

# **CSC 252: Computer Organization**

## **Spring 2023: Lecture 28**

Instructor: Yuhao Zhu

Department of Computer Science  
University of Rochester

# Announcements

- Lab 5 due this Friday 11:59pm.
- Final exam: May 1, 8:30 AM -- 11:30 AM; online.
- Past exam & Problem set: <https://www.cs.rochester.edu/courses/252/spring2023/handouts.html>
- Exam will be electronic using Gradescope.
- If your final is SIGNIFICANTLY better than mid-term...

# Announcements

- Open book test: any sort of paper-based product, e.g., book, **notes**, magazine, old tests.
- Exams are designed to test your ability to apply what you have learned and not your memory (though a good memory could help).
- **Nothing electronic (including laptop, cell phone, calculator, etc) other than the computer you use to take the exam.**
- **Nothing biological**, including your roommate, husband, wife, your hamster, another professor, etc.
- “**I don’t know**” gets 15% partial credit. Must erase everything else.

# Dark Silicon

*n.* [därk, sǐl'i-kən, -kön']

More transistors on chip (due to Moore's Law), but a growing fraction cannot actually be used due to power limits (due to the end of Dennard Scaling).

# Entering the Era of Specialization

# Entering the Era of Specialization

- GPUs are very efficient for massively parallel program

# Entering the Era of Specialization

- GPUs are very efficient for massively parallel program
- But are still fairly general, so there are still many inefficiencies
  - Still need to fetch and decode instructions
  - Still have (very large) caches, so data delivery isn't efficient

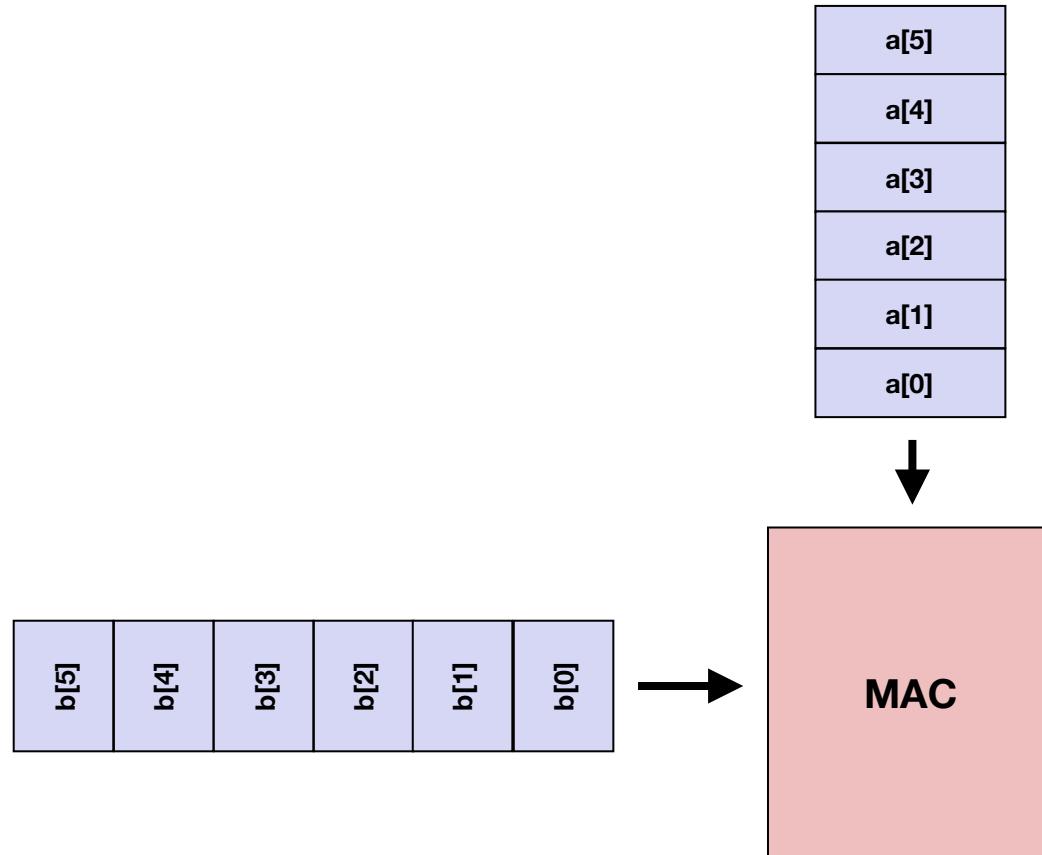
# Entering the Era of Specialization

- GPUs are very efficient for massively parallel program
- But are still fairly general, so there are still many inefficiencies
  - Still need to fetch and decode instructions
  - Still have (very large) caches, so data delivery isn't efficient
- Idea: instead of building general-purpose processors that can do everything, but inefficiently, let's build specialized processors that can only do limited things, but extremely efficiently.

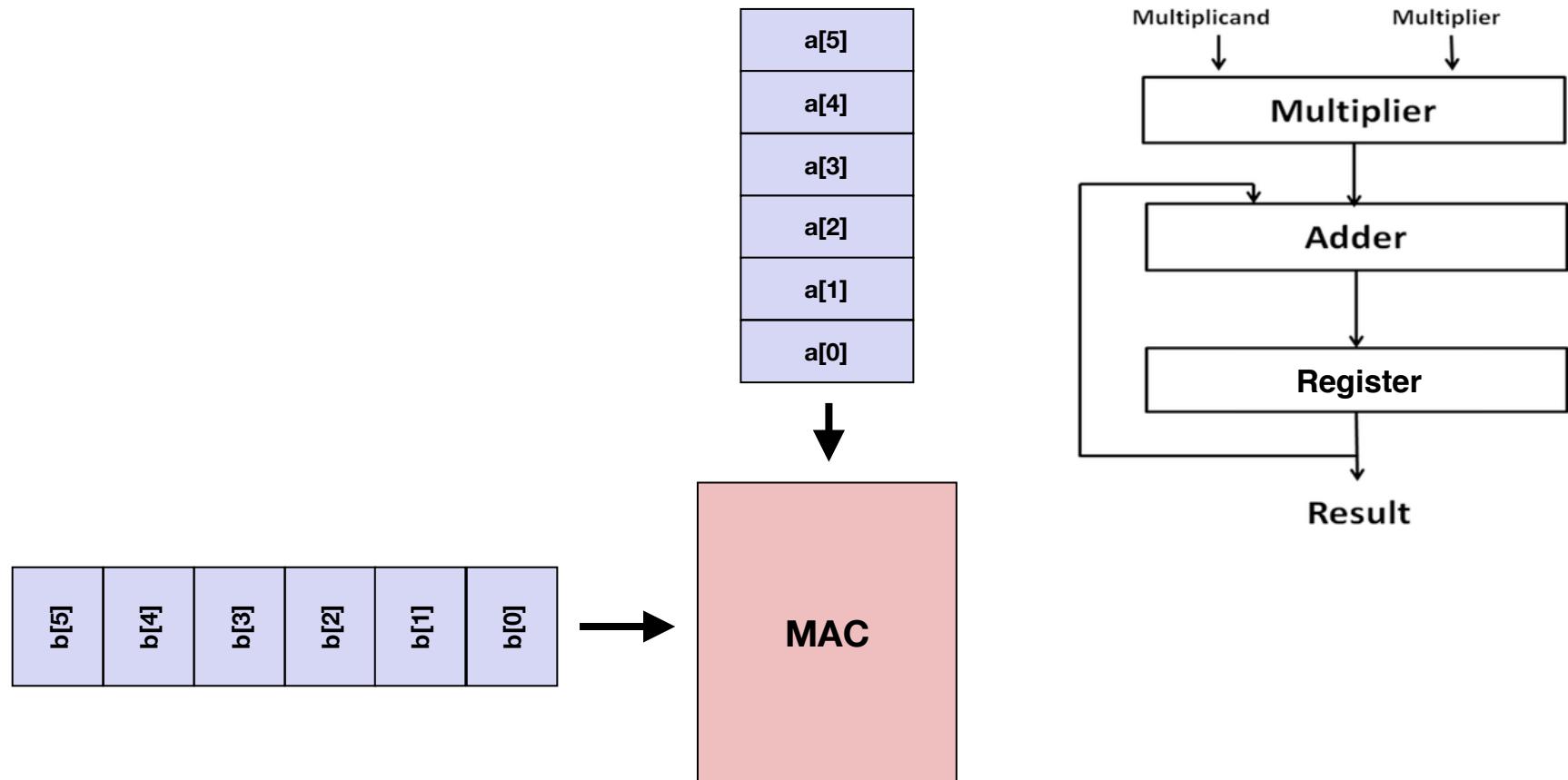
# Entering the Era of Specialization

- GPUs are very efficient for massively parallel program
- But are still fairly general, so there are still many inefficiencies
  - Still need to fetch and decode instructions
  - Still have (very large) caches, so data delivery isn't efficient
- Idea: instead of building general-purpose processors that can do everything, but inefficiently, let's build specialized processors that can only do limited things, but extremely efficiently.
- A.k.a., domain-specific accelerators

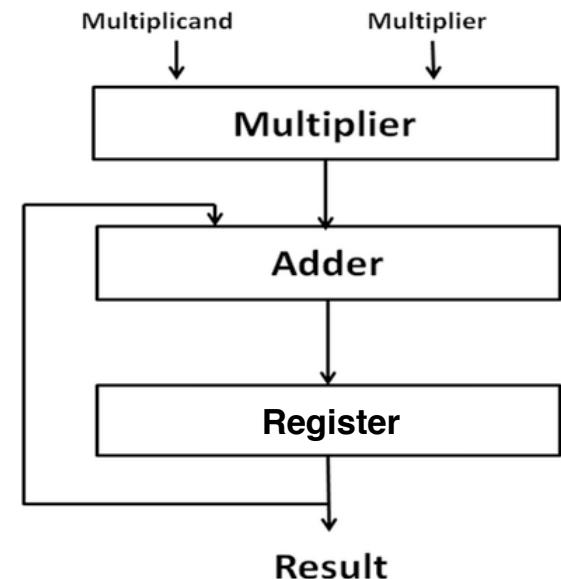
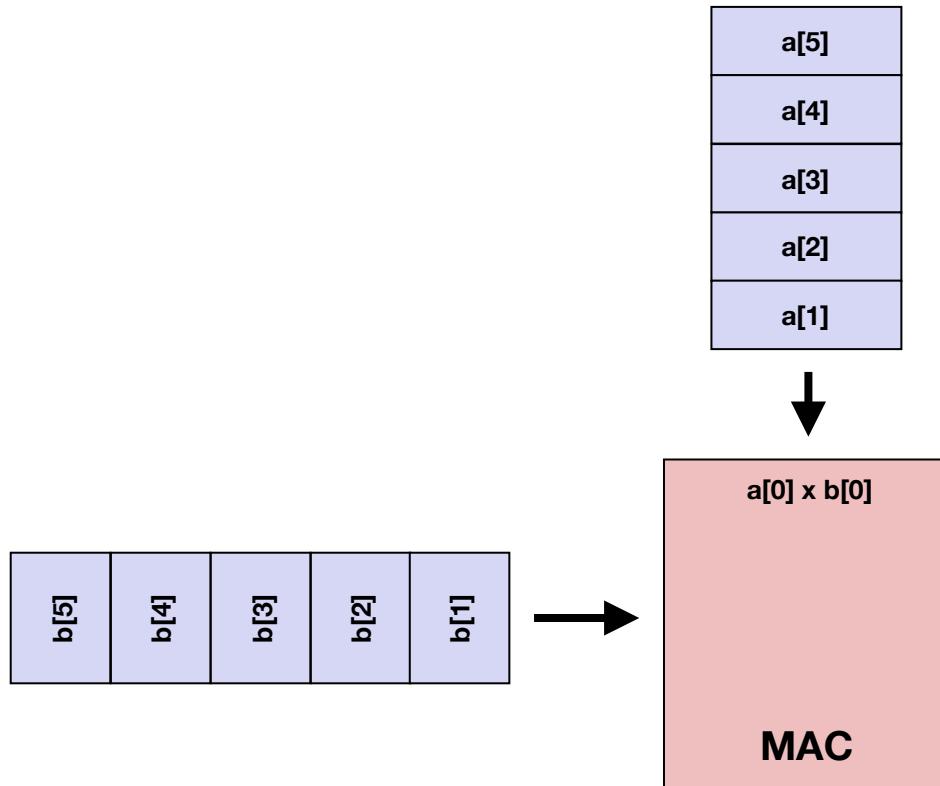
# Example: Vector Dot Product



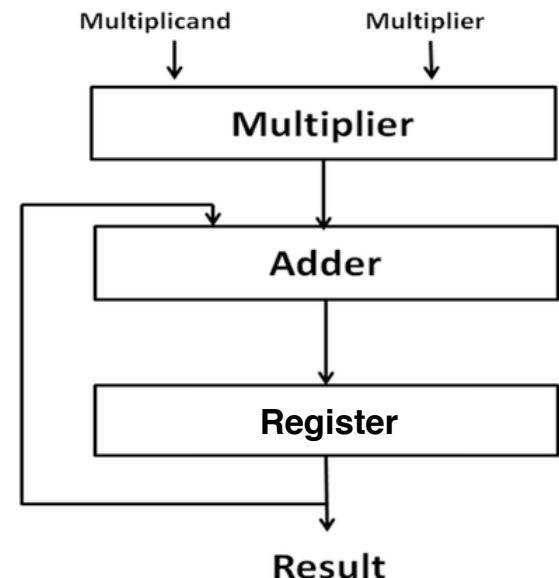
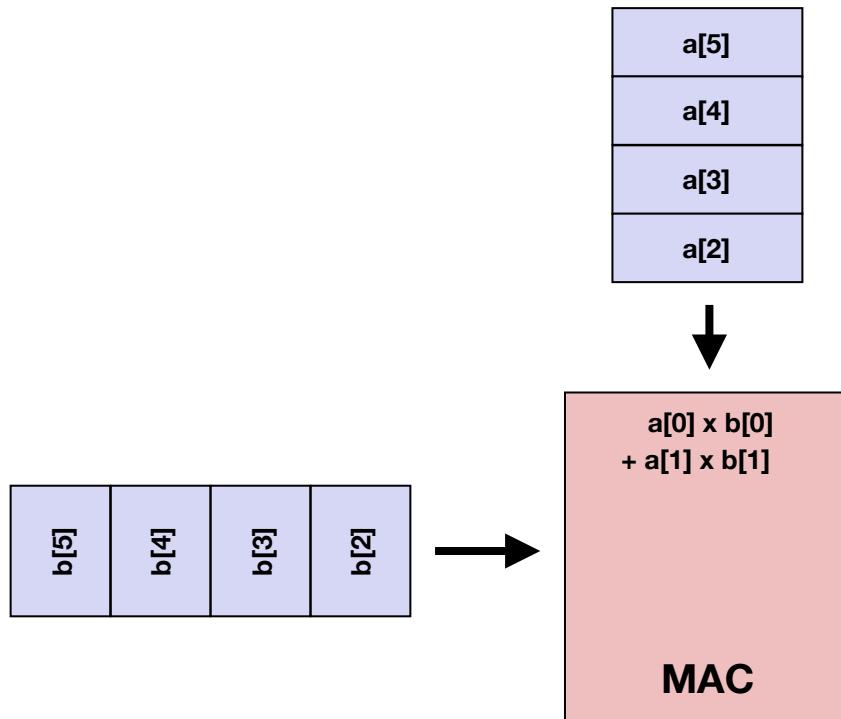
# Example: Vector Dot Product



