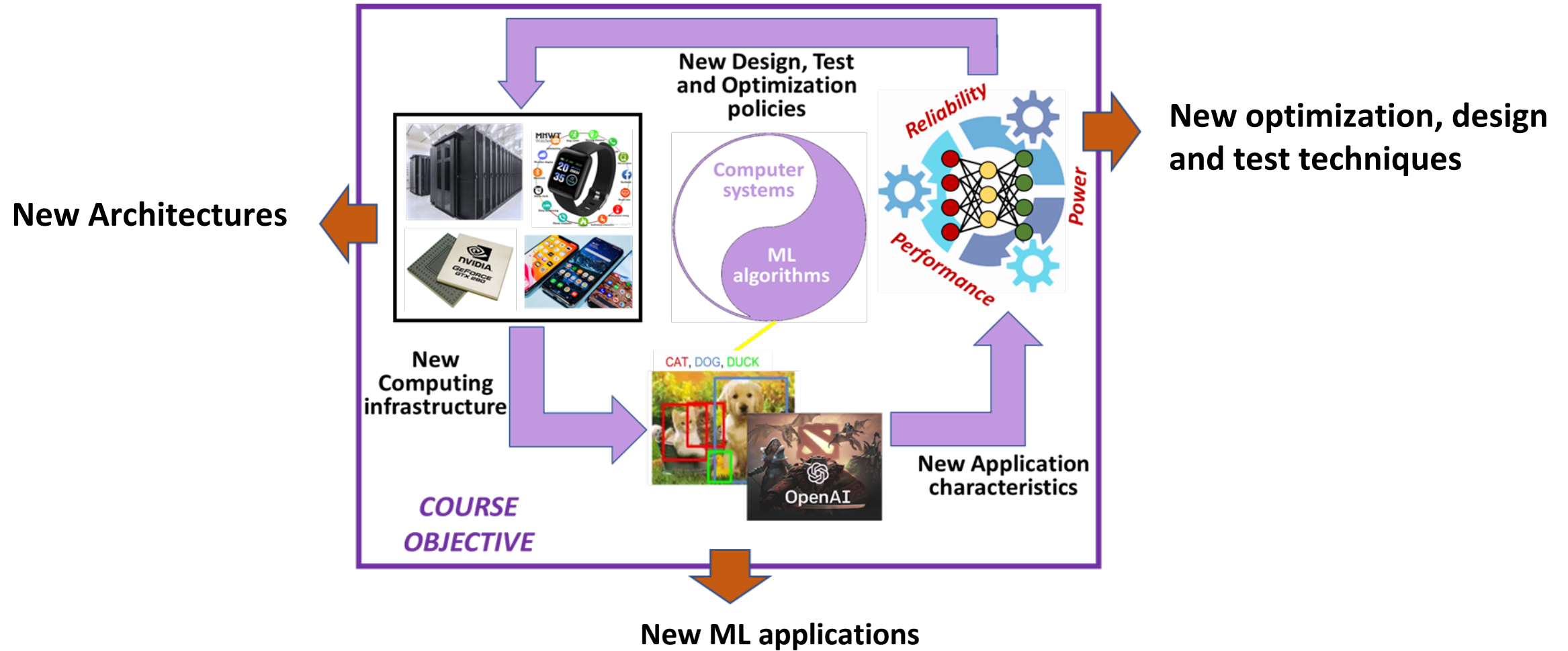


ECE 590: Machine Learning-based hardware design methodologies

Course summary



About the course

- What will be covered?
 - Machine Learning (ML) algorithms such as Random Forest, Neural Networks, etc.
 - ML-based methodologies to improve power-performance trade-offs in hardware systems.
 - Novel hardware designs such as “3D architectures” to accelerate Deep Learning.
- For Students:
 - This course is suited for graduate students and undergraduate students with interest in Hardware systems and Machine Learning.
- Prerequisites
 - ECE 350 (for undergrads), ECE 550 (for graduates), ECE 529 preferred but not compulsory; If you have not taken these courses, you can still attend by permission of instructor
 - Experience with coding in C/C++, Python
 - Prior ML knowledge not required

Outline

- ML Intro (Weeks 1-2)
 - Training/Inference
- ML-based Hardware design methodology (Weeks 3-12)
 - Task Mapping
 - On-chip communication
 - Power management
 - 2.5D/3D architectures
 - Analog design
 - Hardware security
- Project (Weeks 12-13)
 - Individual paper review and presentation

Week by Week breakdown

- Week-1: Familiarization with ML libraries and algorithms
- Week-2: Implementing ML tasks
- Week-3: On-chip communication
- Week-4: Task-mapping
- Week-5: ML for task mapping and on-chip communication
- Week-6: 2.5D and 3D architectures
- Week-7: Power management
- Week-8: Paper presentation
- Week-9: ML for power management
- Week-10: Analog Design
- Week-11: Hardware Security, Rowhammer
- Week 12: ML for analog design and security
- Week-13: Final Project on hardware design using ML

Homework & late policy

- 4 Homework, 16 points each
 - Machine Learning
 - Task mapping & NoC design
 - 3D/2.5D architectures & power
 - Analog design & Security
- Pdf and code can be submitted by email/Dropbox
 - Live demo for code
- Bonus points for open questions
- Late policy:
 - 7 days without any penalty for entire semester
 - No points after that
 - Not applicable for project and presentations

Class projects and presentations

- 1 project, 36 points
 - Research paper published within last 4 years at top-tier conference/journal in ML or EDA
 - 4-6 pages IEEE-style report
 - Survey, Identify research problem, Propose new solution
- Novel reports automatically get 'A' (New solution, Interesting results, New direction of research)
 - No extension/late policy for report
 - Pdf and code can be submitted by Dropbox/email
 - Live code demonstrations in class
- Presentations:
 - Project introductions (10-15 mins)
 - Final project (10-15 mins)
 - Guest lectures

