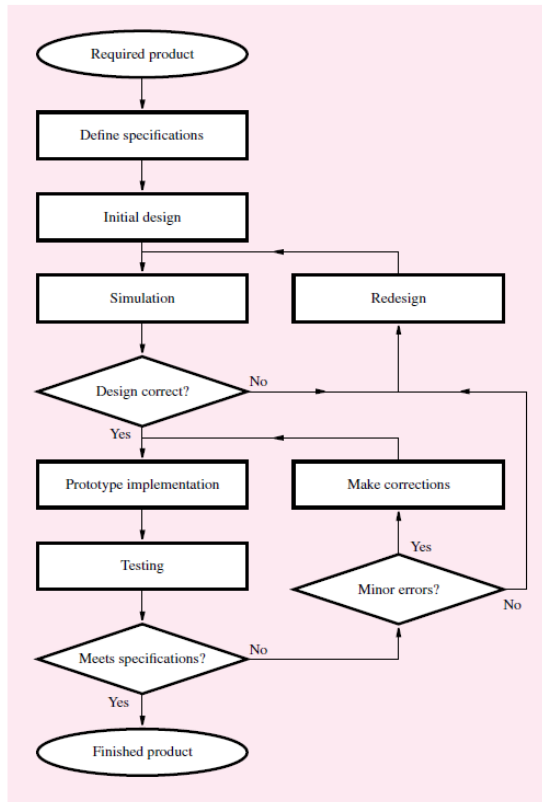
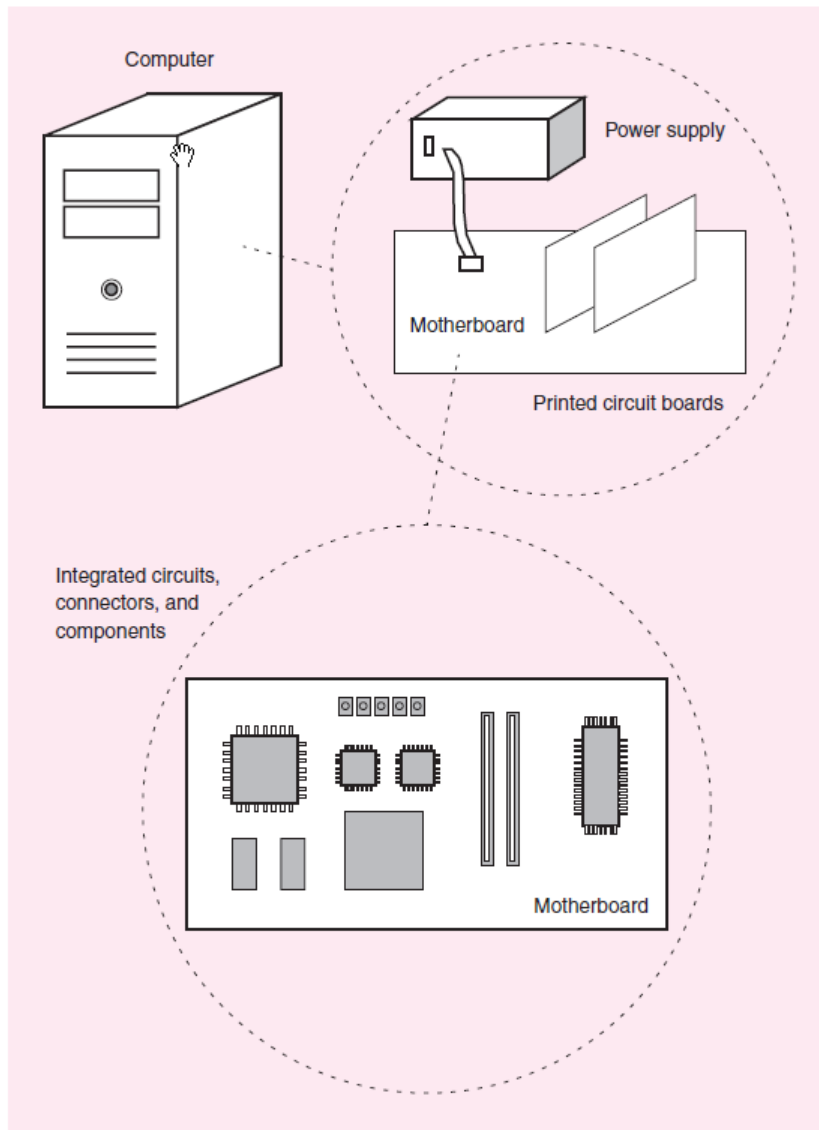