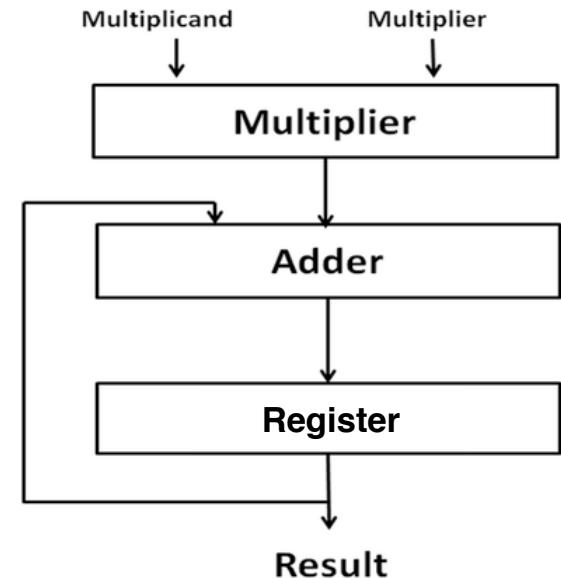
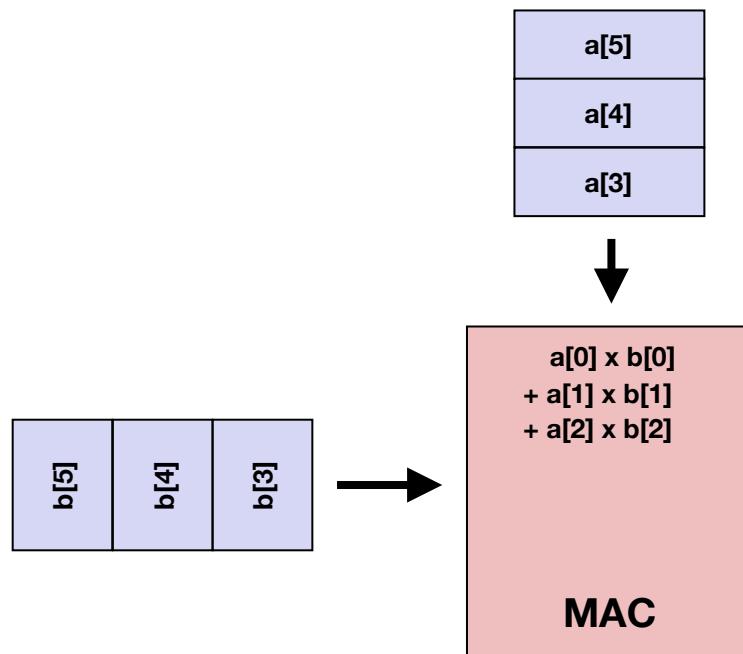
# Example: Vector Dot Product



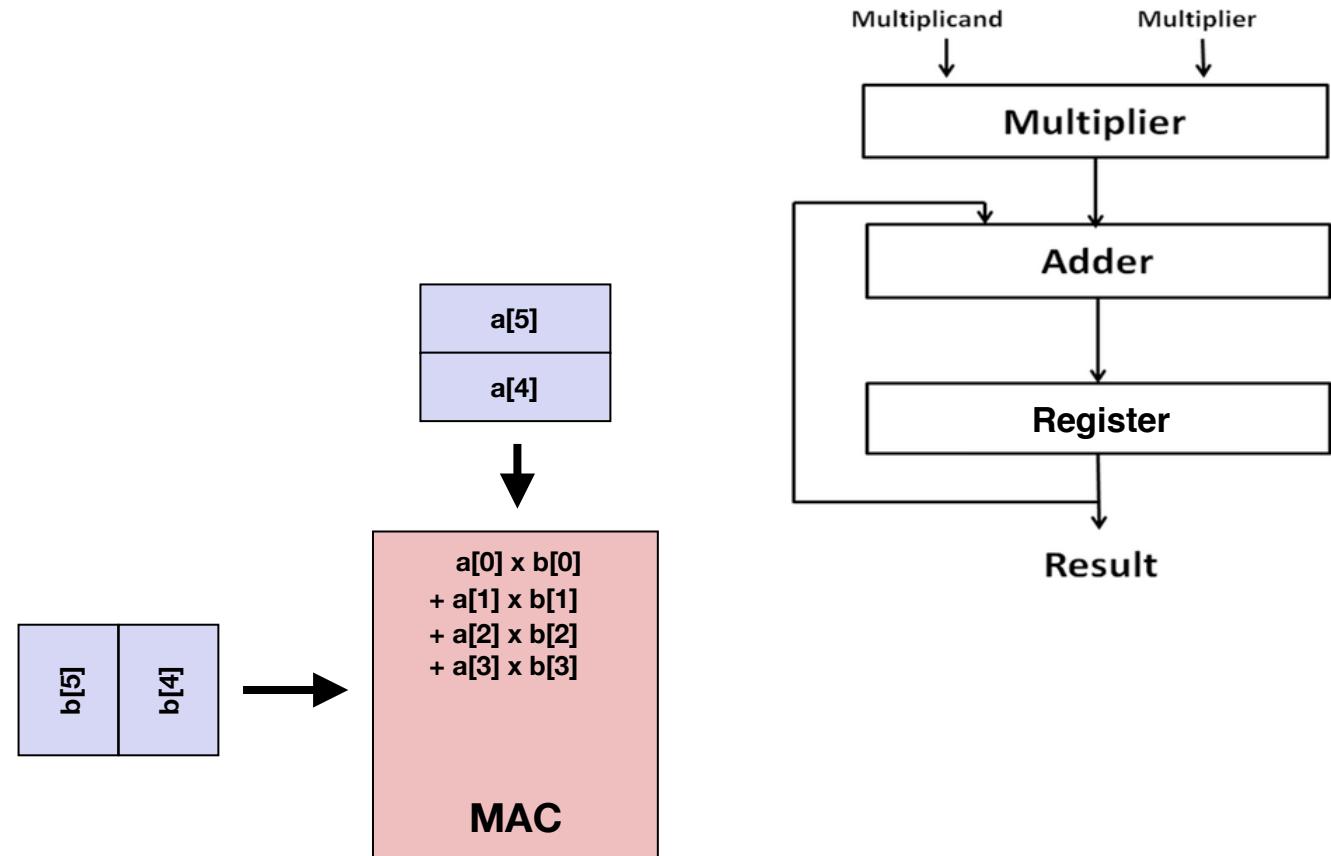
# Example: Vector Dot Product



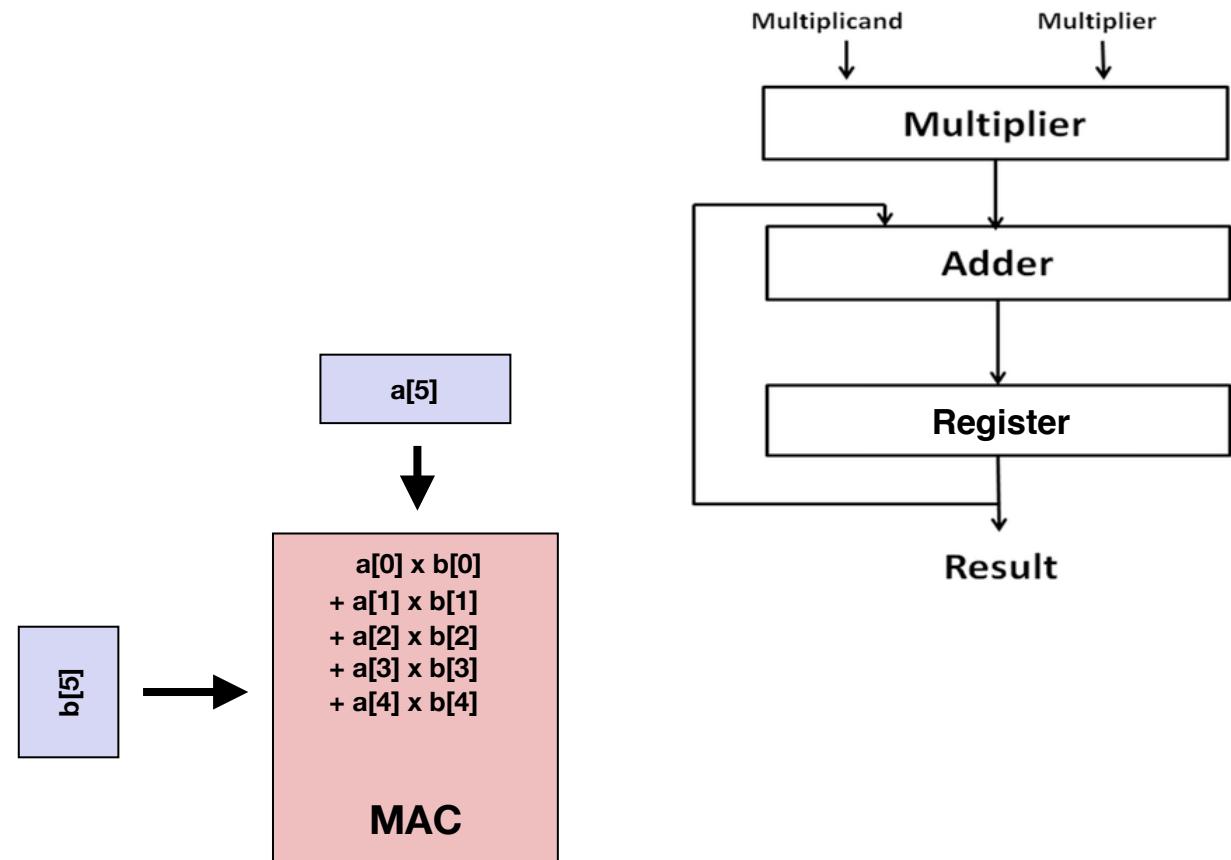
# Example: Vector Dot Product



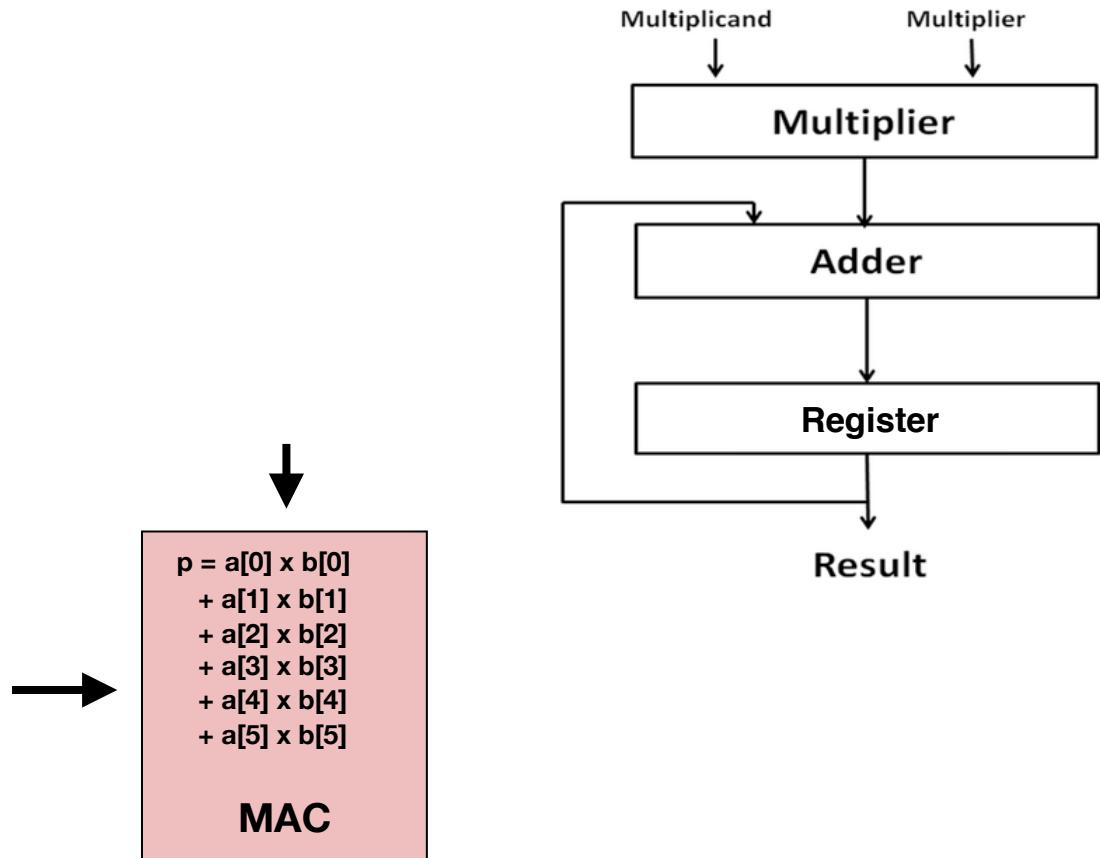
# Example: Vector Dot Product



# Example: Vector Dot Product

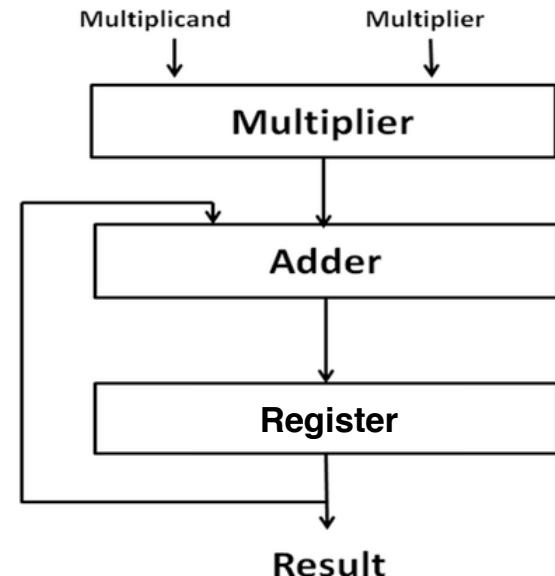
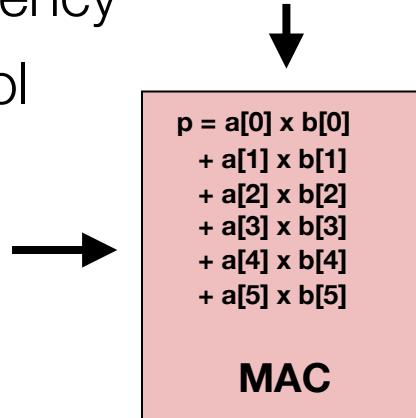


