# Lecture 01
# Introduction to CSE 30321

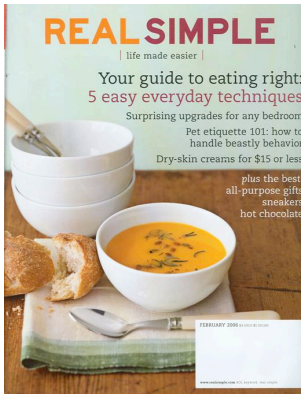




# Huh?

**All of the following are magazines that are regularly delivered to the Niemier household.**







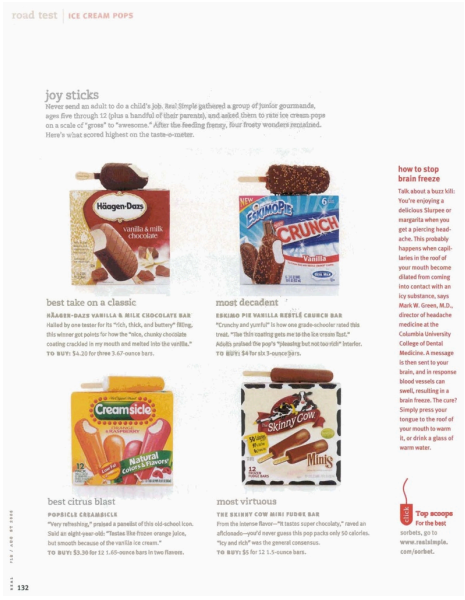



# You can learn about good routes to run if you're visiting Chicago...

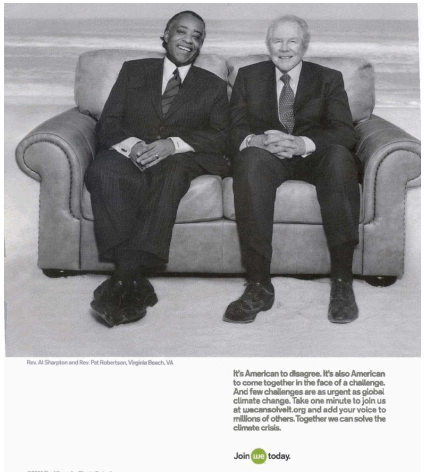





# You can learn about the best ice cream...



# You can read about Pat Robertson and Al Sharpton advocating ways to reduce the effects of global climate change!





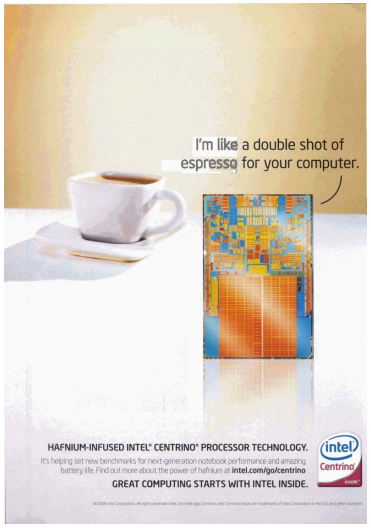



# If you're in the market for a wide-body jet, Boeing has just the thing...






# And if you're looking to buy a computer, Intel suggests their dual core Centrino chip...




**As you might guess, this brings us to CSE 30321!**

**This is essentially a picture of one of Intel's computer architectures...**

**We'll learn about how different parts of a processor are organized and work together**

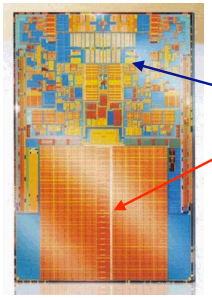




# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - ...describe the fundamental components required in a single core of a modern microprocessor
    - (Also, explain how they interact with each other, with main memory, and with external storage media...)

Example



How do on-chip memory, processor logic, main memory, disk interact?

# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - ...compare and contrast different computer architectures to determine which one performs better...

Example

| Processor | AMD Athlon™ |
|---|---|
| Model | 3200+ |
| OPN Tray | ADA3200AEP5AR |
| OPN PIB | ADA3200BOX |
| Operating Mode 32 Bit | Yes |
| Operating Mode 64 Bit | Yes |
| Revision | |
| Core Speed (MHz) | 2000 |
| Voltages | |
| Max Temps (C) | 70 |
| Wattage | 89 W |
| L1 Cache Size (KB) | 128 |
| L1 Cache Count | 1 |
| L2 Cache Size (KB) | 1024 |
| L2 Cache Count | 1 |

Intel® Pentium® Dual-Core processor

The Intel® Pentium® dual-core processor delivers great performance, low power enhancements, and multitasking for everyday computing.