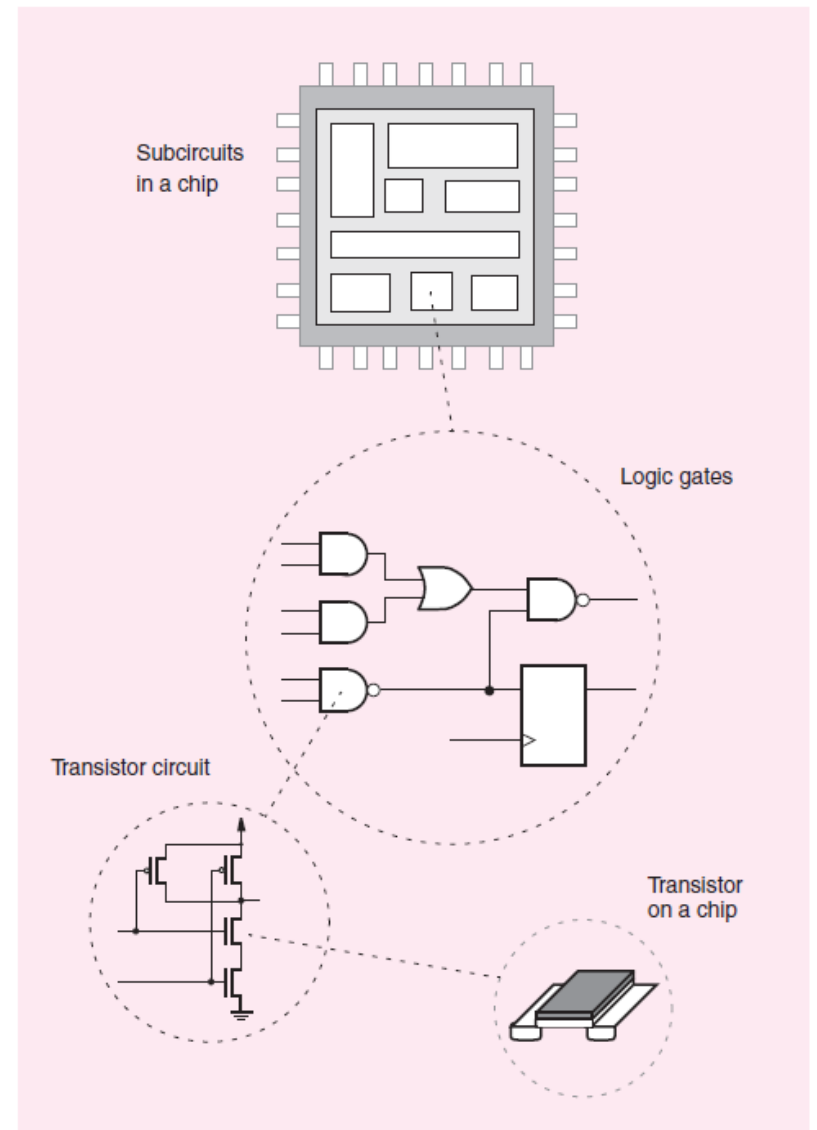