# Example: Vector Dot Product

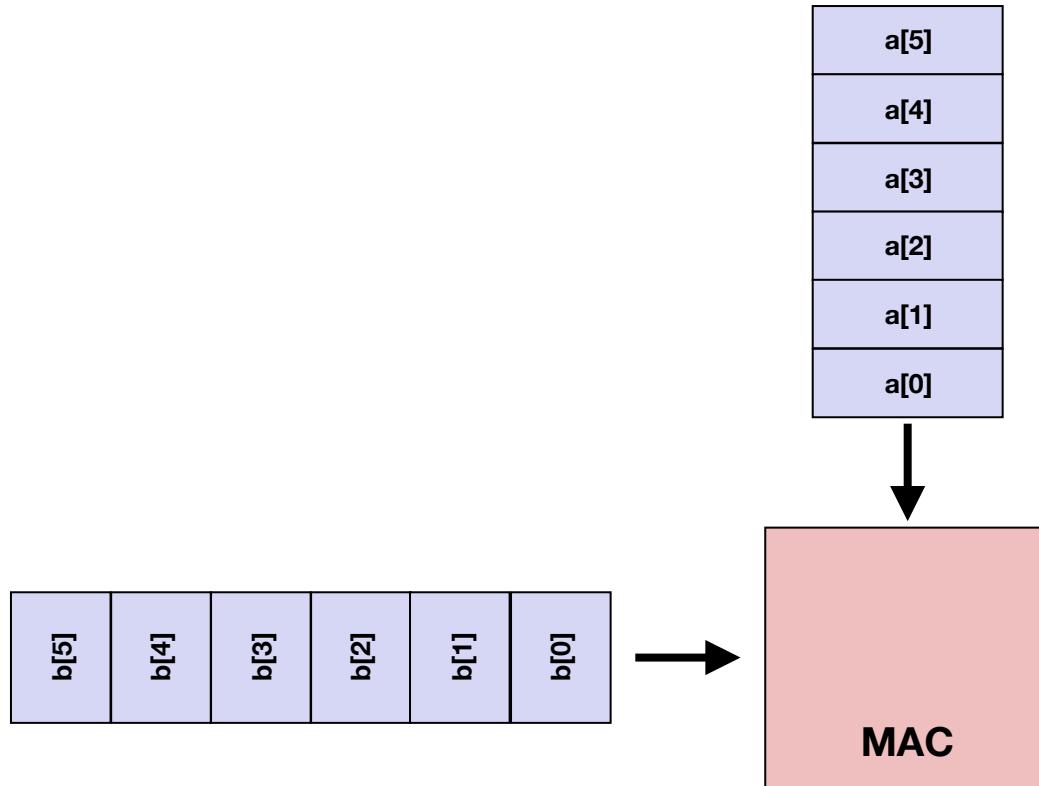


# Example: Vector Dot Product

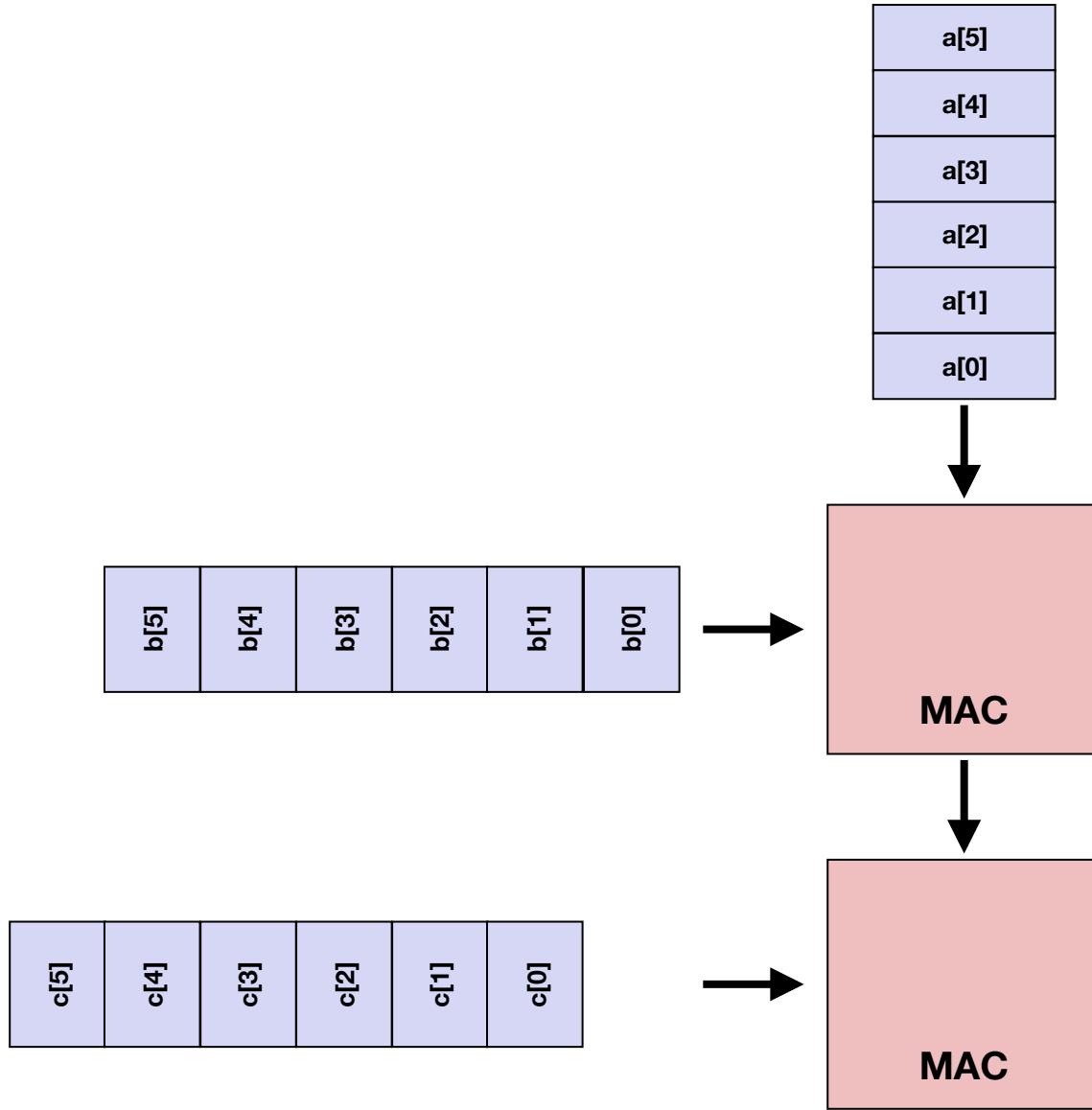
- Does nothing but vector dot product
  - No instruction fetch and decode  
(there is no instruction)
  - The register is close to the ALU and gets reused over and over: good data delivery efficiency
  - Very simple control