» Learn more

| Processor Number[1] | Architecture | Cache | Clock Speed | Front Side Bus | Dual-core | Enhanced Intel SpeedStep® Technology[2] | Execute Disable Bit[3] | Intel® 64[4] |
|---|---|---|---|---|---|---|---|---|
| E2220 | 65 nm | 1MB L2 | 2.40 GHz | 800 MHz | ✓ | ✓ | ✓ | ✓ |
| E2200 | 65 nm | 1MB L2 | 2.20 GHz | 800 MHz | ✓ | ✓ | ✓ | ✓ |
| E2180 | 65 nm | 1MB L2 | 2.00 GHz | 800 MHz | ✓ | ✓ | ✓ | ✓ |
| E2160 | 65 nm | 1MB L2 | 1.80 GHz | 800 MHz | ✓ | ✓ | ✓ | ✓ |
| E2140 | 65 nm | 1MB L2 | 1.60 GHz | 800 MHz | ✓ | ✓ | ✓ | ✓ |
| T2330 | 65 nm | 1MB L2 | 1.60 GHz | 533 MHz | ✓ | ✓ | ✓ | ✓ |
| T2310 | 65 nm | 1MB L2 | 1.46 GHz | 533 MHz | ✓ | ✓ | ✓ | ✓ |
| T2130 | 65 nm | 1MB L2 | 1.86 GHz | 533 MHz | ✓ | ✓ | ✓ | |
| T2080 | 65 nm | 1MB L2 | 1.73 GHz | 533 MHz | ✓ | ✓ | ✓ | |
| T2060 | 65 nm | 1MB L2 | 1.60 GHz | 533 MHz | ✓ | ✓ | ✓ | |
| T2370 | 65 nm | 1MB L2 | 1.73 GHz | 533 MHz | ✓ | ✓ | ✓ | ✓ |

If you want to do X, which processor is best?





# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - ...design a processor architecture to meet a specific performance target...

Example

| Find by Feature | |
|---|---|
| Processor | AMD Athlon™ X2 Dual-Core |
| Model Number | 6400+ |
| Frequency (MHz) | 3200 |
| L2 Cache Size (KB) | 1024 |
| Socket | AM2 |
| Stepping | F3 |
| Manufacturing Tech (CMOS) | 90nm SOI |
| Wattage (W) | 125 W |
| System Bus (MHz) | 2000 |
| AMD Business Class | No |

| Find by Feature | |
|---|---|
| Processor | AMD Athlon™ X2 Dual-Core |
| Model Number | 5600+ |
| Frequency (MHz) | 2900 |
| L2 Cache Size (KB) | 512 |
| Socket | AM2 |
| Stepping | G2 |
| Manufacturing Tech (CMOS) | 65nm SOI |
| Wattage (W) | 65 W |
| System Bus (MHz) | 2000 |
| AMD Business Class | No |

You might choose to add more or less on-chip memory...





# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - ...understand how code written in a high-level language (e.g. C) is eventually executed on-chip...

Example

In C:

```
void insertionSort(int numbers[], int array_size)
{
  int i, j, index;

  for (i=1; i < array_size; i++)
  {
    index = numbers[i];
    j = i;
    while ((j > 0) && (numbers[j-1] > index))
    {
      numbers[j] = numbers[j-1];
      j = j - 1;
    }
    numbers[j] = index;
  }
}
```

In Java:

```
public static void insertionSort(int[] list, int length) {
    int firstOutOfOrder, location, temp;

    for(firstOutOfOrder = 1; firstOutOfOrder < length; firstOutOfOrder++) {
        if(list[firstOutOfOrder] < list[firstOutOfOrder - 1]) {
            temp = list[firstOutOfOrder];
            location = firstOutOfOrder;

            do {
                list[location] = list[location-1];
                location--;
            }
            while (location > 0 && list[location-1] > temp);

            list[location] = temp;
        }
    }
}
```

Both programs could be run on the same processor... how does this happen?





# A tangent...

- We'll talk about 2 more course goals in a little bit, but right now, I'd like to ask the class a few questions...
  - Question 1:
    - How many people are EE, CPEG, CS, other?
  - Question 2:
    - By major, does anyone have any definitive thoughts about what they want to do after graduation?
  - Question 3:
    - Preface: last slide talked about SW...
    - How many people are more interested in the SW side of CSE than the HW side of CSE?
  - Question 4:
    - How many people view CSE 30321 as more a "HW course"
    - How many people think other more "SW oriented" courses are more relevant for their major?