Course references and books

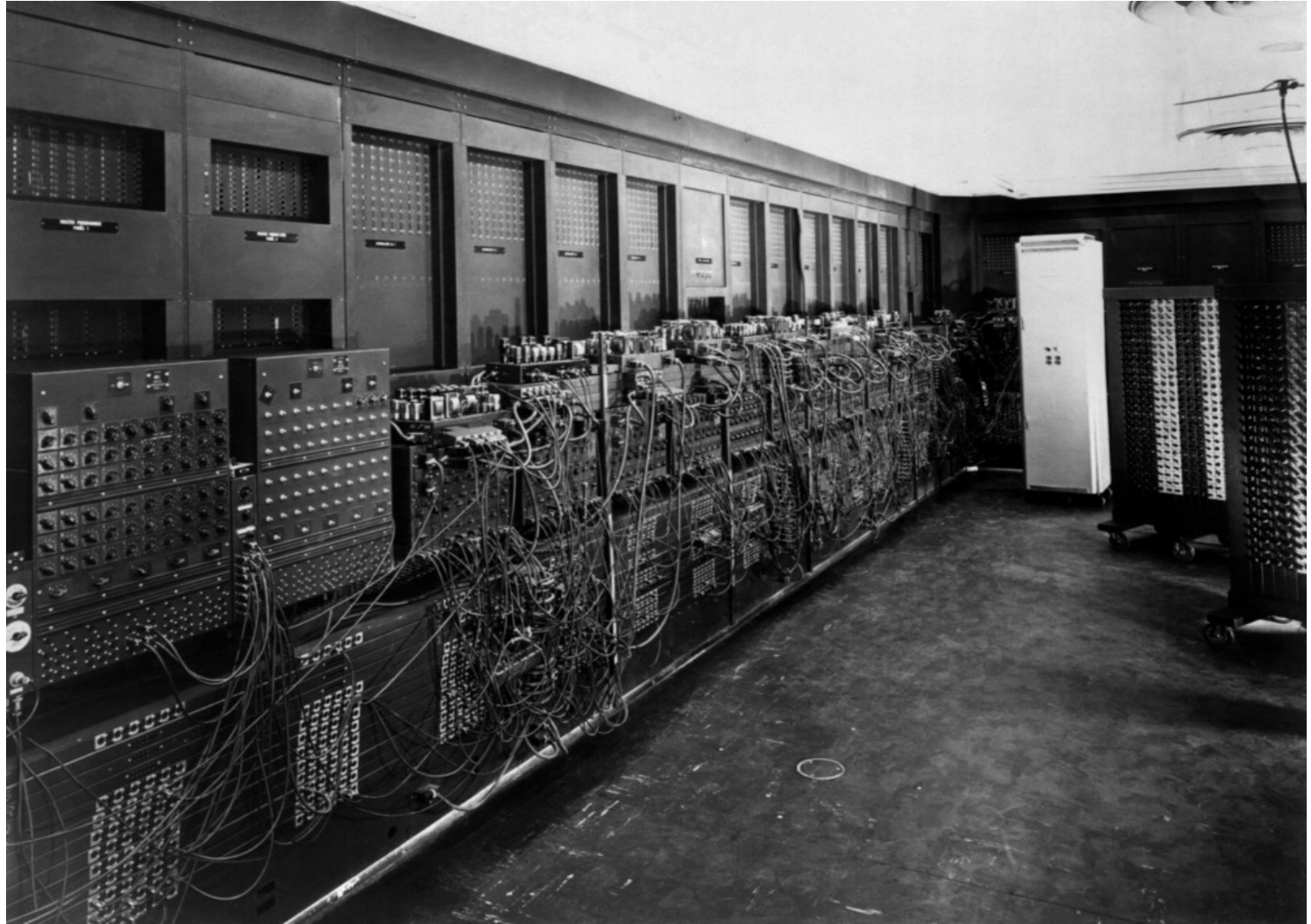
- Mostly research papers
 - DATE, DAC, ICCAD, etc.
- Follow class slides
- Website: Sakai and <https://bjoardar.github.io/course.html>
- Books
 - <https://www.amazon.com/Deep-Learning-Hardware-Design-2nd/dp/B085K85SNW>
 - https://www.google.com/books/edition/Machine_Learning_in_VLSI_Computer_Aided/2jeNDwAAQBAJ?hl=en&gbpv=0
- Office hours
 - Wed 4-5PM? (after class?)
 - Available by email

Computing resources & Feedback

- Computing resources:
 - CPU: Duke virtual machines (includes sudo permissions)
 - GPU: GPU Scavenger container from Duke OIT (shared by other class)
- Participate in class discussions, Q&A
- Feedback at the end of the course & during class

Computers in the past: ENIAC

- Year: 1946
- 17,468 vacuum tubes, 70,000 resistors, 10,000 capacitors, 1,500 relays, 6,000 manual switches and 5 million soldered joints.
- 1,800 square feet (167 square meters) of floor space
- weighed 30 tons
- 160 kilowatts of electrical power.



Computers in the past: Apollo guidance computer

- Year: 1960s
- 32,768 bits of RAM memory
- Clock speed 2.048 MHz
- Weighed 70 lb (32 kg)
- Consumed 55W



https://en.wikipedia.org/wiki/Apollo_Guidance_Computer

Computers: Today



Datacenters



PCs



Smartphones



**Smartwatches
/Wearables**

- **Many sizes and shapes**
- **Clock speed: Up to 4 GHz**
- **Power: Often less than 1W in smartphones and wearables**
- **Weight: less than a pound to several pounds**

Computers: Future

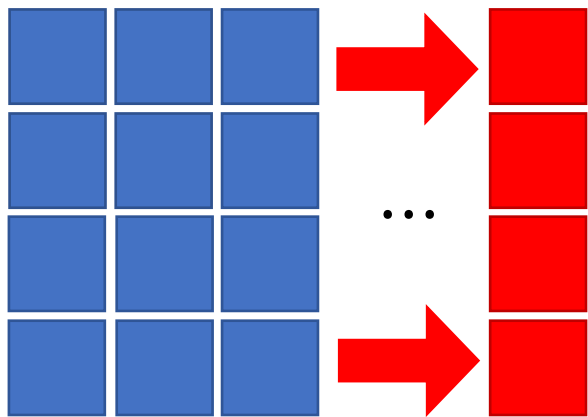


- **Datacenter-on-Chip**
- Target performance: Several TFLOP/s
- Ultra low power
- Reliable, Secure

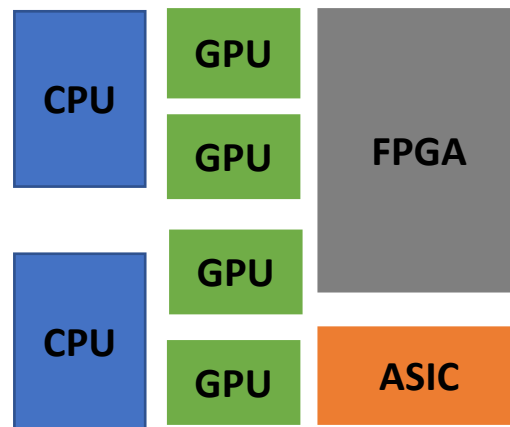
Why Machine Learning?

- Hardware Design is becoming more complex
 - Larger system size
 - More heterogeneity
 - New design constraints and objectives
- Existing methods do not scale well with design complexity

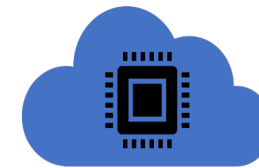
ML inspired techniques present a promising direction!



