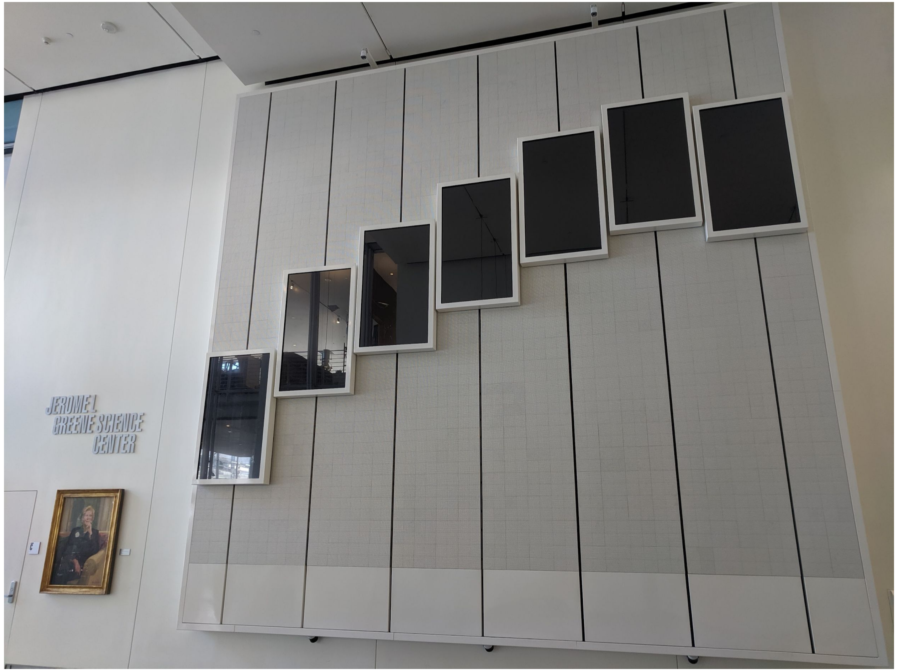
Creative Embedded Systems: Class 4

Generative Art and, like, other Stuff

Today:

- Homework questions
- Review of Lab
  - Observations
- Generative Art



Previously, in 334...

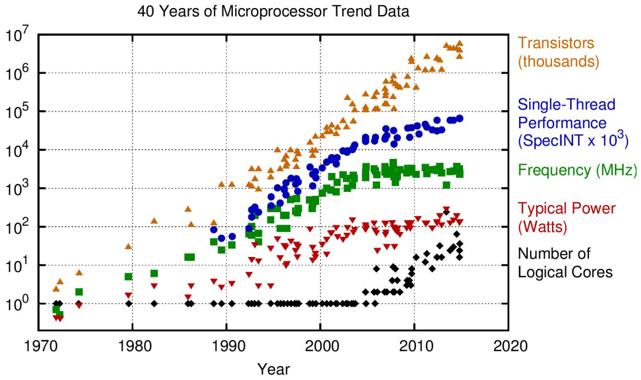
- Connected a screen
- Moved code between devices and benchmarked
- Ran code on boot

notes on canvas

Why RasPi is more than 2x slower than a laptop

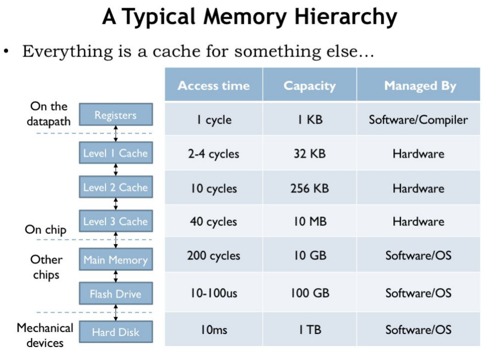
- CPU clockspeed does not define performance
  - RasPi 3B+: 1.4 GhZ, Macbook Pro 2018: 1.4 GhZ*

- Must also consider:
  - Multicore
  - Overclocking
  - Instruction set
  - Instruction pipelining
  - data caching and instruction caching
  - cache hierarchy



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Caching

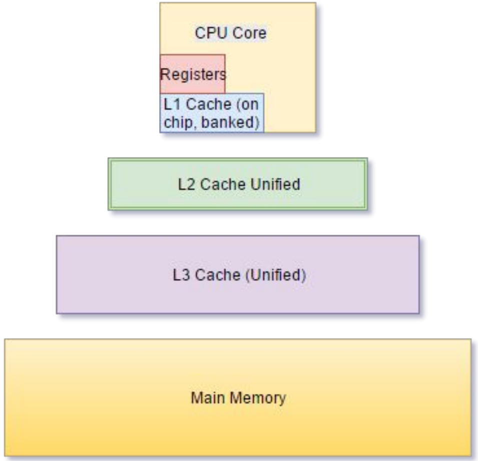


A Typical Memory Hierarchy

- Everything is a cache for something else…

On the datapath
On chip
Other chips
Mechanical devices

	Access time	Capacity	Managed By
Registers	1 cycle	1 KB	Software/Compiler
Level 1 Cache	2-4 cycles	32 KB	Hardware
Level 2 Cache	10 cycles	256 KB	Hardware
Level 3 Cache	40 cycles	10 MB	Hardware
Main Memory	200 cycles	10 GB	Software/OS
Flash Drive	10-100us	100 GB	Software/OS
Hard Disk	10ms	1 TB	Software/OS