# Let's digress...

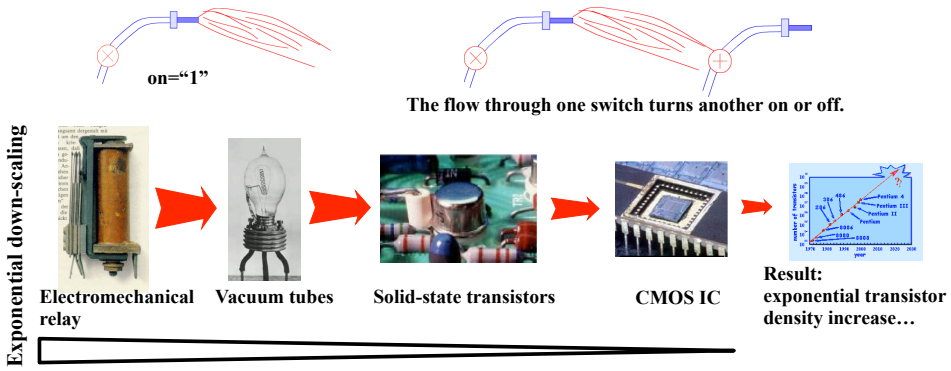
- I asked the questions on the last slide not just to gauge interest, but to bring up an important point...
  - For last 20 years, if interested in SW, computer architecture was probably *not* the most important class for you.
- But...changes in technology are having a profound impact on conventional/established computer architectures
  - We're presently at the very beginning of this storm...
- We'll need significant engagement from programmers to continue the processor performance scaling trends of the last 40 years...
  - ... this will impact your career...





# A little history... Zuse's paradigm

- Konrad Zuse (1938) Z3 machine
  - Use binary numbers to encode information
  - Represent binary digits as on/off state of a current switch








# A little history... programs

- Stored program model has been around for a long time...

First Draft of a Report
on the EDVAC

by

John von Neumann

Contract No. W–670–ORD–4926

Between the

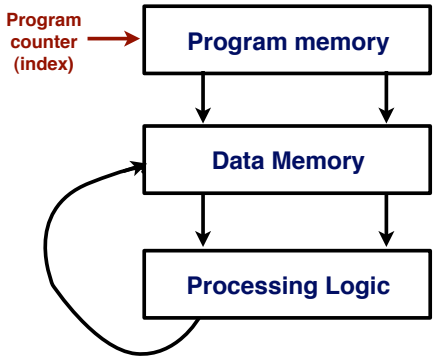
United States Army Ordnance Department

and the

University of Pennsylvania

Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

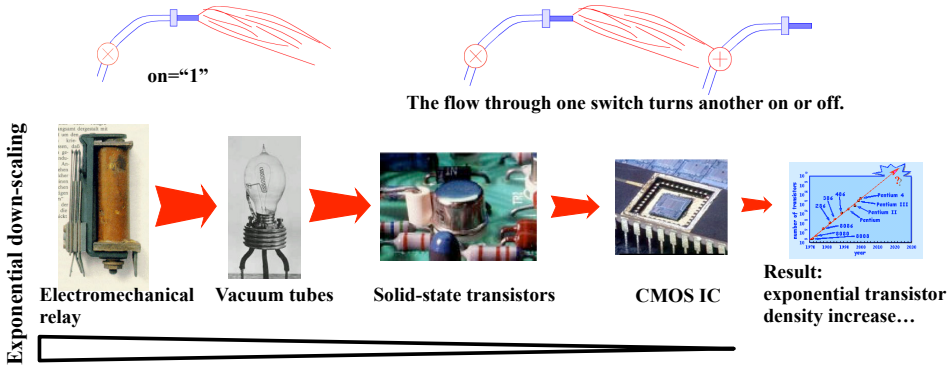






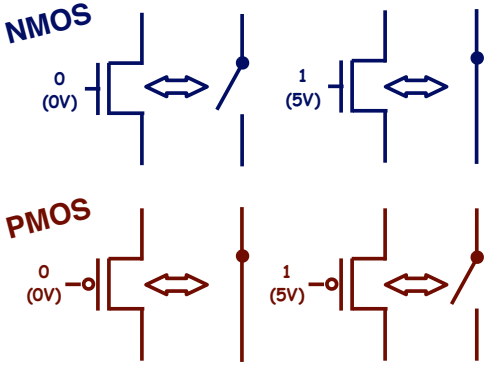

# A little history... Zuse's paradigm

- Konrad Zuse (1938) Z3 machine
  - Use binary numbers to encode information
  - Represent binary digits as on/off state of a current switch