System size increases



Heterogeneity increases



Cloud: Performance



Embedded System: Real-time constraints, Reliability, Low-Power

No. of design objectives increases

Increasing system sizes

Product name	Intel Pentium IV	Intel I5-3570K	Nvidia GTX 1080Ti	Nvidia Tesla V100	Cerebras WSE	Cerebras WSE-2
Release year	2000	2012	2017	2017	2019	2021
Die size (mm ²)	217	160	471	815	46225	46225
#Transistors	42M	1.4B	12B	21B	1.2T	2.6T



- Design methodologies used for single core no longer applicable
- Pentium IV → Nvidia Tesla, ~400X transistors
- Design and testing complexities are higher

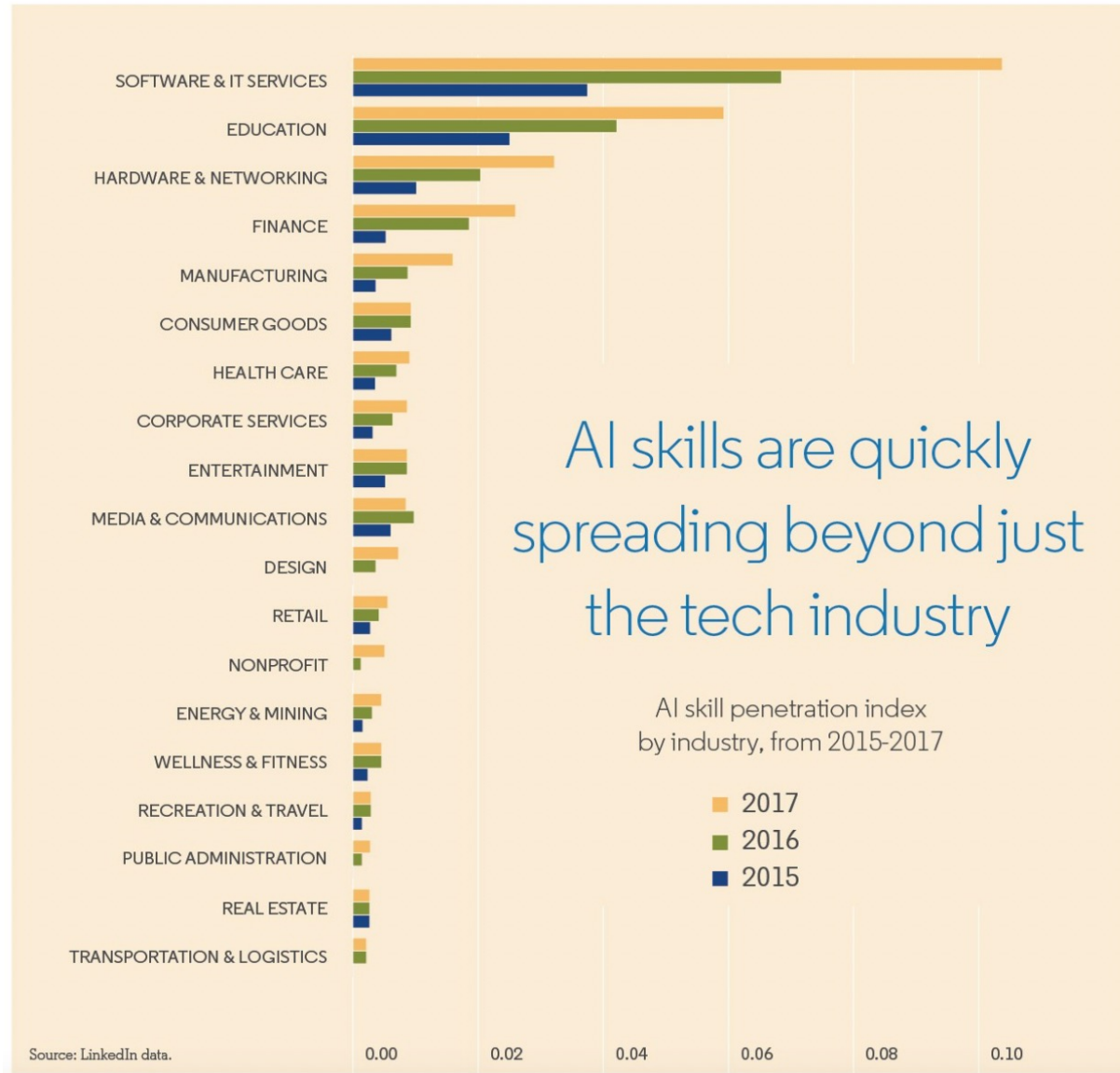


ML in EDA: research



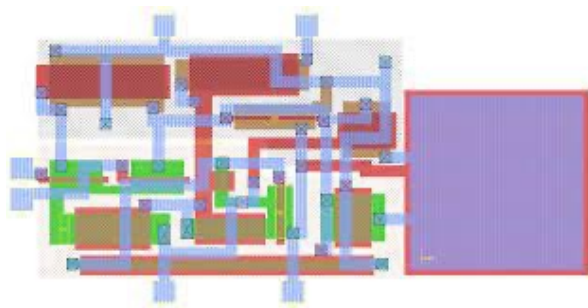
- Top EDA conference
- ~20-25% had ML-related keywords in title (2019)

ML in EDA: industry

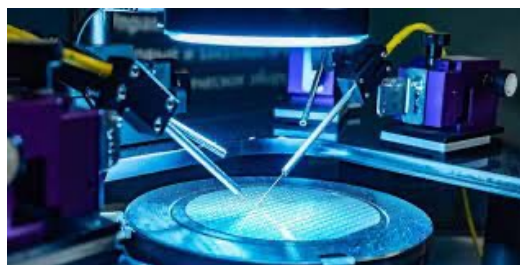


- AI skills are among the fastest-growing skills on LinkedIn, and saw a 190% increase from 2015 to 2017
- Industries with more AI skills present among their workforce are also the fastest-changing industries

What can you do with ML?