Figure 1.3 The development process.

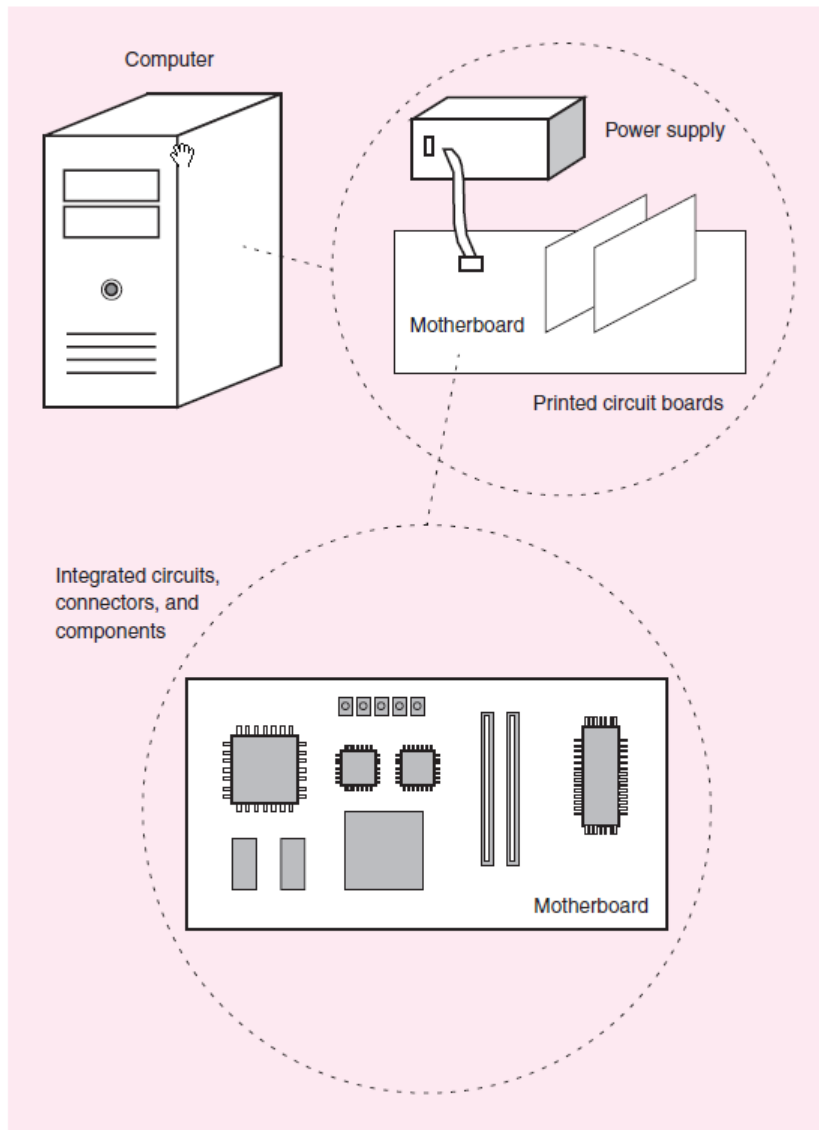
- Computer-Aided Design (CAD) tools are available in almost every step of the design process
- Expensive process