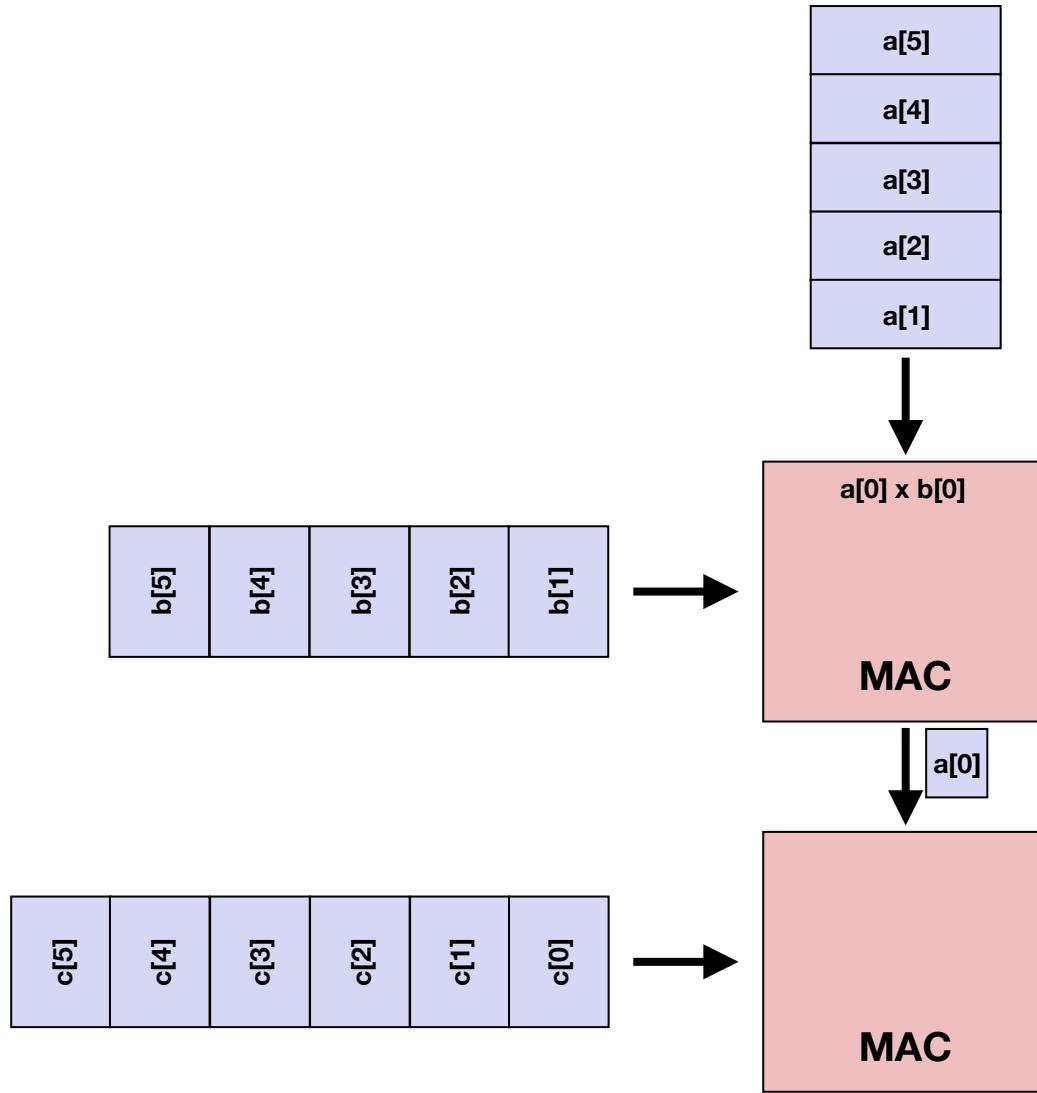
# Matrix Vector Multiplication



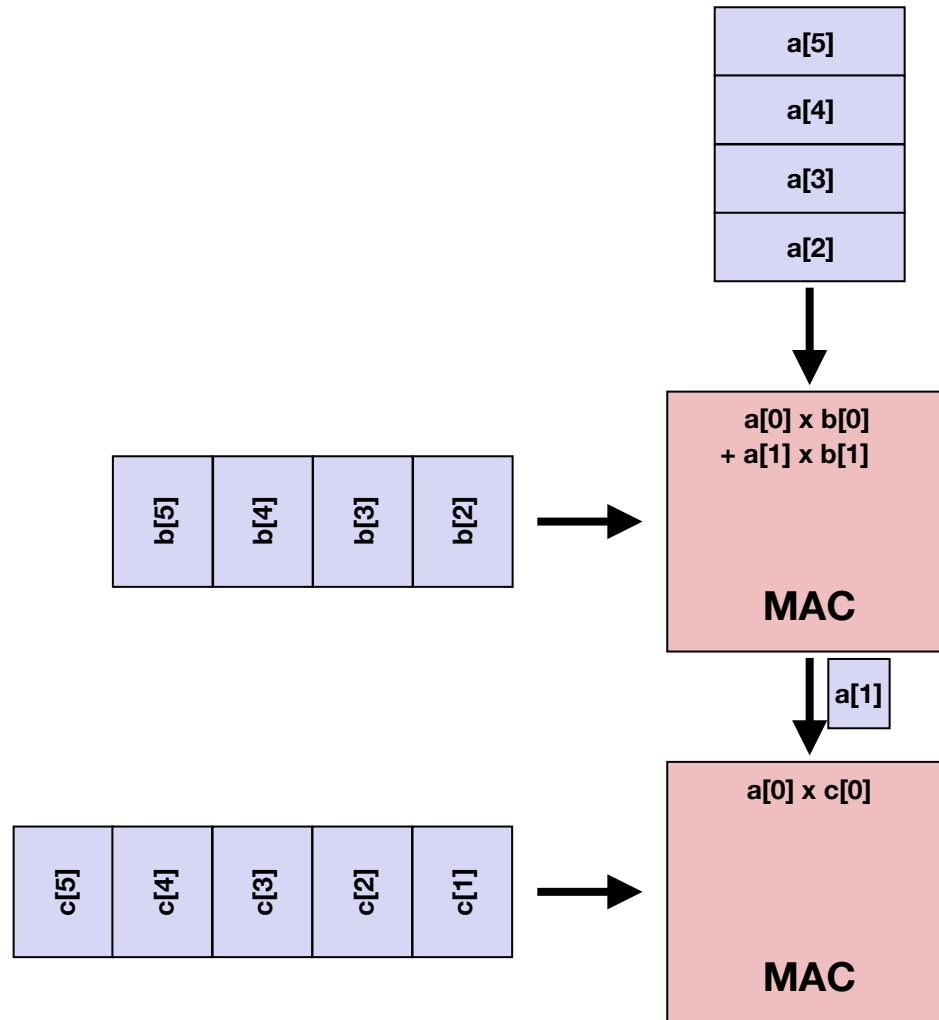
# Matrix Vector Multiplication



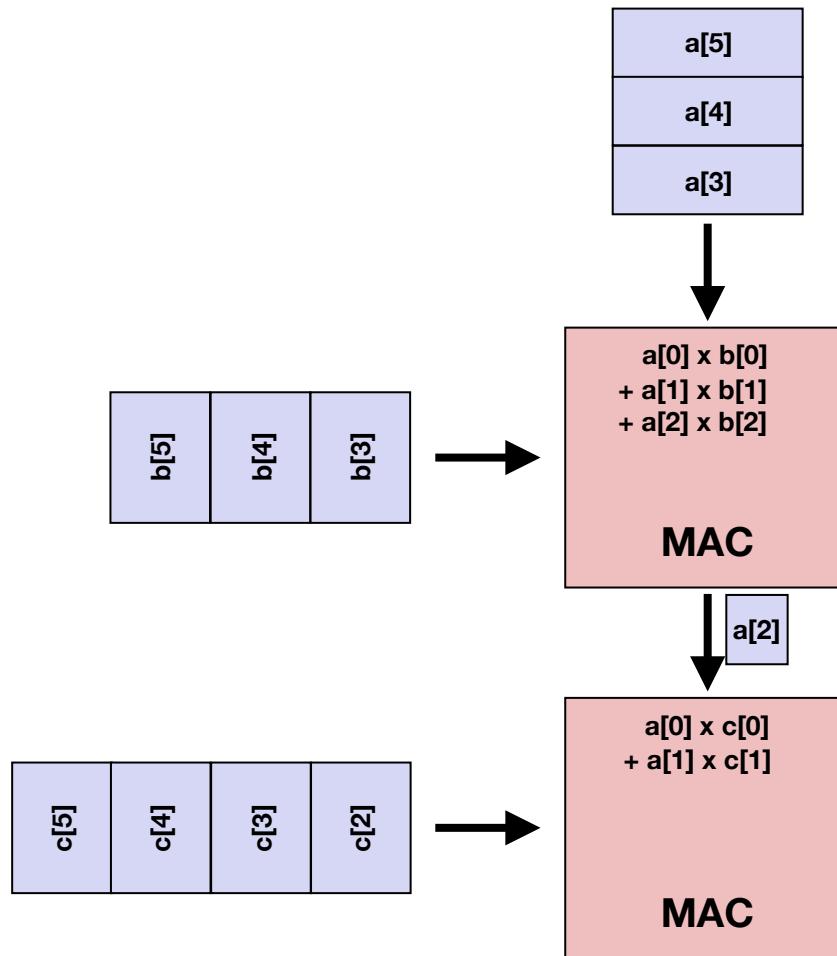
# Matrix Vector Multiplication



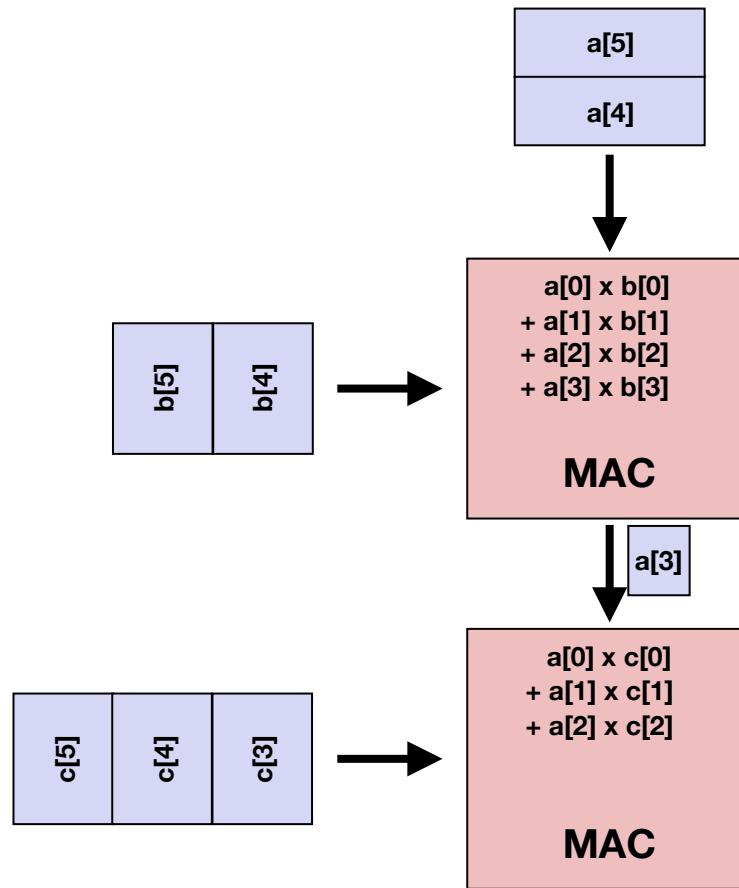
# Matrix Vector Multiplication



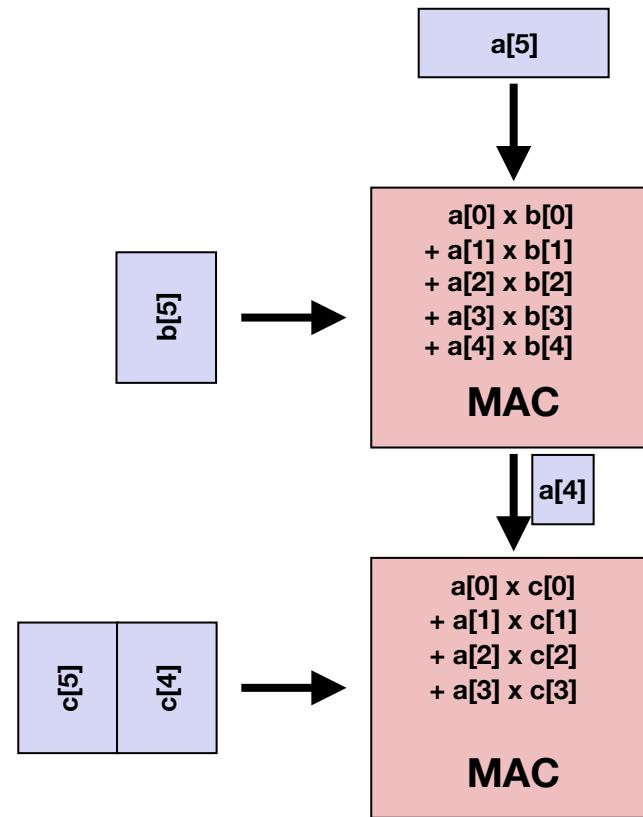
# Matrix Vector Multiplication



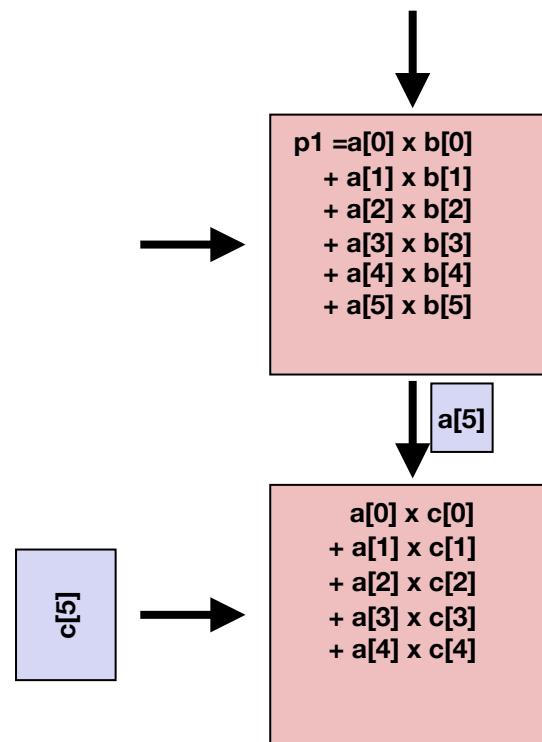
# Matrix Vector Multiplication



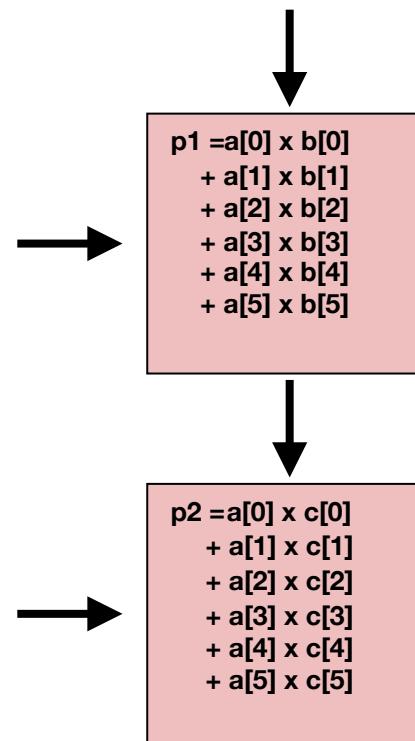