Design



Testing

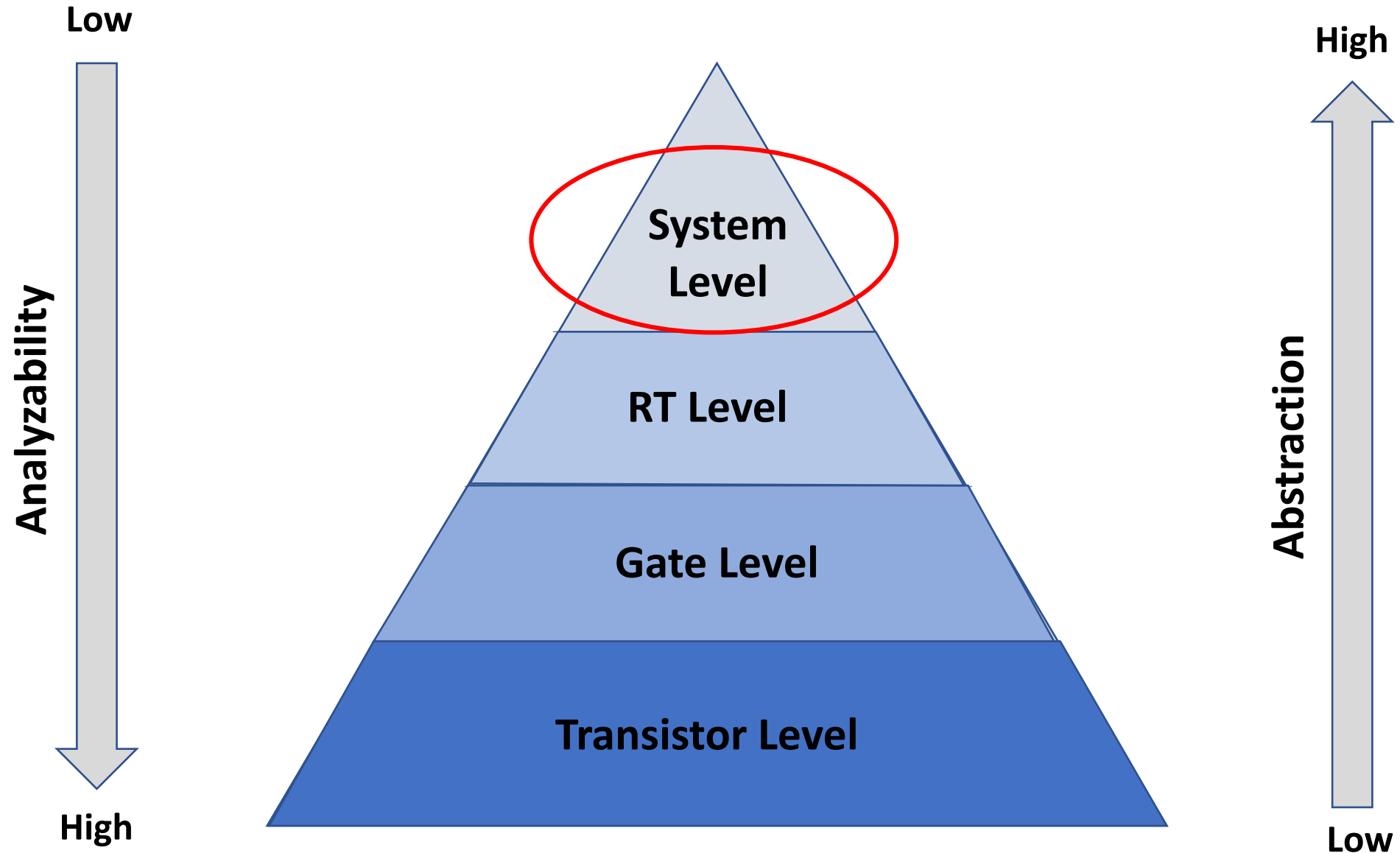


Security

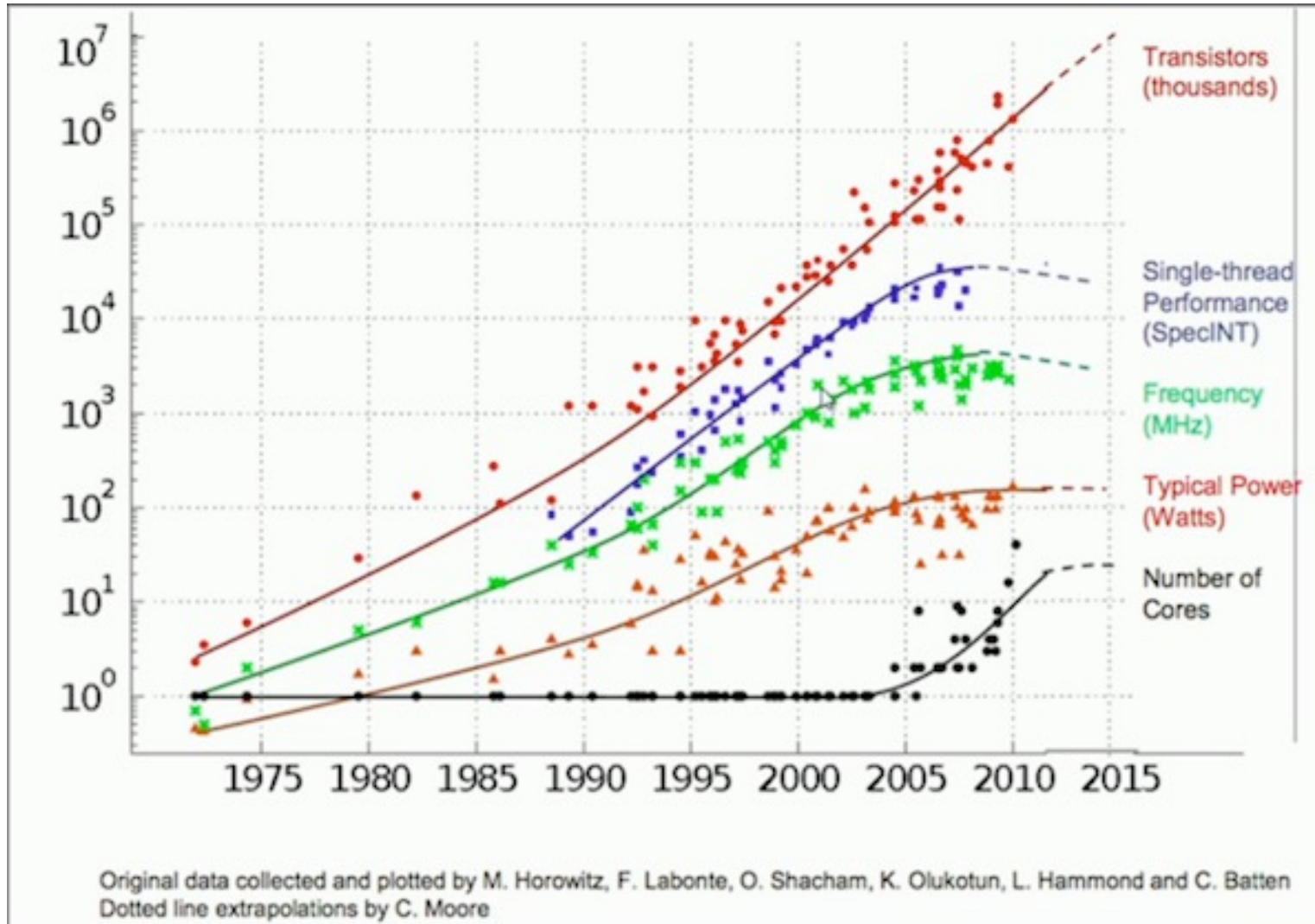


Optimization

Our focus



Hardware Design Trends



- Transistors still increasing
- Single-thread performance slowed/saturated
 - Frequency wall
 - Power wall
- Trend towards **Manycore systems**

Intel's Tick-Tock model

Every microarchitecture change (tock) was followed by a die shrink of the process technology (tick).