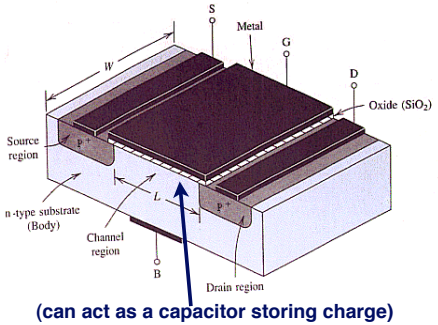


# Transistors used to manipulate/store 1s & 0s

**Switch-level representation**




**Cross-sectional view**




(can act as a capacitor storing charge)

**Using above diagrams as context, note that (with NMOS) if we (i) apply a suitable voltage to the gate & (ii) then apply a suitable voltage between source and drain, current will flow.**





# Moore's Law

- **"Cramming more components onto integrated circuits."**

  **- G.E. Moore, Electronics 1965**

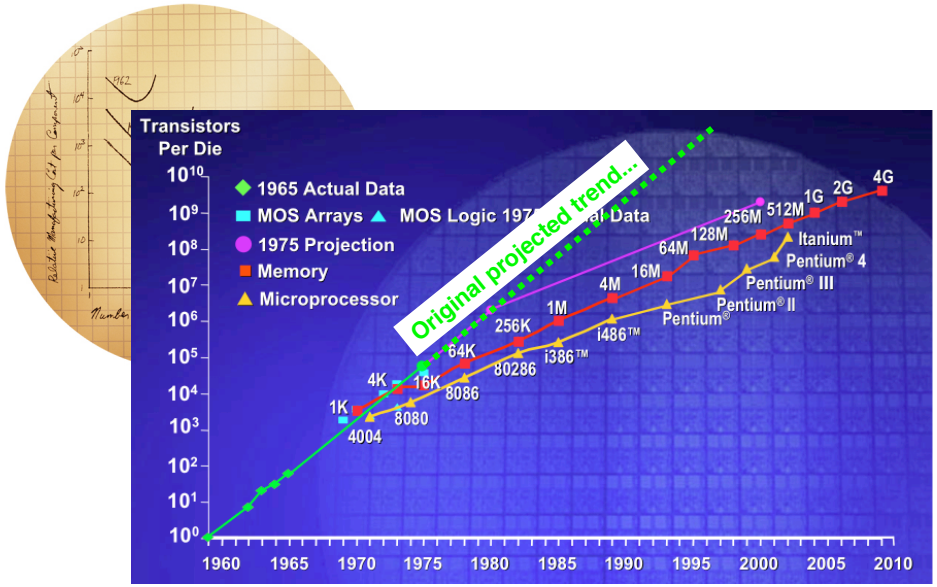
- **Observation: DRAM transistor density doubles annually**
  - **Became known as "Moore's Law"**
  - **Actually, a bit off:**
    - Density doubles every 18 months (now more like 24)
    - (in 1965 they only had 4 data points!)
- **Corollaries:**
  - **Cost per transistor halves annually (18 months)**
  - **Power per transistor decreases with scaling**
  - **Speed increases with scaling**
  - **Reliability increases with scaling**
    - Of course, it depends on how small you try to make things
      - » (I.e. no exponential lasts forever)

**Remember these!**





# Moore's Law








# Feature sizes...

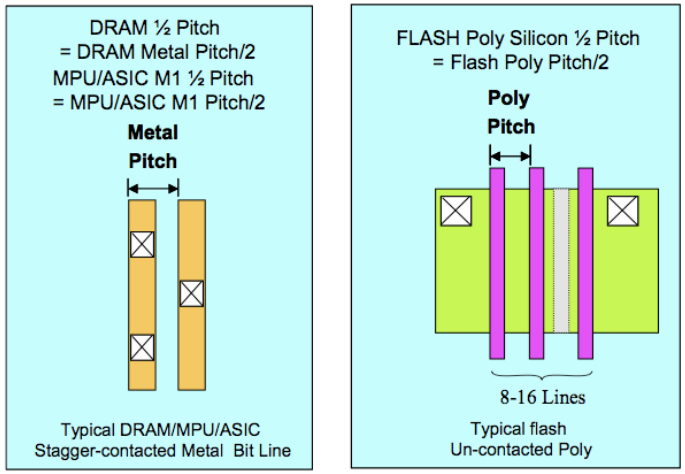



*Figure 2 2005 Definition of Pitches*

# Moore's Law

- Moore's Curve is a self-fulfilling prophecy
  - 2X every 2 years means ~3% per month
    - I.e. ((1 X 1.03) * 1.03)*1.03… 24 times = ~2
  - Can use 3% per month to judge performance features
  - If feature adds 9 months to schedule…it should add at least 30% to performance
    - ($1.03^9 = 1.30$ ⇨ 30%)