# Matrix Vector Multiplication



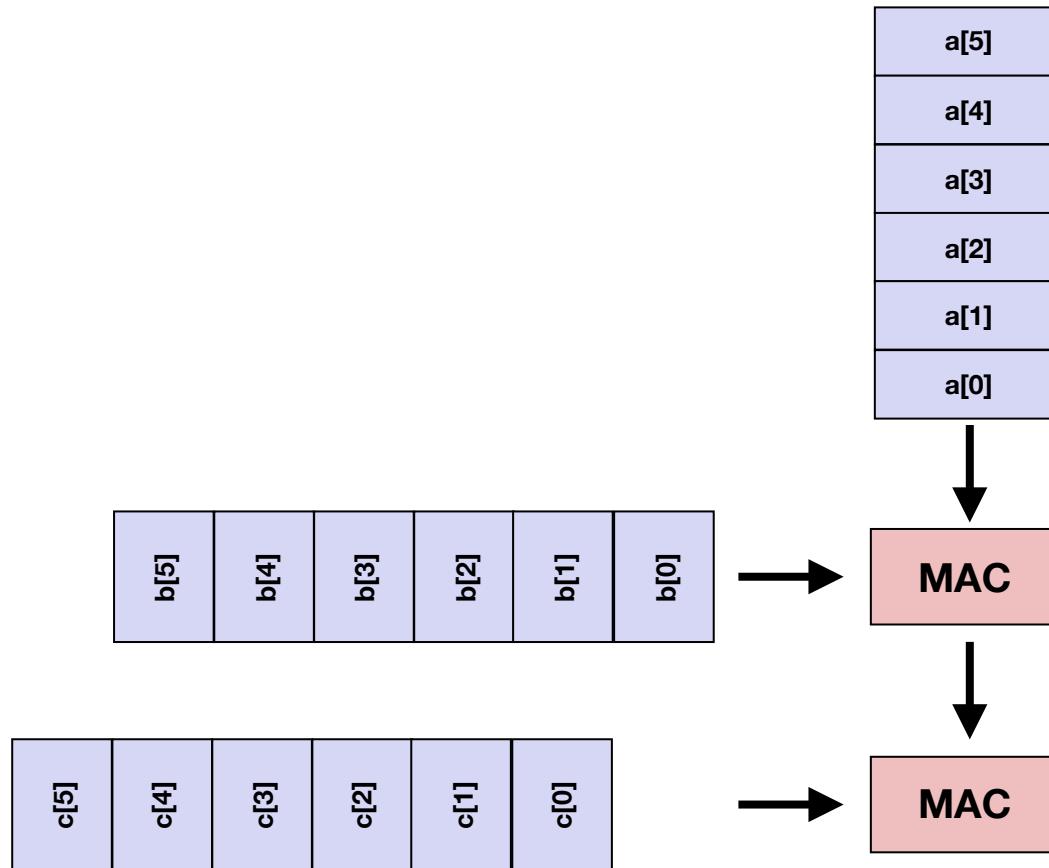
# Matrix Vector Multiplication



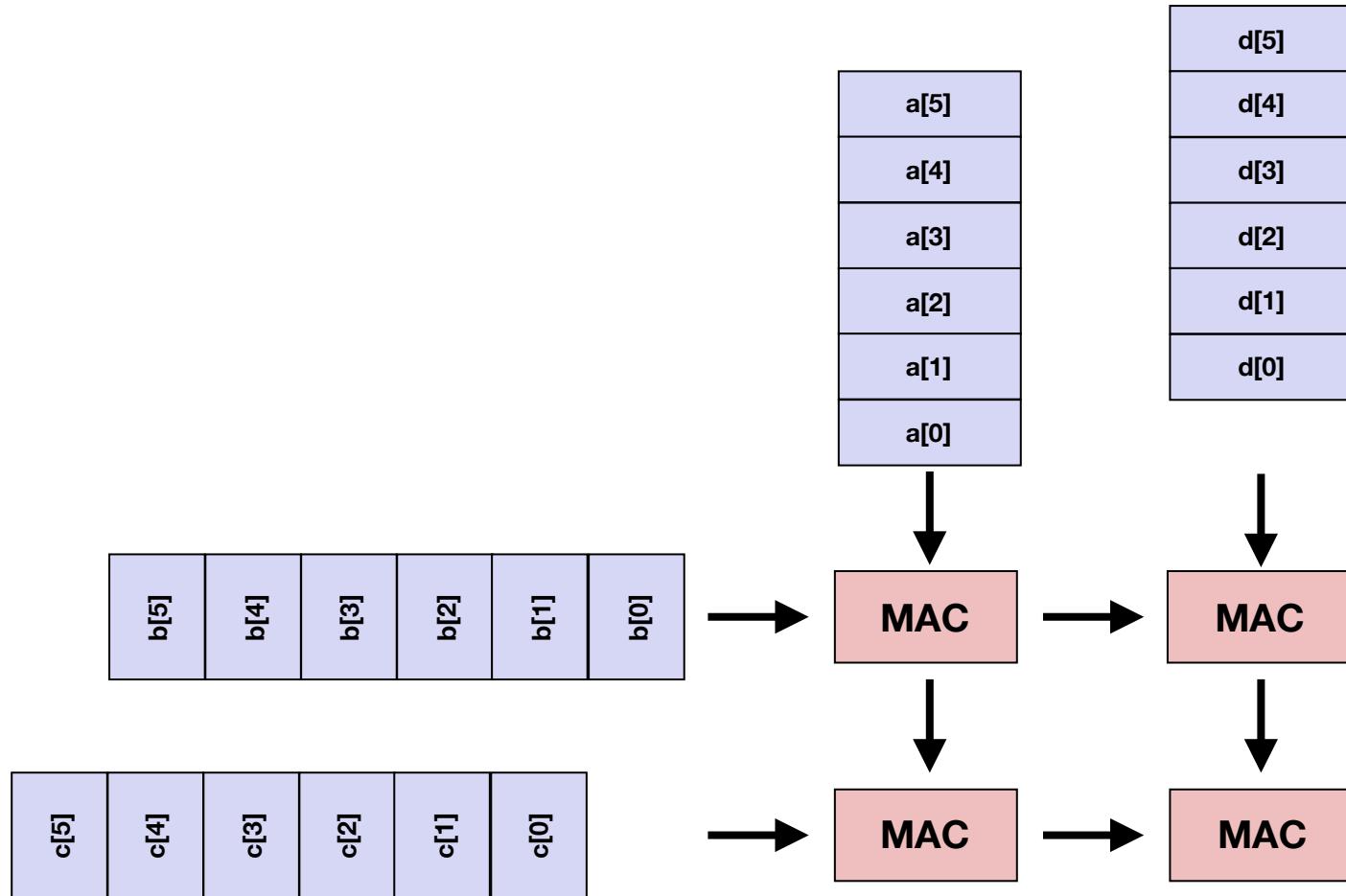
# Matrix Vector Multiplication



# Matrix Matrix Multiplication

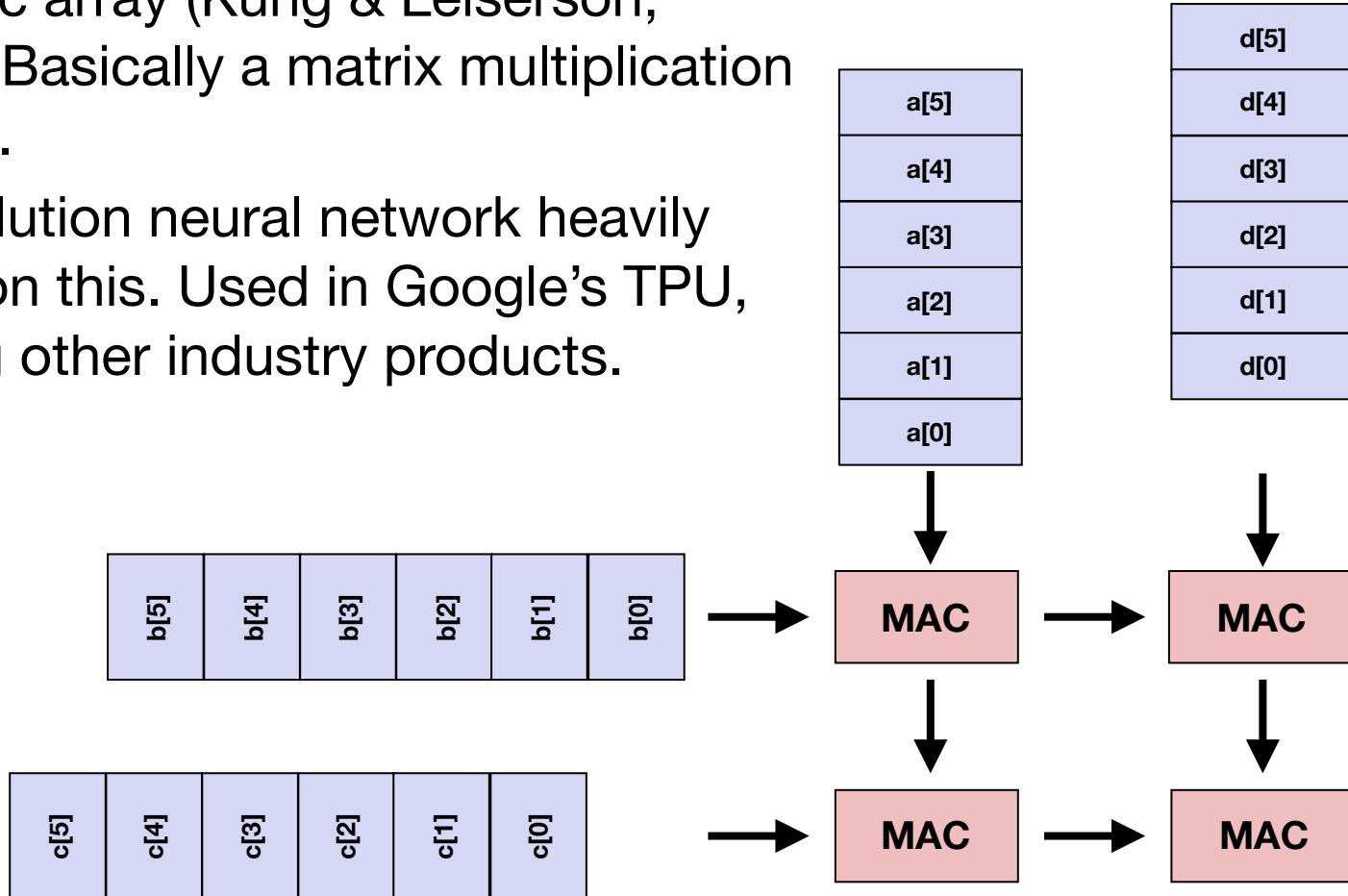


# Matrix Matrix Multiplication



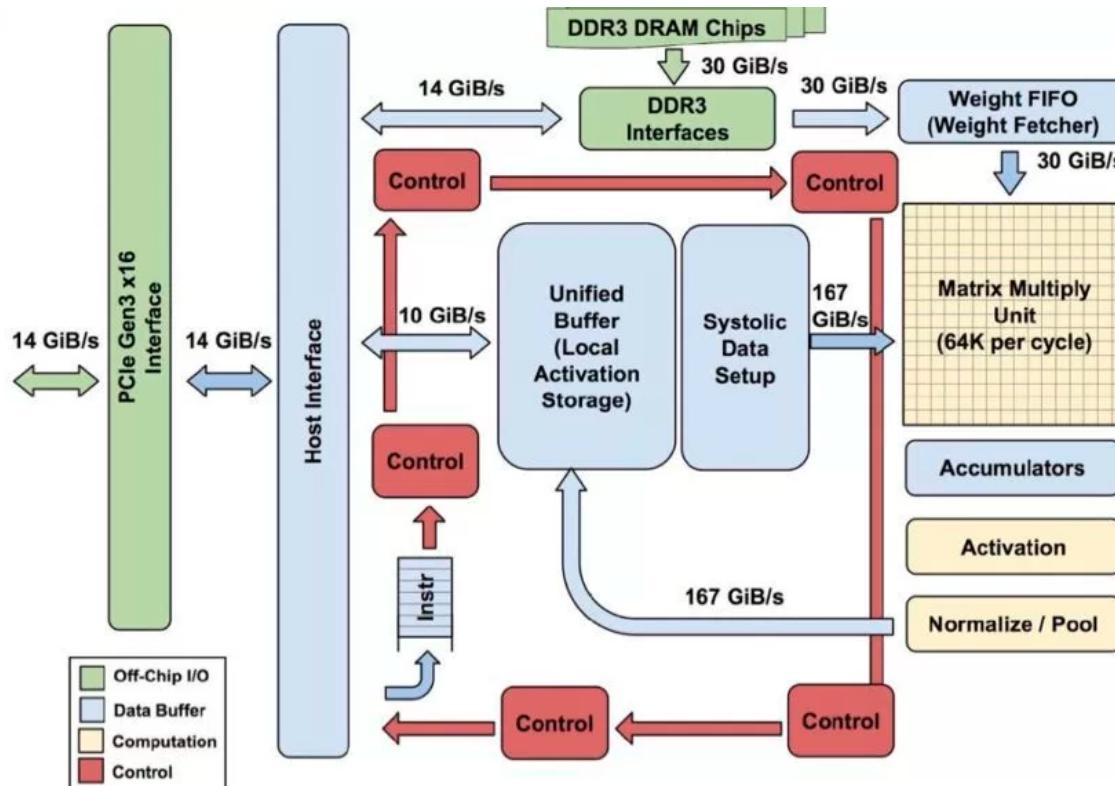
# Matrix Matrix Multiplication

- Systolic array (Kung & Leiserson, 1978). Basically a matrix multiplication engine.
- Convolution neural network heavily relies on this. Used in Google's TPU, among other industry products.