Change (step)	Fabrication process	Micro-architecture
Tick (new fabrication process)	65 nm	P6, NetBurst
Tock (new micro-architecture)		Core
Tick	45 nm	Nehalem
Tock		
Tick	32 nm	Sandy Bridge
Tock		
Tick	22 nm ^[19]	Haswell
Tock		
Refresh		

Refresh	14 nm ^[19]	Haswell	Haswell Refresh, Devil's Canyon ^[25]
Tick		Skylake ^[26]	Broadwell ^[26]
Tock			Skylake ^[26]
Optimizations (refreshes) [4][29][30][31]			Kaby Lake ^[32]
			Kaby Lake R ^{[35][36]}
			Coffee Lake
			Whiskey Lake, Amber Lake ^[39]
			Comet Lake ^[40]

Real-world manycore examples

8-core CPU

The highest-performance CPU we've ever built.

Up to **3.5x** faster CPU performance¹

M1
8-core CPU

TESLA V100

21B transistors
815 mm²

80 SM
5120 CUDA Cores
640 Tensor Cores

16 GB HBM2
900 GB/s HBM2
300 GB/s NVLink



*full GV100 chip contains 84 SMs

INTEL® XEON® SCALABLE PROCESSORS

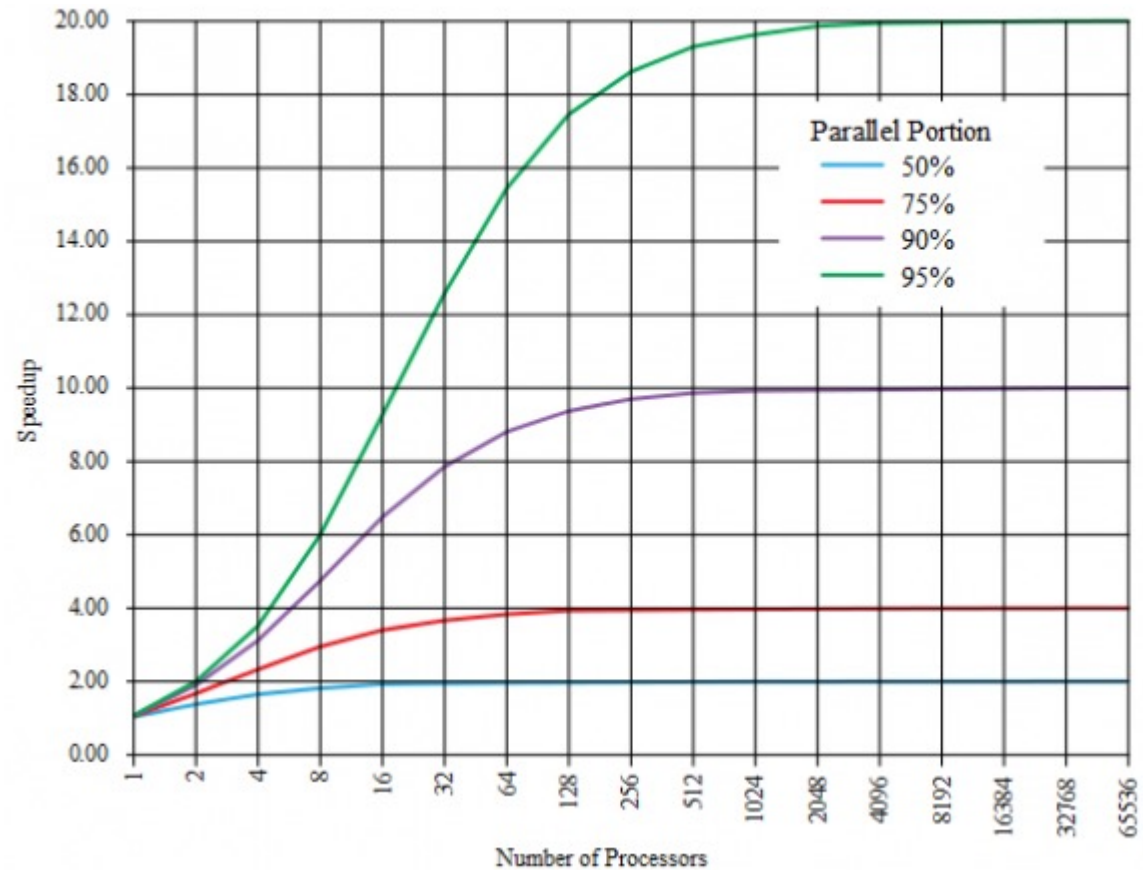
THE FOUNDATION FOR AGILE, SECURE WORKLOAD-OPTIMIZED HYBRID CLOUD

BEST	GREAT	GOOD	ENTRY
<p>UP TO 28 CORES</p> <p>UP TO 2, 4 & 8 SOCKET SUPPORT WITH UP TO 3 UPI LINKS</p> <p>DDR4 2666 MHz WITH UP TO 1.5 TB TOPLINE MEMORY CHANNEL BANDWIDTH</p> <p>HIGHEST ACCELERATOR THROUGHPUT</p>	<p>UP TO 22 CORES</p> <p>2 & 4 SOCKET SUPPORT</p> <p>UP TO 3 UPI LINKS</p> <p>ADVANCED RELIABILITY, AVAILABILITY AND SERVICEABILITY</p>	<p>SCALABLE PERFORMANCE AT LOW POWER STANDARD RAS</p> <p>MODERATE TASKS</p> <p>INTEL® TURBO BOOST TECHNOLOGY AND INTEL® HYPER-THREADING TECHNOLOGY FOR MODERATE WORKLOADS</p>	<p>SCALABLE PERFORMANCE HARDWARE-ENHANCED SECURITY STANDARD RAS</p> <p>LIGHT TASKS</p> <p>ENTRY PERFORMANCE, PRICE SENSITIVE FOR LIGHT WORKLOADS</p>
MAINSTREAM	EFFICIENT	ENTRY	

