Software Systems
Brian Demsky

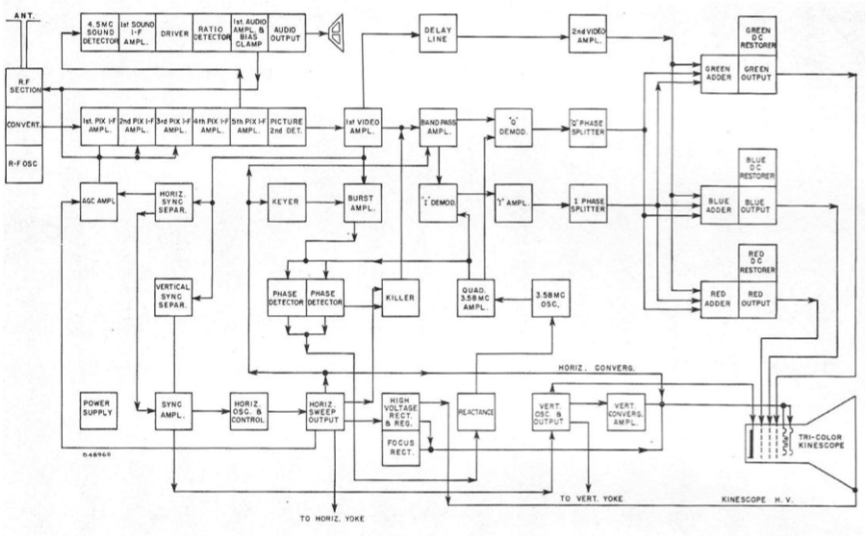
Electrical & Computer Engineering

At the most basic level, we implement computation in matter

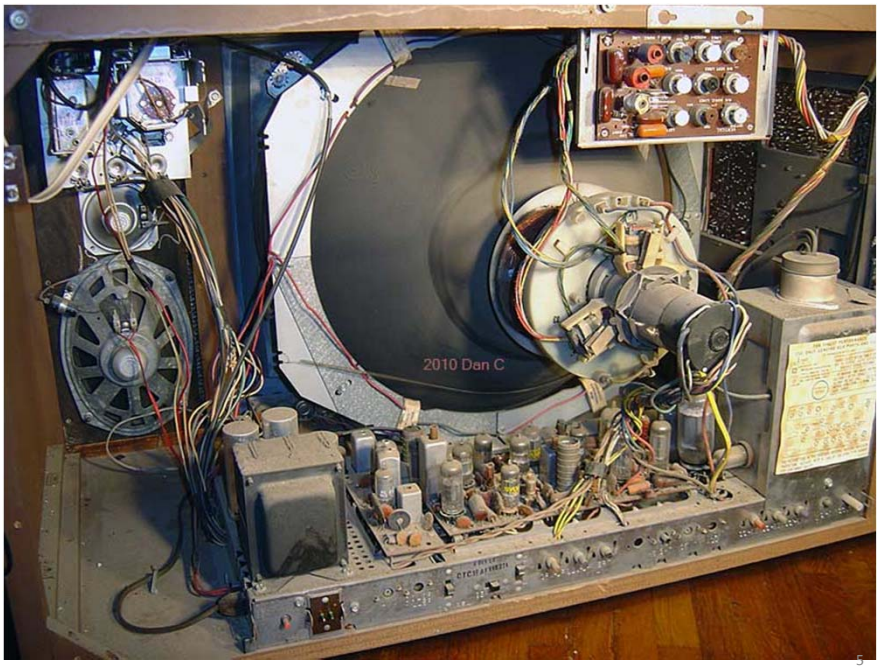
2

The Early Days

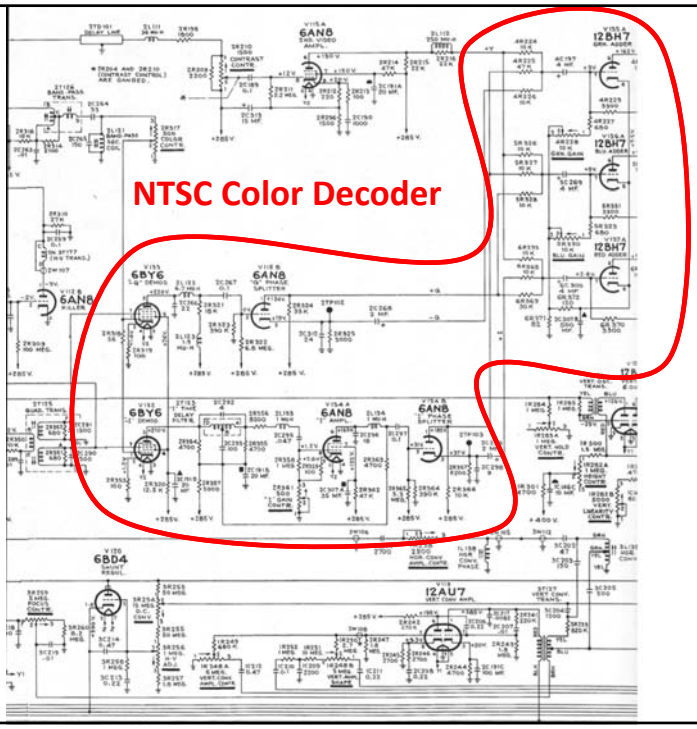
3



4



5



6

Summary

Expensive, big, power-hungry components

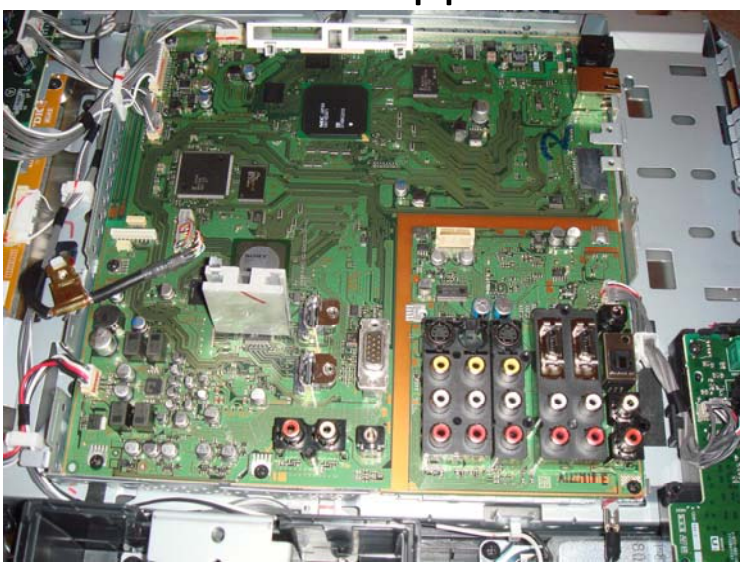
Slow computation

Computation hardwired

Cheap to change layout

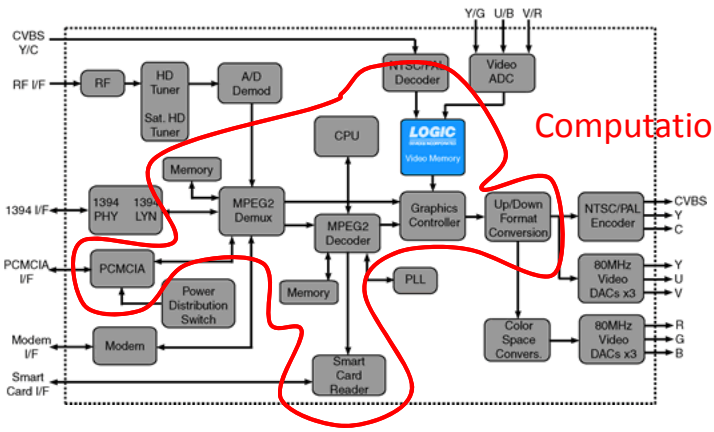
7

A Modern Approach



8

A Modern Approach



Computation


```
float signal_levels[256*8] = {...}; // Eight signal levels for each pixel, normalized to 0..1 range.
Calculated as above.

  unsigned Width; // Input: Screen width. Can be not only 256, but anything up to 2048.
  float phase;    // Input: This should the value that was PPU_cycle_counter * 8 + 3.9
                  // at the BEGINNING of this scanline. It should be modulo 12.
                  // It can additionally include a floating-point hue offset.
  for(unsigned x = 0; x < Width; ++x)
  {
    // Determine the region of scanline signal to sample. Take 12 samples.
    int center = x * (256*8) / Width + 0;
    int begin = center - 6; if(begin < 0)    begin = 0;
    int end   = center + 6; if(end   > 256*8) end   = 256*8;
    float y = 0.f, i = 0.f, q = 0.f; // Calculate the color in YIQ.
    for(int p = begin; p < end; ++p) // Collect and accumulate samples
    {
      float level = signal_levels[p] / 12.f;
      y  =  y + level;
      i  =  i + level * cos( M_PI * (phase+p) / 6 );
      q  =  q + level * sin( M_PI * (phase+p) / 6 );
    }
    render_pixel(y,i,q); // Send the YIQ color for rendering.
  }
```