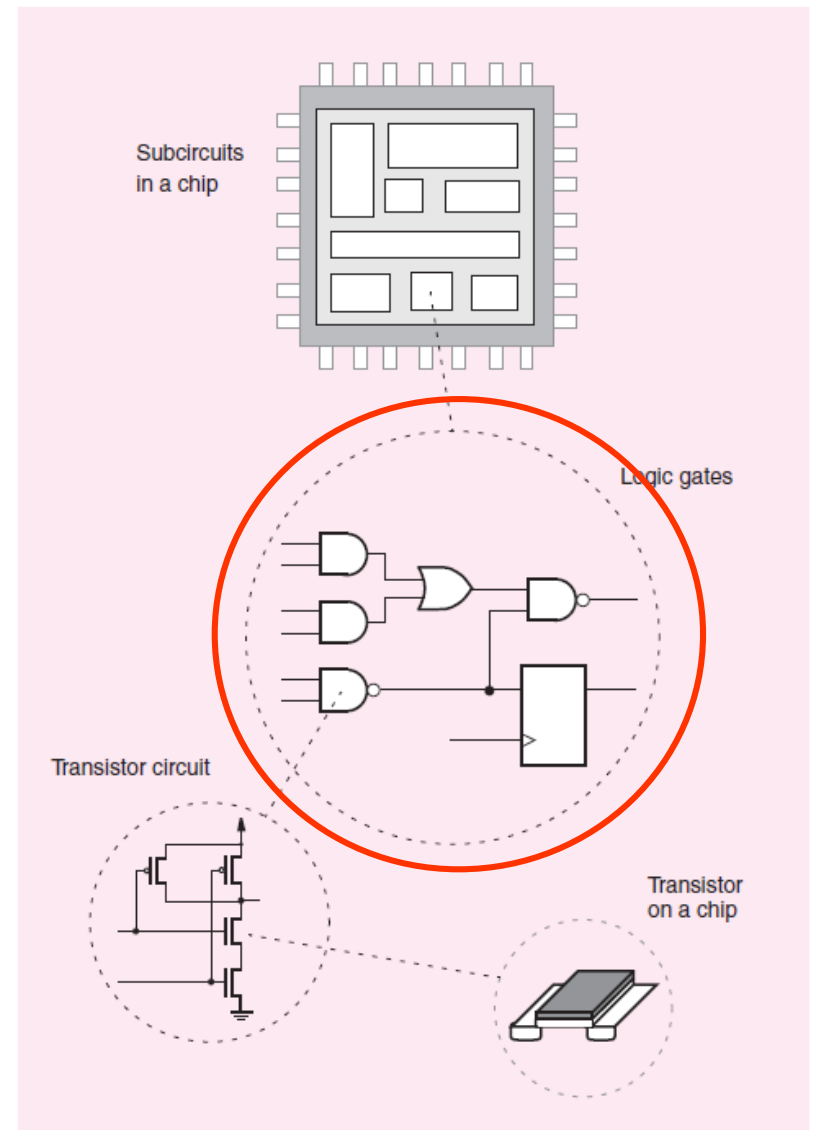
**Figure 1.4** A digital hardware system (Part a).



**Figure 1.4** A digital hardware system (Part b).



**Figure 1.4** A digital hardware system (Part a).



**Figure 1.4** A digital hardware system (Part b).



# Binary Numbers

Decimal representation	Binary representation
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

- Using  $n$  bits allows representation of positive integers in the range 0 to  $2^n - 1$
- Least-Significant Bit (LSB) and Most-Significant bit (MSB)
- Nibble: 4 bits
- **Byte: 8 bits**
- Negative number?
- Decimal point?

Convert  $(857)_{10}$

				Remainder	
$857 \div 2$	$=$	428	1		LSB
$428 \div 2$	$=$	214	0		
$214 \div 2$	$=$	107	0		
$107 \div 2$	$=$	53	1		
$53 \div 2$	$=$	26	1		
$26 \div 2$	$=$	13	0		
$13 \div 2$	$=$	6	1		
$6 \div 2$	$=$	3	0		
$3 \div 2$	$=$	1	1		
$1 \div 2$	$=$	0	1		MSB