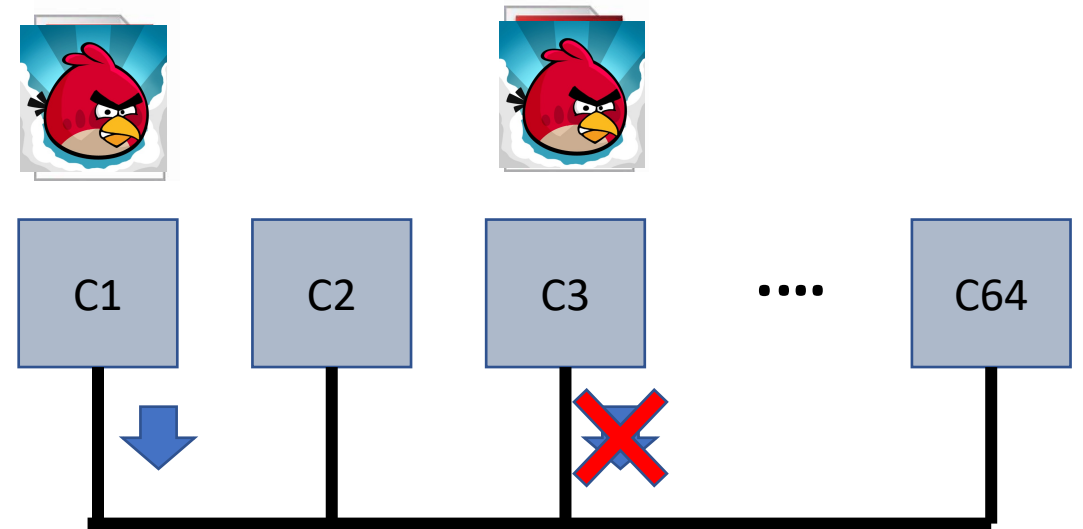
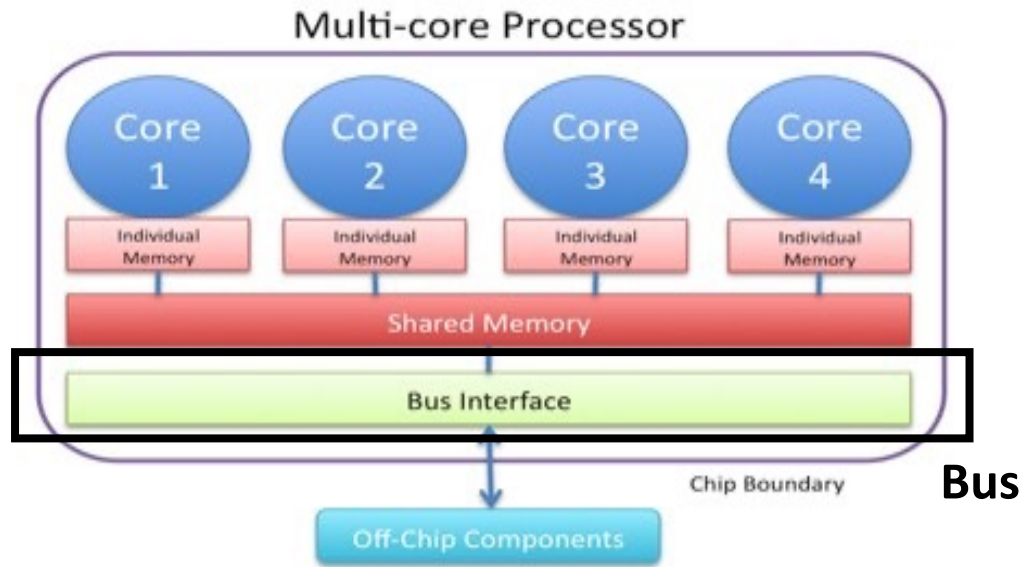
Press Workshops – June 2017 Content Under Embargo Until 9:15 AM PST July 11, 2017 intel | 18

Manycore challenges: Software

- #Cores \neq Exec. Time speedup
- Law of diminishing returns
- Some portions of applications are serial
 - Atomic operations
 - I/O operations