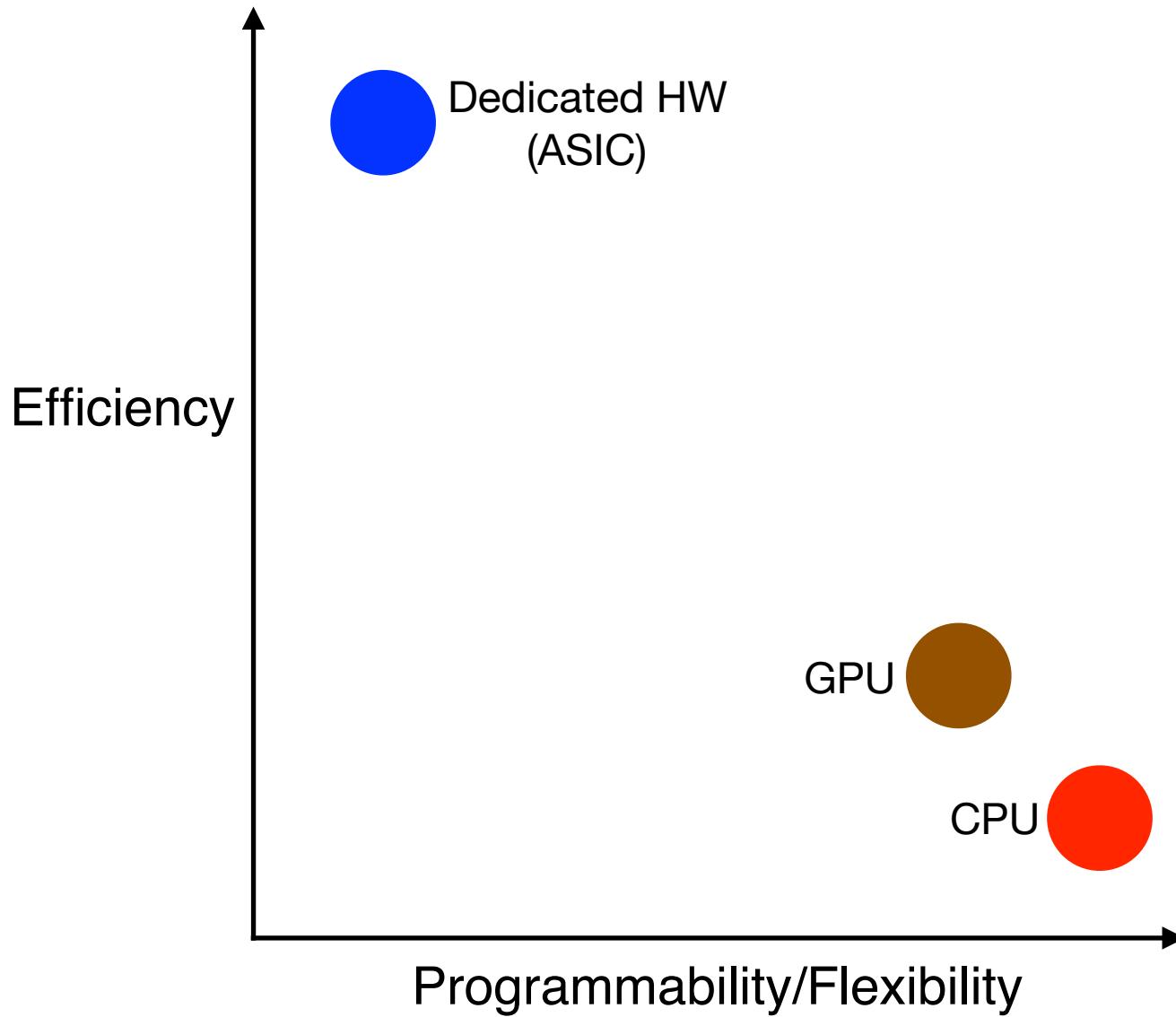


# Google Tensor Processing Unit

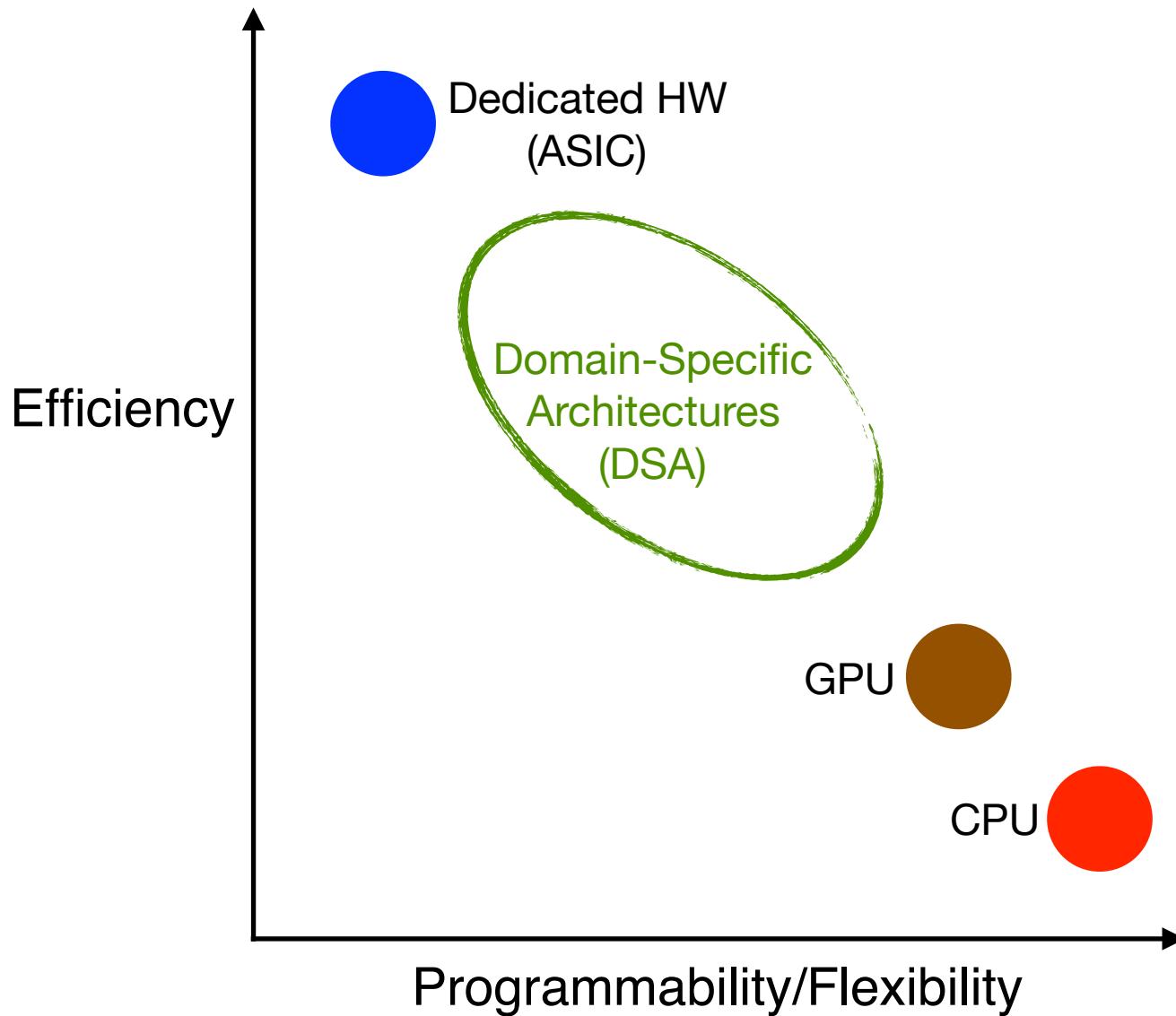
- Convolution in deep learning can be transformed to matrix multiplication.
- TPU: specialized processor (i.e., systolic array architecture) for tensor processing (matrix multiply)
  - 30x~80x more power-efficient than GPU



# The Main Trade-off: Programmability vs. Efficiency



# The Main Trade-off: Programmability vs. Efficiency



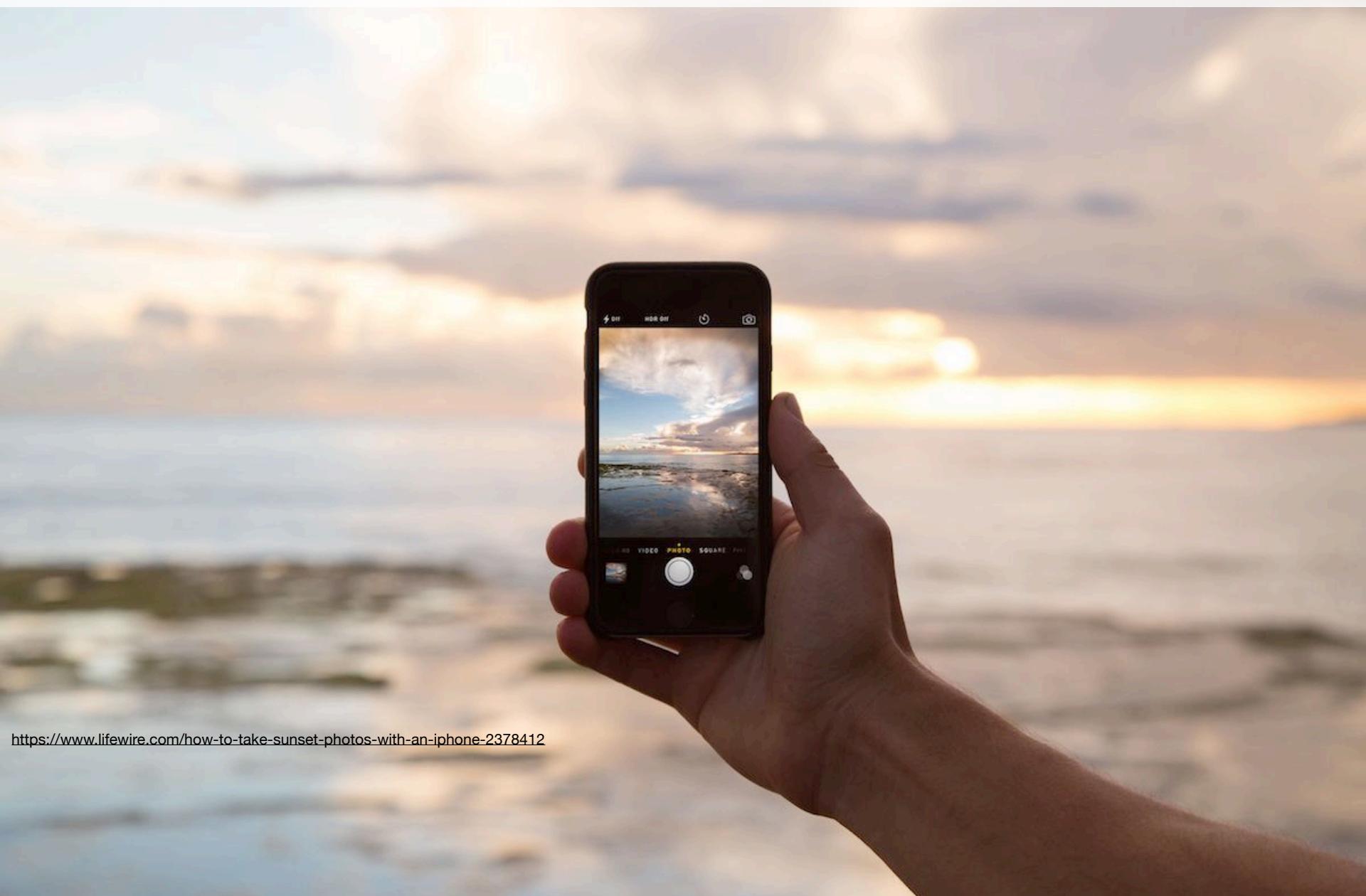
# A Whirlwind of Application Domains (a.k.a., What CSC 292/572 Will Cover)

# Conventional Digital Camera



<https://www.shutterbug.com/content/dslr-strikes-back-why-mirrored-cameras-still-matter-2020>

# Today's Digital Camera



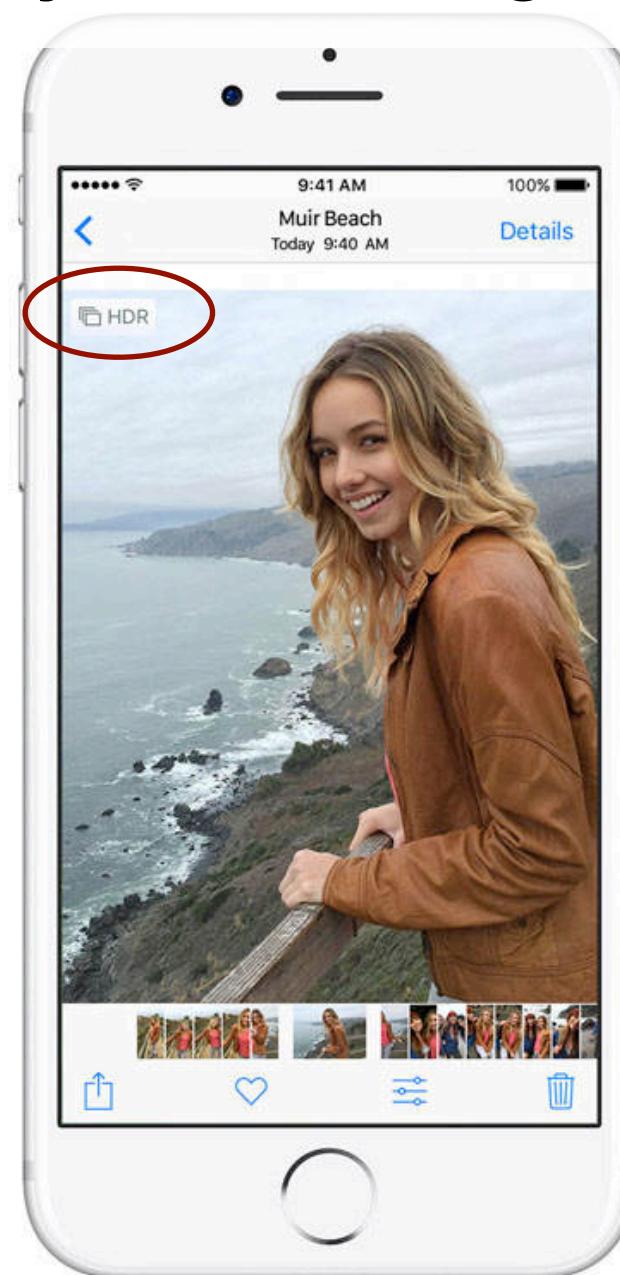
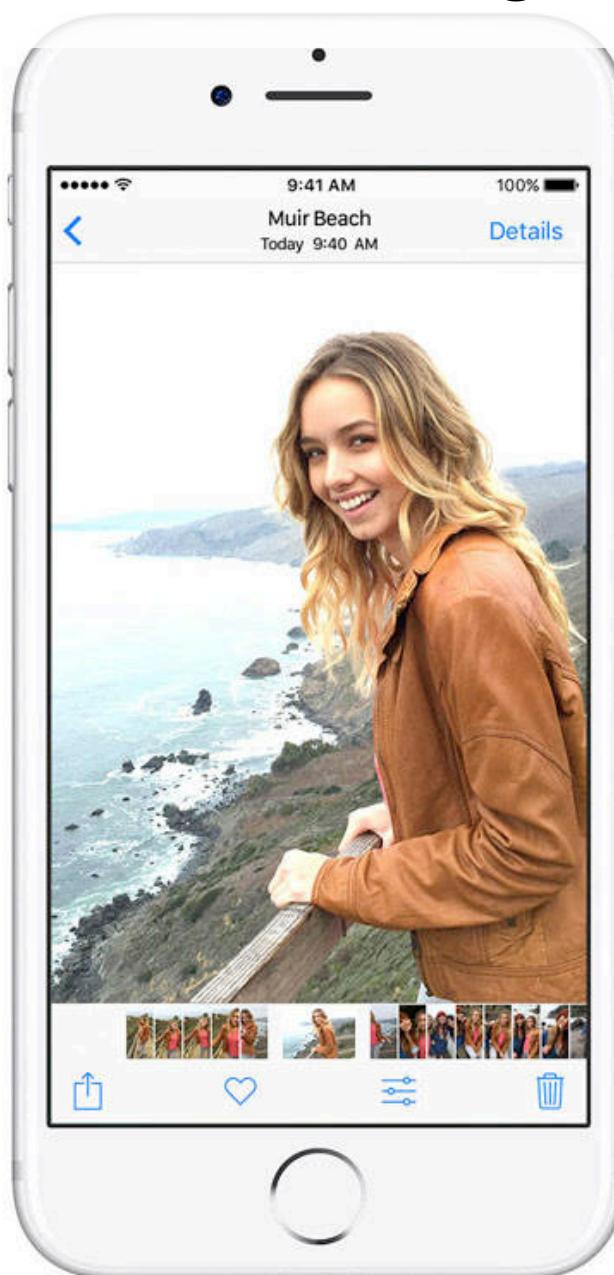
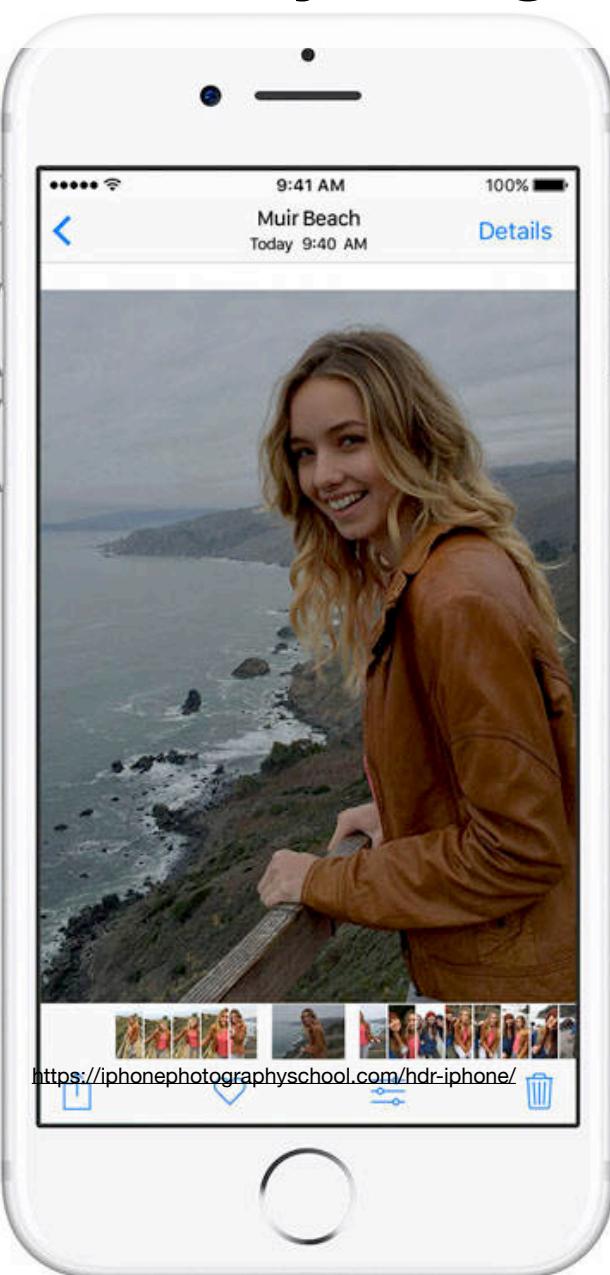
<https://www.lifewire.com/how-to-take-sunset-photos-with-an-iphone-2378412>