page is open

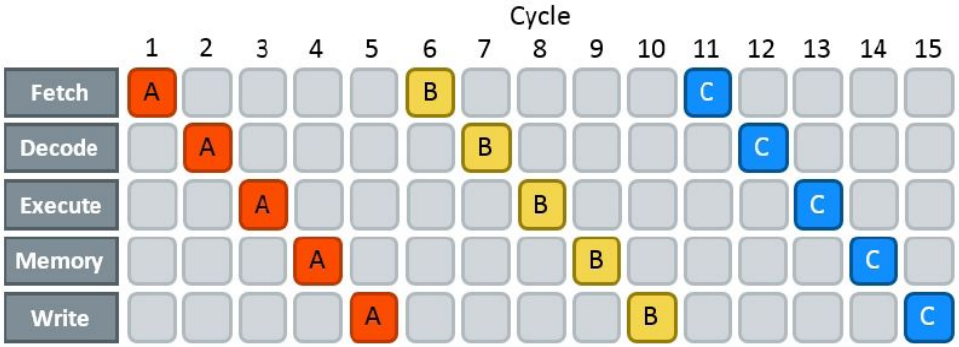
on your desk

in your bookshelf

in Sterling

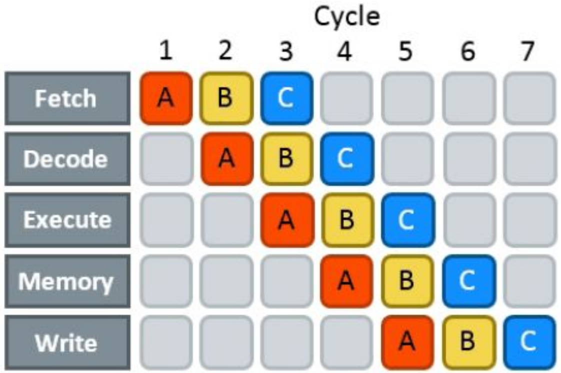
inter-library loan
(network access)

Instruction Pipelining



Cycle

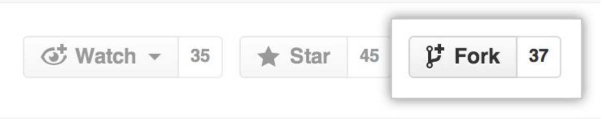
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Fetch	A					B					C				
Decode		A					B					C			
Execute			A					B					C		
Memory				A					B					C	
Write					A					B					C



Cycle

	1	2	3	4	5	6	7
Fetch	A	B	C				
Decode		A	B	C			
Execute			A	B	C		
Memory				A	B	C	
Write					A	B	C

Github



Forks, branches, and merge conflicts, oh my!

For Task 2, you each need your own repository - fork your group's project

When developing a new feature in a repository that others are using, you need to branch

Again we see, dev vs prod

https://guides.github.com/activities/forking/

Generative Art

Overview, sort of.

Generative Systems or works are those created with the assistance of a program that significantly, sometimes entirely, informs the composition and realization of the work.

Generative art can be static (complete) or dynamic (iterative).

Attractions of Generative Strategies

Roads, Composing Electronic Music (pdf)

Characteristics of Generative Systems

- There is usually an impetus, a vision or idea that the algorithm explores. This idea can be simple (landscape) to very complex (organic evolution).

- some element of programmatic change

- some elements of parameterization

- some elements of randomness (output varies, but is deterministic)

- limited required interaction (usually, but not always)

- limited input/output languages (consistency)

- theme (subject) and variation (transformation)

Generative Art Expo (emphasis on Processing)

https://www.behance.net/manoloide

http://manoloide.com/works/terrains/

https://www.creativeapplications.net/processing/zero-one-generative-video-for-zero-one-technology-festival/

https://www.youtube.com/watch?v=LaarVR1AOvs

https://www.youtube.com/watch?v=zT7We17o4zU

https://www.youtube.com/watch?v=IvymqDIf_9g

https://www.kickstarter.com/projects/fbz/knityak-custom-mathematical-knit-scarves

Activity

1. Crowd-source generative algorithms
2. Present your ideas

Crowdsourced Generative Algorithms:

1. Form your teams from Task 1 and crowdsource four (4) generative systems.
2. Each member of the group should conceive a primary idea, a starting point for a piece. (~5')
3. The members of the team then take turns addressing each individual idea and fleshing it out. (~25)

Strategies:

(Your piece ideas should be realistically achievable in the next week…)

- ID how you want to approach the screen divisions (same, different processes?)

- Use algorithms you know (sorting, recursion, whatevs)

- Think conceptually

- Compose with shapes

- Compose with colors

- Organize/conceptualize with animations/motions, sections or transitions

Present your work ~25-30'

Each team should choose one (1) system/piece idea to present to the class (~5' each)

The piece you choose to present should be the "best" conceptually (by consensus) and, simultaneously, the most achievable (realistic)

Your fellow teams (and instructors) will help clarify your vision with questions and suggestions for implementations, etc.