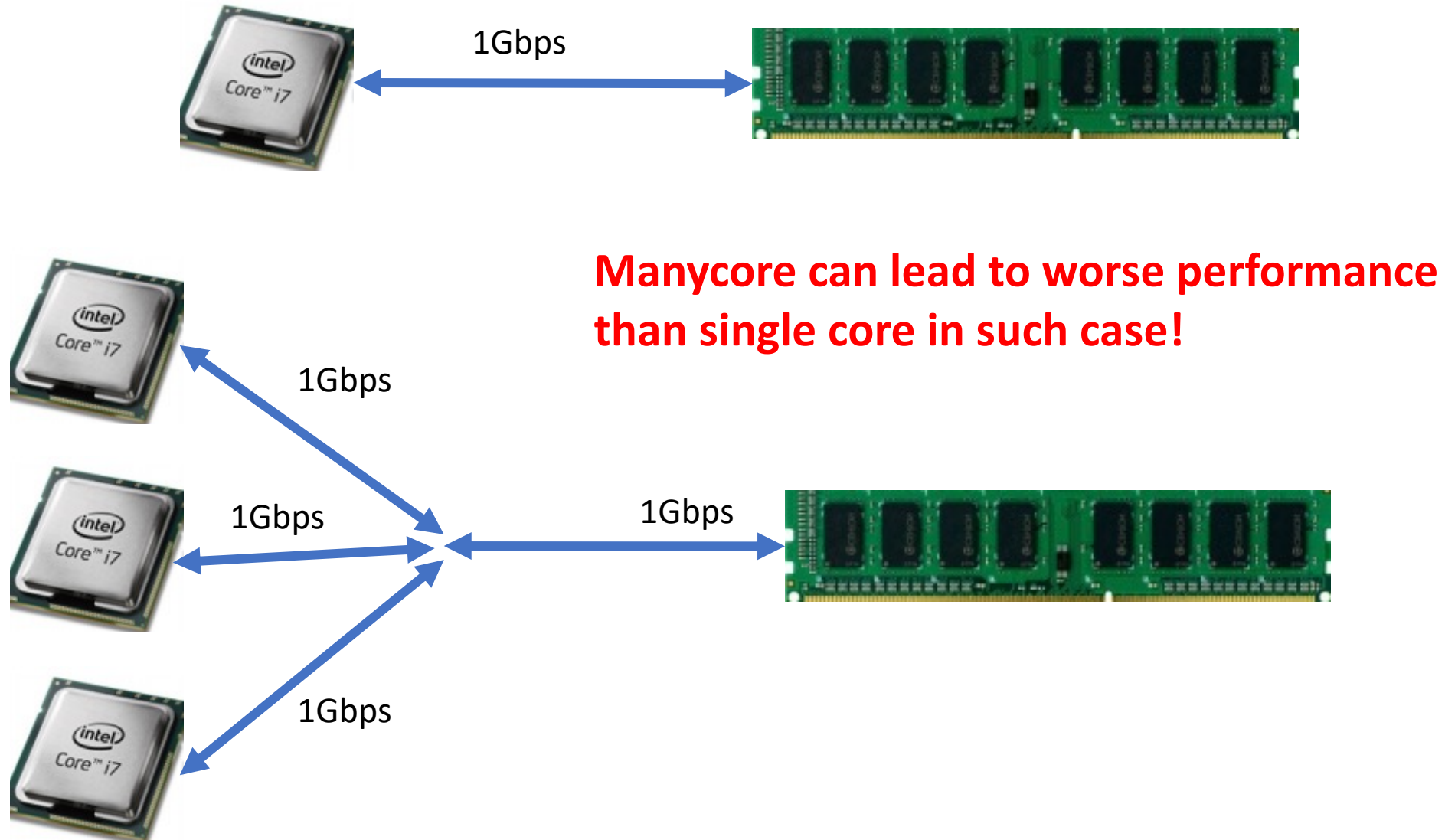


Manycore systems: Communication bottleneck

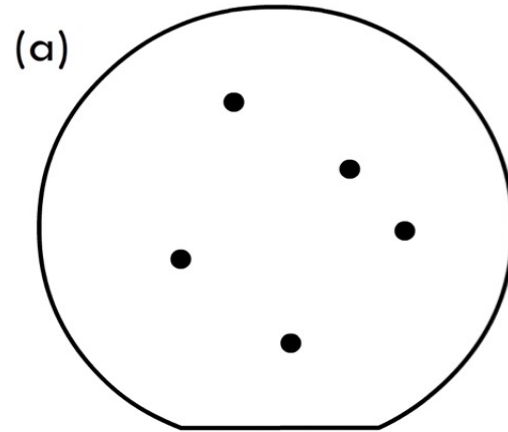


- Conflicts happen in shared communication medium
- Poor performance when #Cores larger
- Poor performance when application is communication heavy

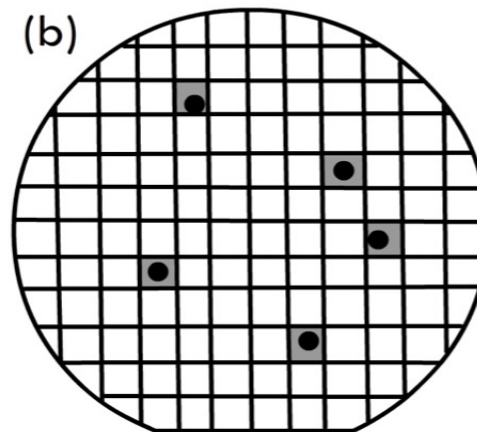
Manycore Systems: Memory Bandwidth bottleneck



Manycore Systems: Manufacturability

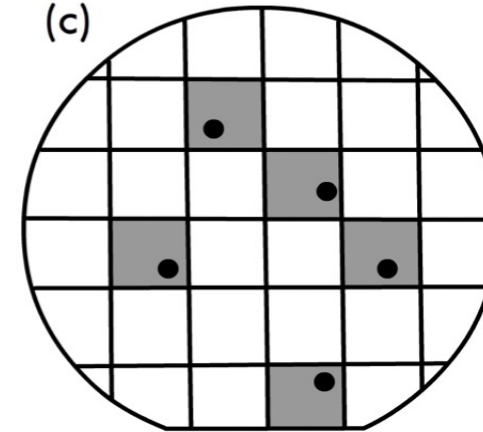


Defects = 5



Defects = 5

$$\text{Yield} = \frac{138 - 5}{138} = 96\%$$

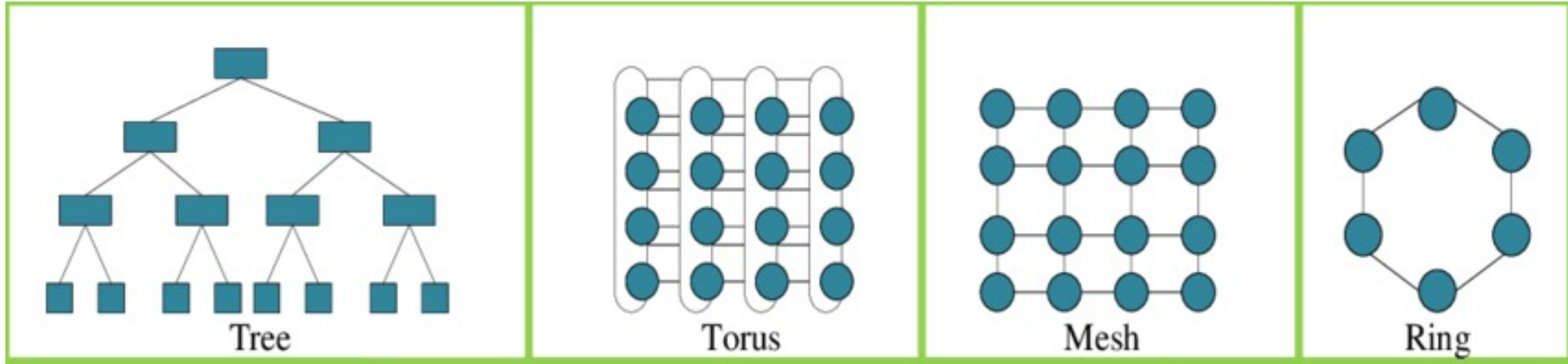


Defects = 5

$$\text{Yield} = \frac{16 - 5}{16} = 69\%$$

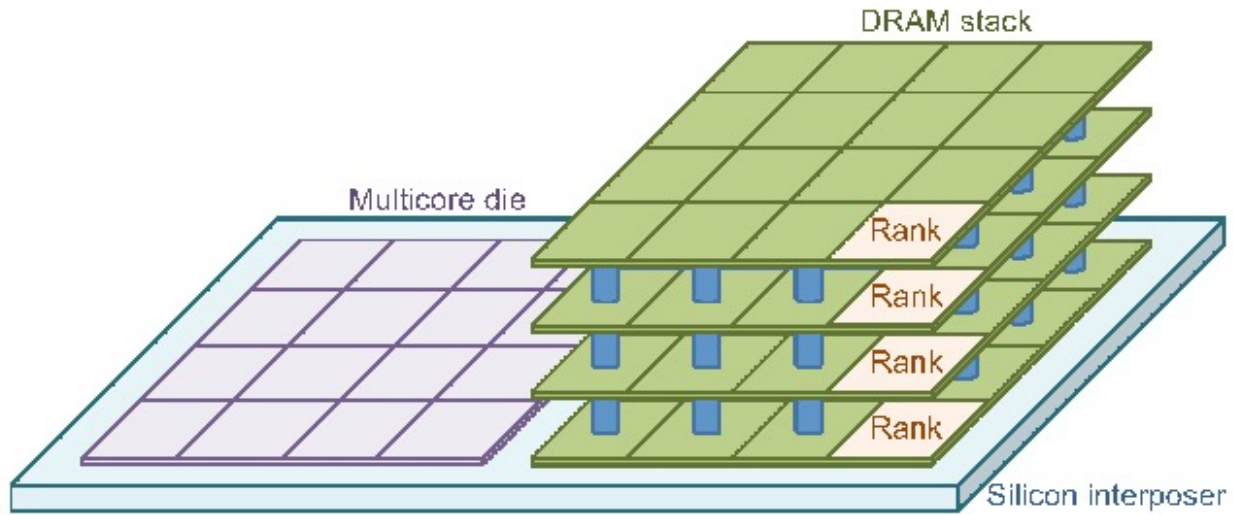
- More silicon = More chances of failure
- Massive chips are not economical
 - Low yields
- In this course, we will assume ideal systems i.e., no faults