# Today's Digital Camera: Portrait Mode



<https://ai.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html>

# Today's Digital Camera: High Dynamic Range



<https://iphonephotoschool.com/hdr-iphone/>



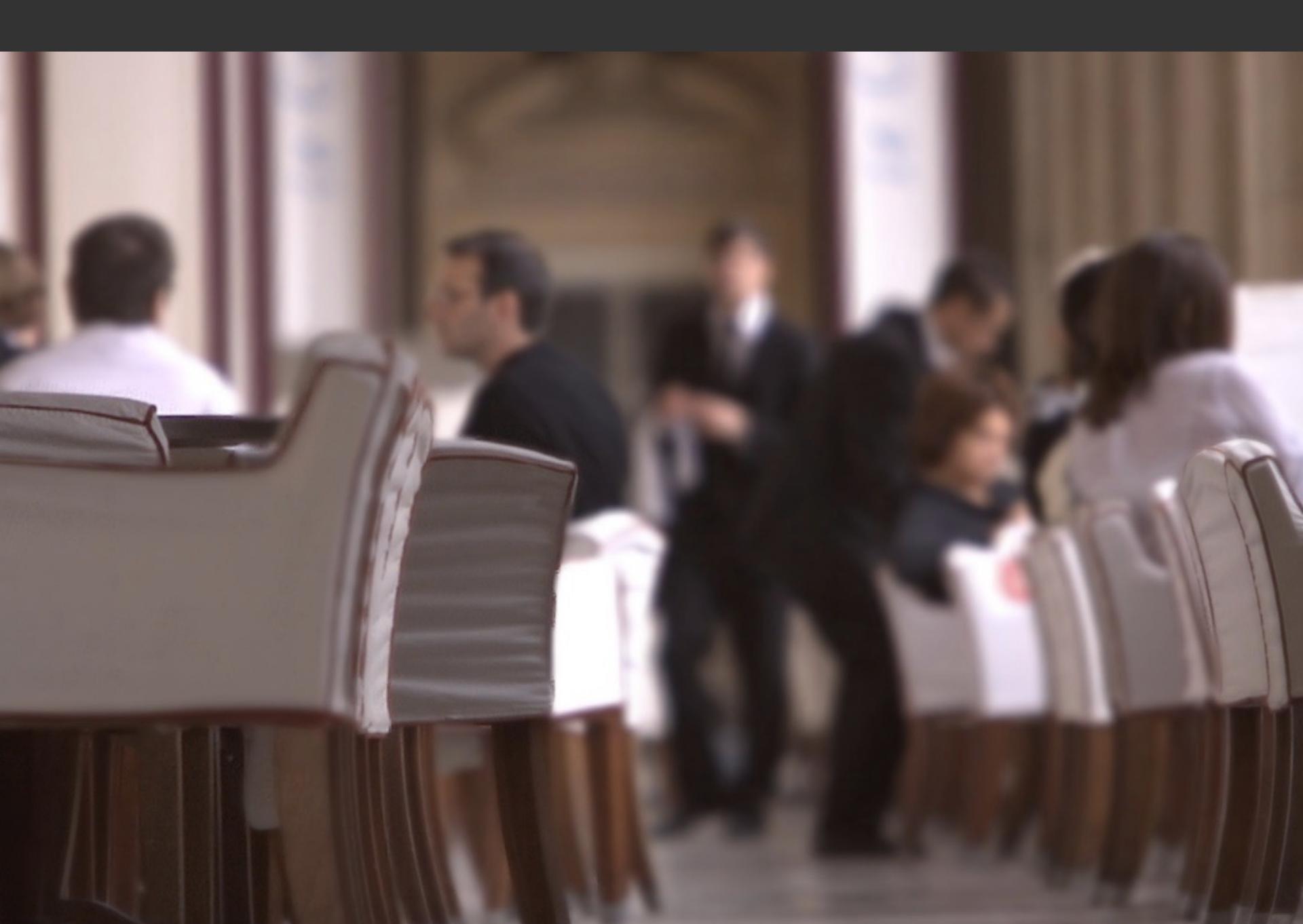




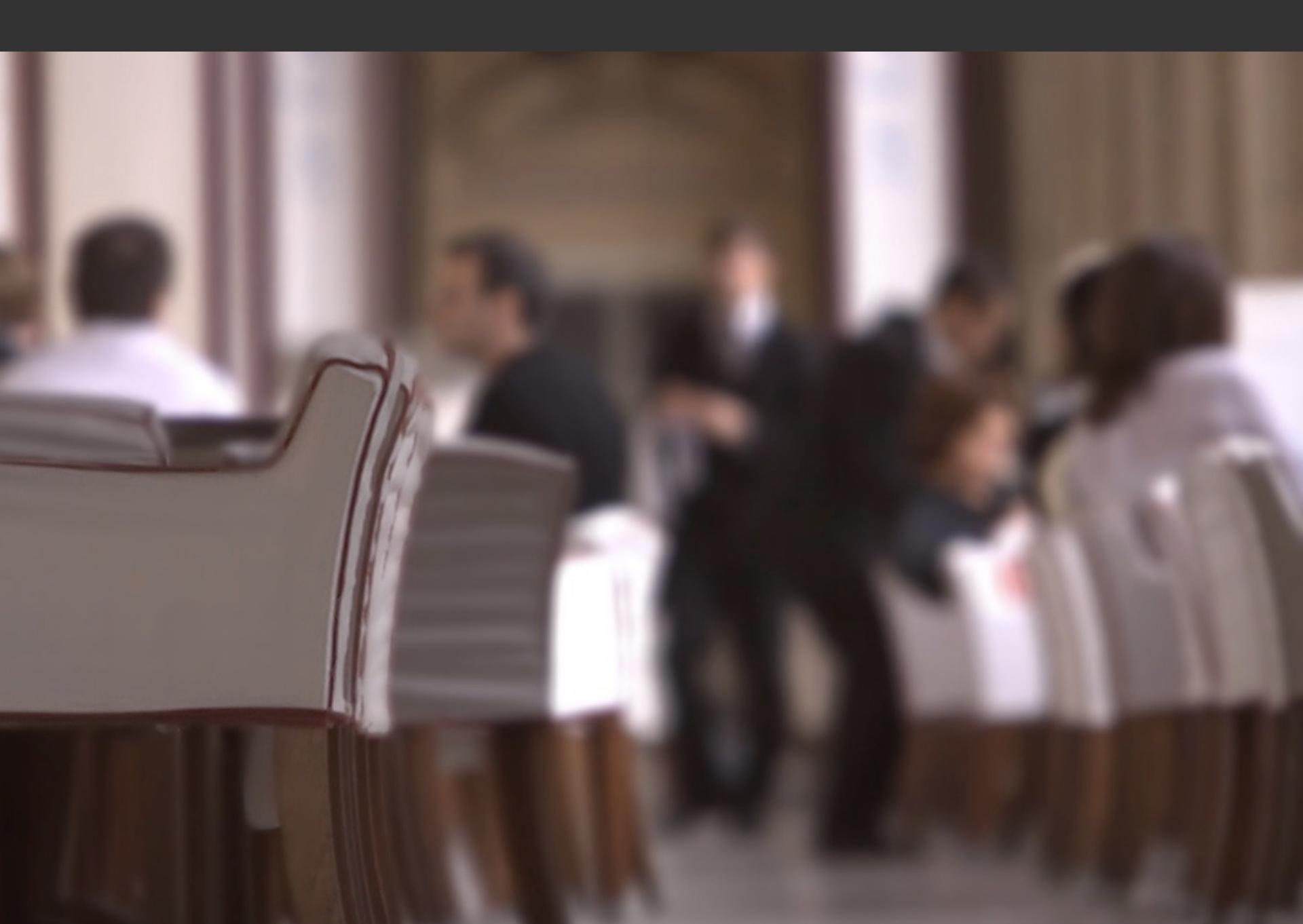
[Slide courtesy Ren Ng]



[Slide courtesy Ren Ng]



[Slide courtesy Ren Ng]



[Slide courtesy Ren Ng]

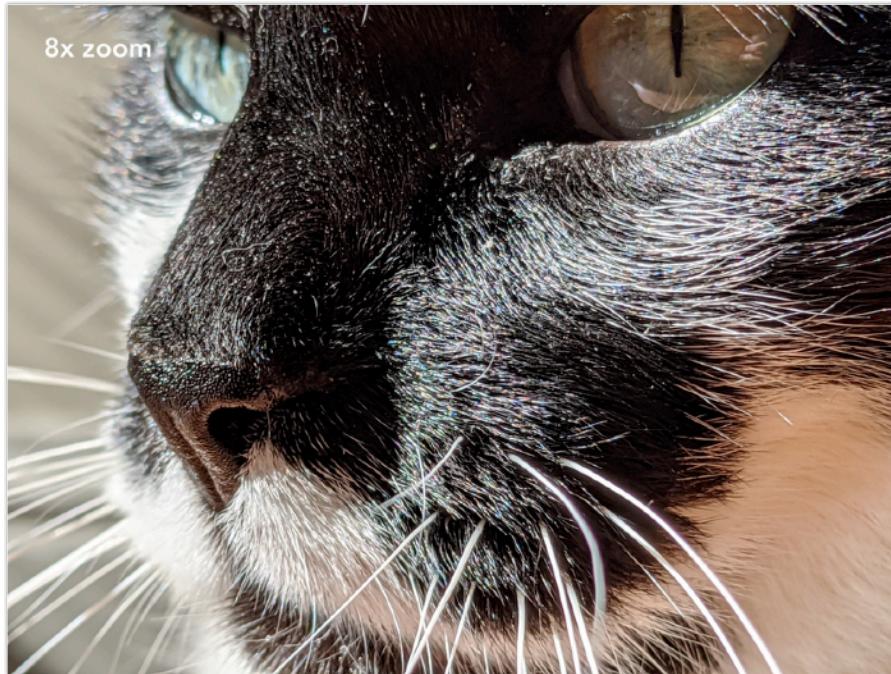


[Slide courtesy Ren Ng]

# Cameras in Modern Smartphones



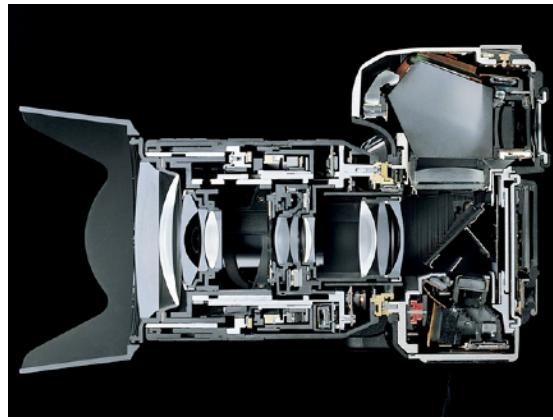
# Telephoto (Long-Focus) Lens



**Telephoto lens ( $f = 48 \text{ mm}$ ) + computational photography algorithm for super resolution**

# Traditional Telephoto (Long-Focus) Lens







**Conventional cameras use complex and expensive lenses and sensors to minimize imperfections.**



Conventional cameras use complex and expensive lenses and sensors to minimize imperfections.



Conventional cameras use complex and expensive lenses and sensors to minimize imperfections.



Today's cameras use better processing algorithms to make up for the weak optics and sensors —in real time.



# Photorealistic Rendering



<https://notrianglestudio.com/all-categories-blog/why-photorealistic-renderings-should-be-part-of-your-real-estate-marketing-strategy>

# Photorealistic Rendering



[https://www.youtube.com/watch?v=uY4cE\\_nq2lY](https://www.youtube.com/watch?v=uY4cE_nq2lY)

# Real-Time Photorealistic Rendering (Gaming)



<https://www.digitaltrends.com/gaming/battlefield-v-dxr-ray-tracing-tested/>

# Physics Simulation



<https://www.flickr.com/photos/65945817@N07/6333871893>

# Another Domain: Video Compression



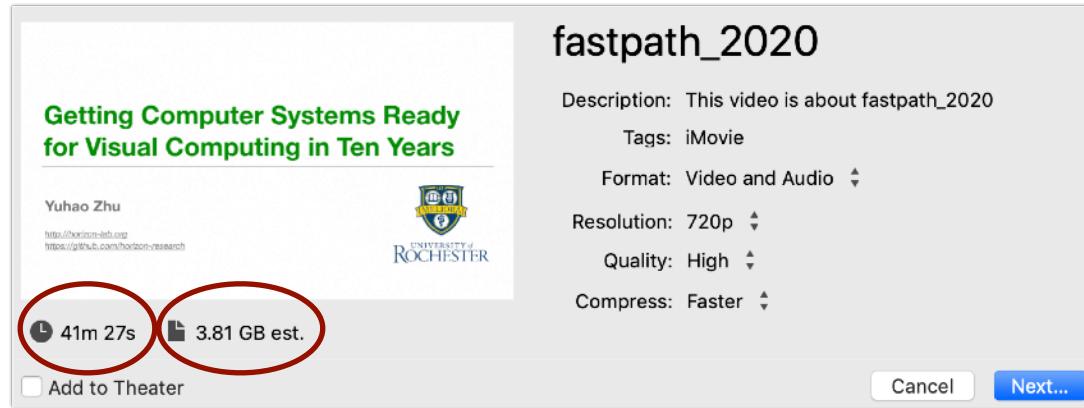
# Another Domain: Video Compression

30-second video @ 1080p resolution (1920 x 1080 pixels per frame) @ 30 frames per second (FPS)  
3 colors per pixel + 1 byte per color → 6.2 MB/frame → 6.2 MB x 30 s x 30 FPS = 5.2 GB total size  
Actual H.264 video file size: 65.4 MB (**80-to-1 compression ratio**).