NTSC Color Decoder

Changes

- Transistors are extremely cheap
- Costs to fabricate new chip designs are huge
- Designs are mostly a handful of analog devices around edges with nearly all computation done digitally

11

Programming Hardware vs. Software

- Computation implemented by both
- Computation for both specified by code
- Differences:
  - Hardware costs more to build
  - Hardware is expensive to change
  - Hardware has limits on complexity and expressiveness
  - Hardware has better performance
  - Software is cheaper to build, cheaper to change, lower performance

12

Future trends

As both hardware design and fabrication costs increases and performance increases

- More computation moves to software
- Less diversity of hardware designs
- Shift from thinking of transistors as components to processors as components
- Most of design complexity in software
- Nearly at the end of scaling for fabrication

13

Software in Embedded Systems

- Control car engines
- Control nearly every device in a home (Ovens, thermostats, refrigerator, …)
- Control factories
- Stop lights

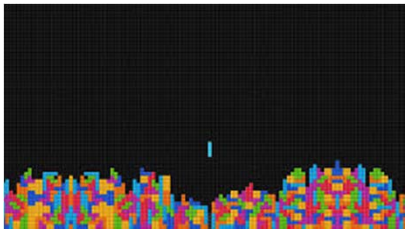
14

Mobile

From Apple

15

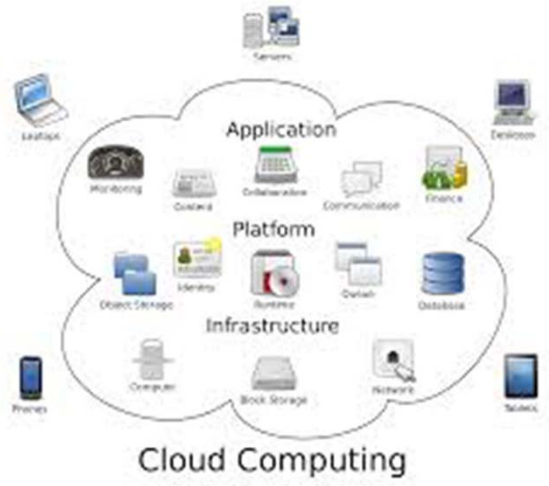
Desktop Software



From Internet

16

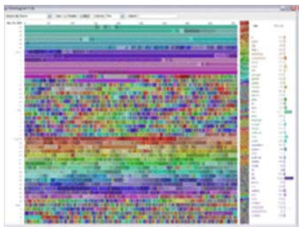
Cloud



From wikipedia

17

Big Data



Visualization of daily Wikipedia edits created by IBM. At multiple terabytes in size, the text and images of Wikipedia are an example of big data.

From wikipedia

18

Proliferation of Web/Mobile Applications

- Google
- Uber
- Netflix
- Airbnb
- SnapChat
- Hulu
- Facebook
- Github
- Amazon
- Lyft
- Nest
- Ebay
- Pinterest
- Waze

19

Internet of Things?

20

Software Classes

- Recommend all students learn programming
  - EECS 10/12 C/Python (required)
  - EECS 22/22L C
  - EECS 40 Java
  - EECS 111 OS
  - EECS 114 Algorithms
  - EECS 118 Software Engineering

21

Software Faculty

Mohammad A. Al Faruque

- Embedded systems

Pai Chou

- Embedded systems/ wireless sensors

22

Software Faculty

Brian Demsky
- Compilers
- Programming Languages
- Parallel Software

Rainer Doemer
- Embedded systems

23

Software Faculty

Kwei-Jay Lin
- Service Oriented Computing

Chen-Yu Phillip Sheu
- Semantic Computing

24

Software Faculty

Aparna Chandramowlishwaran

- High-Performance Computing

25