Network-on-Chip

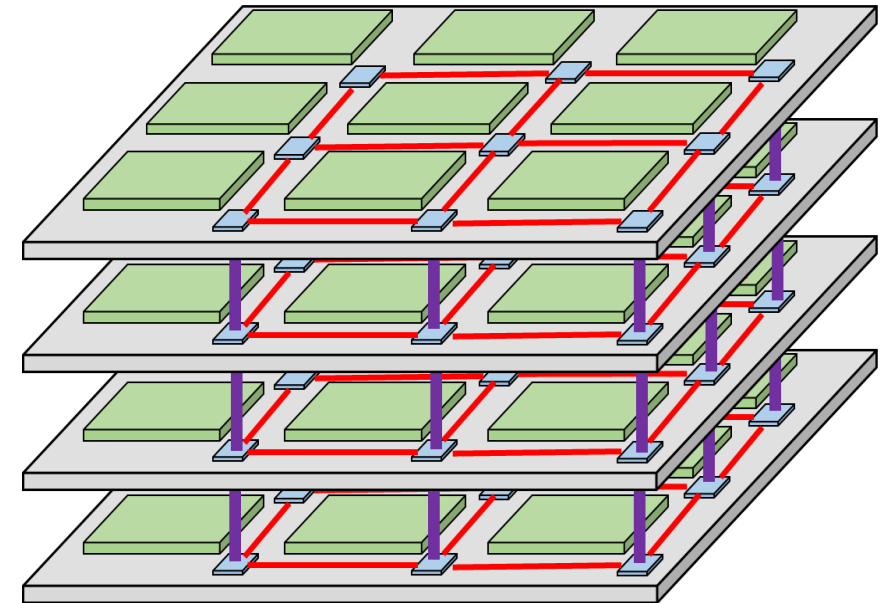


- **New direction in on-chip communication**
- **Low latency, High-throughput, Energy-efficient**
- **Scalable to massive manycore systems**

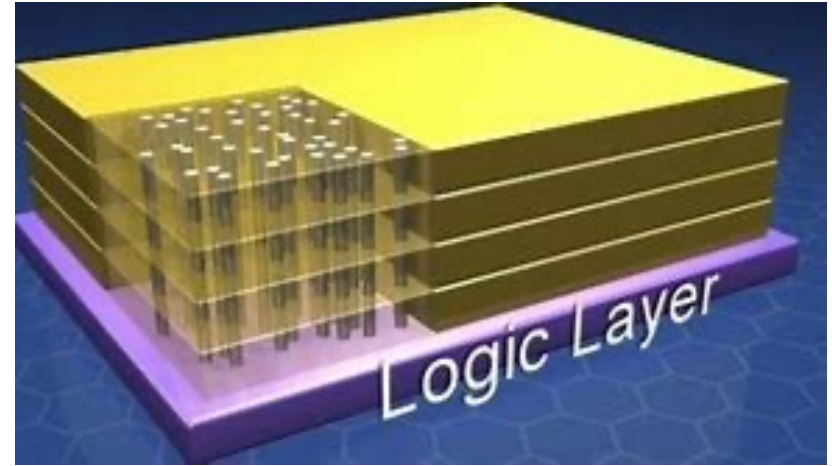
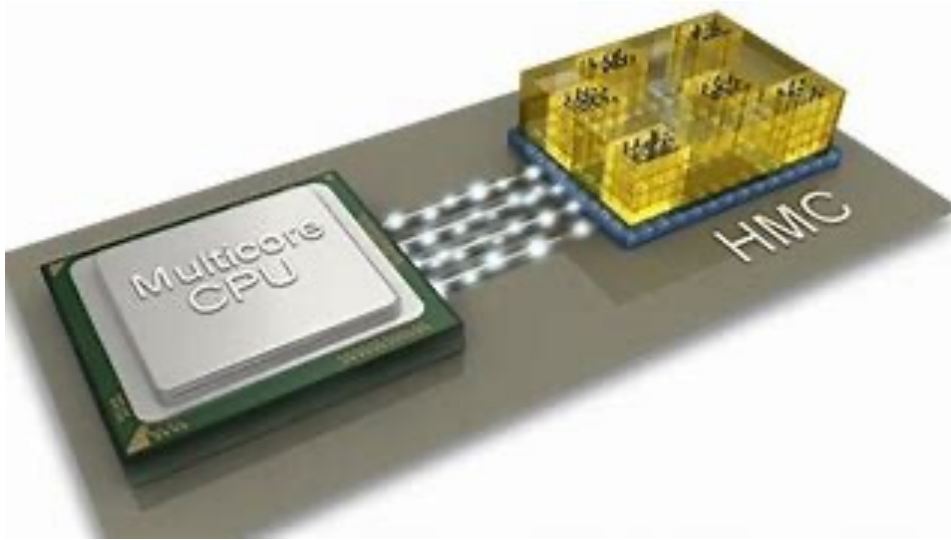
2.5D and 3D Manycores



- More integration density
- High throughput, Energy efficiency
- Accelerates applications like Neural Networks
- Micron's Hybrid Memory Cube (HMC)

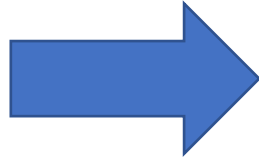


Processing-in/near-memory






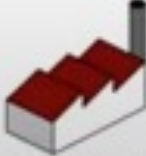




- **Compute near data or inside memory**
- **Extremely high throughput**
- **Accelerates Neural Networks, Graph, Bioinformatics applications**

Hardware security



- Hardware needs to be secure
- Threats:
 - Counterfeiting
 - DDoS attacks
 - Trojans
 - Rowhammer, etc.

	 Trusted user	 Untrusted user
 Trusted foundry		 Camouflaging
 Untrusted foundry	 Split manufacturing	 Logic Encryption