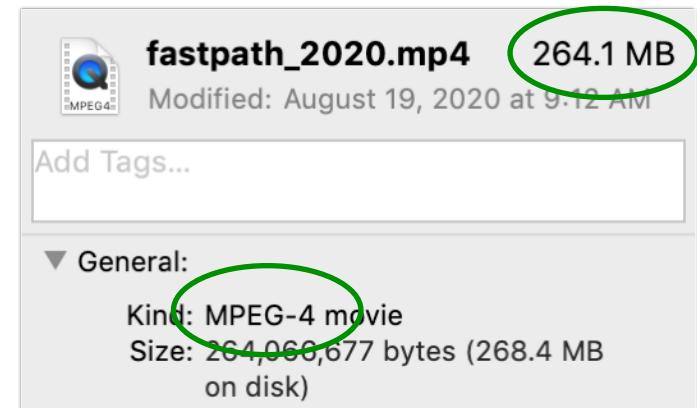
**Compression/encoding done in real-time without you even realizing it!**



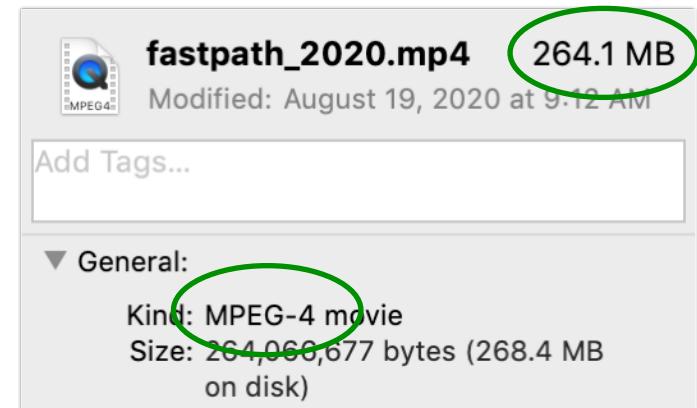
# Video Compression



# Video Compression

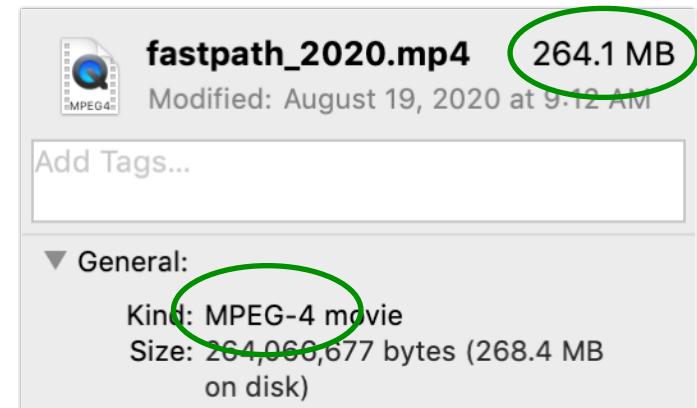


# Video Compression



**WiFi bandwidth test at my home**

# Video Compression



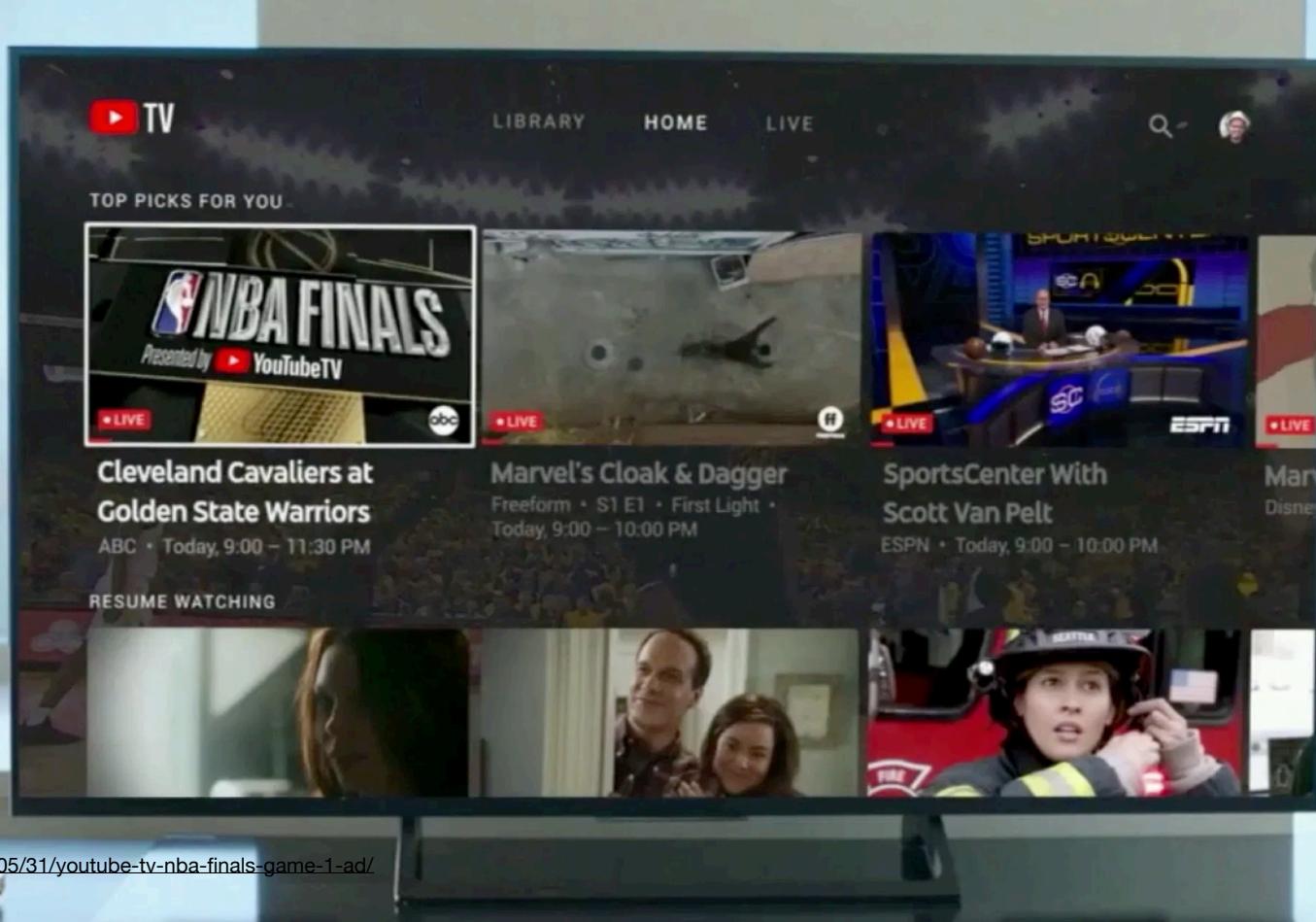
WiFi bandwidth test at my home



$$\frac{3.81 \text{ GB}}{\frac{11.68 \text{ Mb/s}}{8}} \approx 45\text{min}$$
$$\frac{264.1 \text{ MB}}{\frac{11.68 \text{ Mb/s}}{8}} \approx 3\text{min}$$



# Video Compression



<https://9to5google.com/2018/05/31/youtube-tv-nba-finals-game-1-ad/>

# Augmented Reality



<https://techcrunch.com/2018/03/20/wayfairs-android-app-now-lets-you-shop-for-furniture-using-augmented-reality/>

# Virtual Reality



<https://fortune.com/2016/11/15/virtual-reality-gaming-entertainment-tech/>

# Autonomous Machines



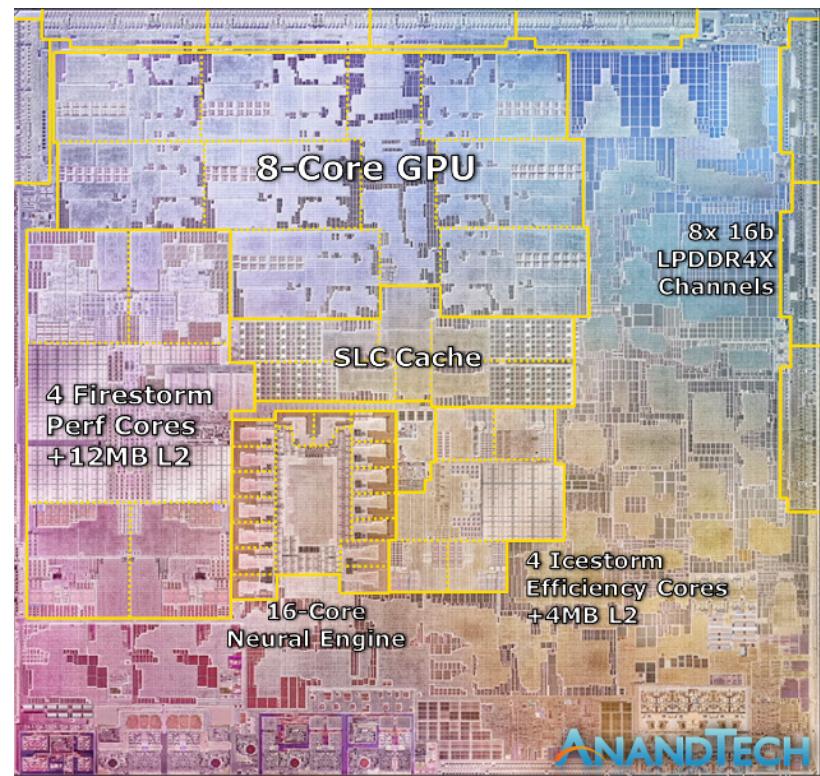
<https://www.wired.com/story/news-rules-clear-way-self-driving-cars/>

# Autonomous Machines



<https://www.wired.com/2017/05/the-physics-of-drones/>

# Today's Processor Chips are Full of Accelerators



# Traditional Scope of Computer Systems

- Take a program and try to figure out how to best execute on the hardware

Problem

---

Algorithm

---

Program

---

Instruction Set  
Architecture

---

Microarchitecture

---

Circuit

# Real Scope of Computer Systems

- Understand the problem to be solved, design algorithms, understand algorithms characteristics to design the best computer systems.
- It's no longer enough to work with a given program without understanding it.

Problem

---

Algorithm

---

Program

---

Instruction Set  
Architecture

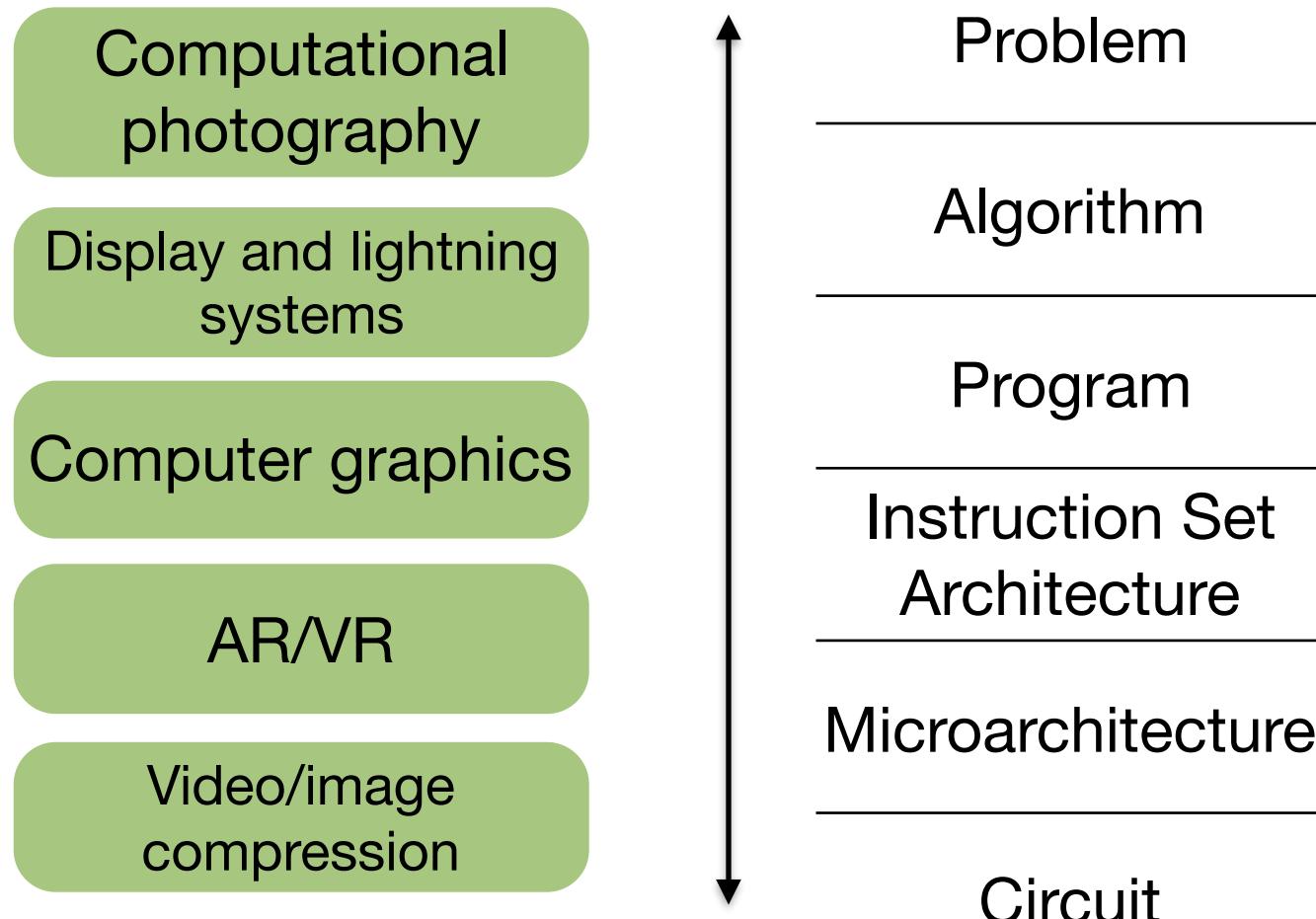
---

Microarchitecture

---

Circuit

# CSC 292/572: Mobile Visual Computing



# The Most Important Take Away of 252

- “There is no magic.”
- Every thing can be derived from first principles. Trust your logical reasoning.
- Apply to virtually everything in science and engineering.

# The Second Most Important Take Away of 252

- “Things don’t have to be this way.”
- As long as you don’t violate physics, you can design a computer however you want.
- But every design decision you make usually involves certain trade-offs. Be clear what your design goal is.

# The Third Most Important Take Away of 252

- Virtually all computer system design practices follow a small set of basic principles.
- It is these basic principles that are important, not the practices.