Result is  $(1101011001)_2$

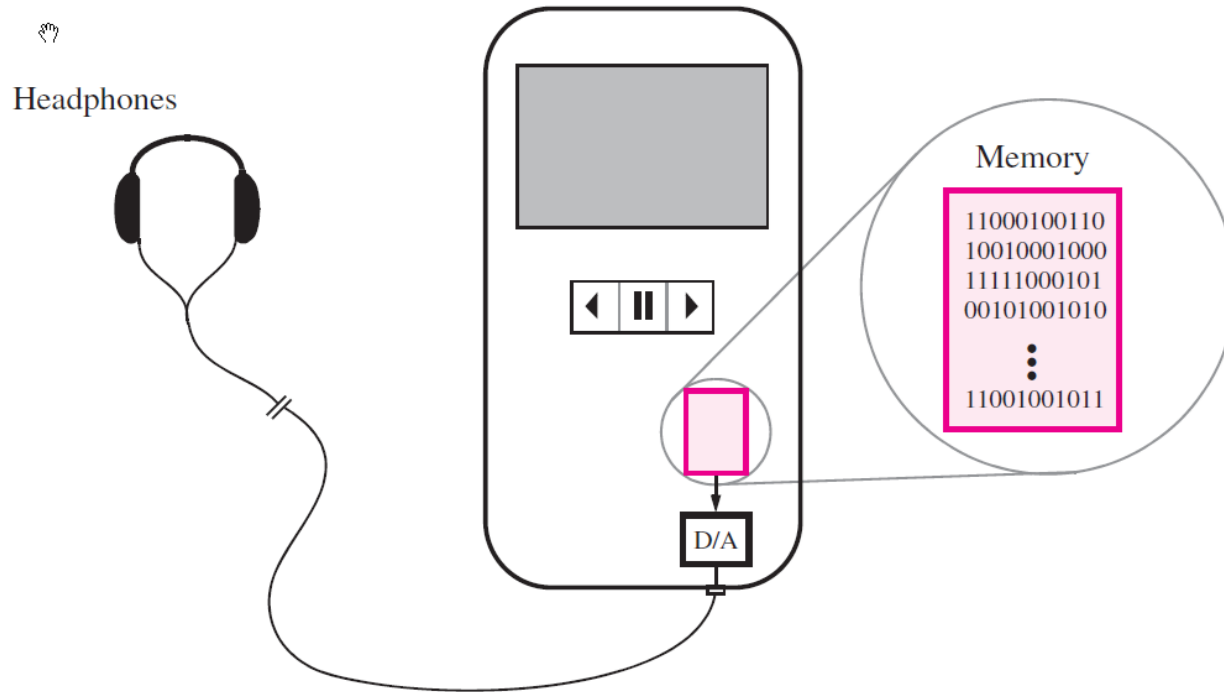
# ASCII Character Code

**Table 1.2** The seven-bit ASCII code.

Bit positions 3210	Bit positions 654														
	000	001	010	011	100	101	110	111							
0000	NUL	DLE	SPACE	0	@	P	^	p							
0001	SOH	DC1	!	1	A	Q	a	q							
0010	STX	DC2	"	2	B	R	b	r							
0011	ETX	DC3	#	3	C	S	c	s							
0100	EOT	DC4	\$	4	D	T	d	t							
0101	ENQ	NAK	%	5	E	U	e	u							
0110	ACK	SYN	&	6	F	V	f	v							
0111	BEL	ETB	'	7	G	W	g	w							
1000	BS	CAN	(	8	H	X	h	x							
1001	HT	EM	)	9	I	Y	i	y							
1010	LF	SUB	*	:	J	Z	j	z							
1011	VT	ESC	+	;	K	[	k	{							
1100	FF	FS	,	<	L	\	l								
1101	CR	GS	-	=	M	]	m	}							
1110	SO	RS	.	>	N	^	n	~							
1111	SI	US	/	?	O	—	o	DEL							
NUL	Null/Idle		SI		Shift in										
SOH	Start of header		DLE		Data link escape										
STX	Start of text		DC1-DC4		Device control										
ETX	End of text		NAK		Negative acknowledgement										
EOT	End of transmission		SYN		Synchronous idle										
ENQ	Enquiry		ETB		End of transmitted block										
ACQ	Acknowledgement		CAN		Cancel (error in data)										
BEL	Audible signal		EM		End of medium										
BS	Back space		SUB		Special sequence										
HT	Horizontal tab		ESC		Escape										
LF	Line feed		FS		File separator										
VT	Vertical tab		GS		Group separator										
FF	Form feed		RS		Record separator										
CR	Carriage return		US		Unit separator										
SO	Shift out		DEL		Delete/Idle										
Bit positions of code format = <table><tr><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>									6	5	4	3	2	1	0
6	5	4	3	2	1	0									

- American Standard Code for Information Interchnage
- Represent 128 characters in a computer
  - Alphabets
  - Punctuation marks
  - Control characters
- Each need 7 bits but it fits in a Byte with the additional 1 bit for error correction

# Digital & Analog Information



- ADC: convert analog to digital signals
- DAC: convert digital to analog signals

End of the Slides