# A bit on device performance...

- One way to think about switching time:
  - Charge is carried by electrons
  - Carrier velocity is proportional to the lateral E-field between source and drain
    - i.e. $v = mE$
      - m = carrier mobility (and can be though of as a constant)
  - Electric field defined as: $E = V_{ds}/L$
  - Time for charge to cross channel = length/speed
    - (i.e. meters / (meters/s) = seconds)
    - $= L/v$
    - $= L/(mE)$
    - $= L/(m*(V_{ds}/L))$
    - $= L^2/(mV_{ds})$

Thus, to make a device faster, we want to either increase $V_{ds}$ or decrease feature sizes (i.e. L)





# Some more important relationships

- What about power (i.e. heat)?
  - First, need to quickly discuss equation for capacitance:
    - $C_L = (e_{ox}WL)/d$
      - $e_{ox}$ = dielectric, WL = parallel plate area, d = distance between gate and substrate
  - Then, dynamic power becomes:
    - $P_{dyn} = C_L V_{dd}^2 f_{0\text{-}1}$
      - Dynamic power is a function of the frequency of 0 to 1 or 1 to 0 transitions (as this involves the movement of charge)
        - » Note frequency in this context is NOT clock frequency
      - Note that as W and L scale, $C_L$ decreases which in turn will cause a decrease in $P_{dyn}$.
      - Note that while an increase in $V_{dd}$ will *decrease* switching time, it will also cause a quadratic *increase* in dynamic power.

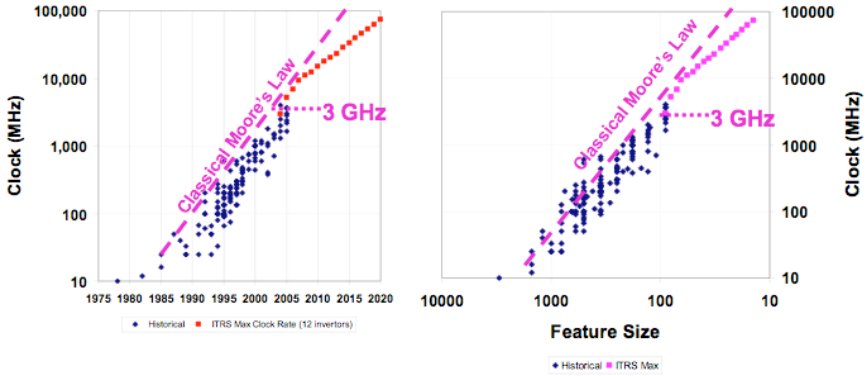




# A funny thing happened on the way to 45 nm

- Speed increases with scaling...

Remember these!

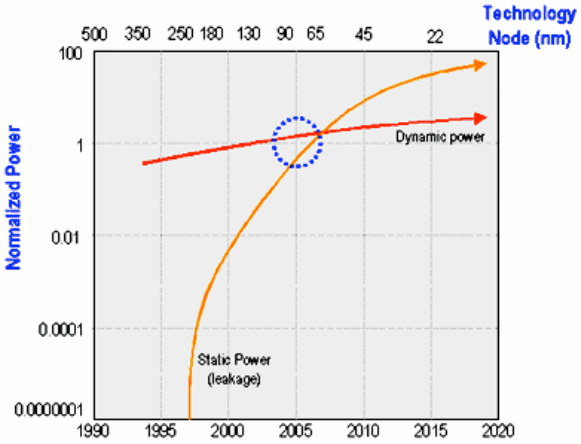



2005 projection was for 5.2 GHz - and we didn't make it in production. Further, we're still stuck at 3+ GHz in production.

# A funny thing happened on the way to 45 nm

- Power decreases with scaling...

Remember these!





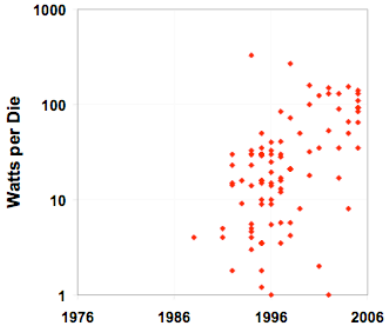



# A funny thing happened on the way to 45 nm

- Speed increases with scaling...
- Power decreases with scaling...

Remember these!

**Why the clock flattening? POWER!!!!**





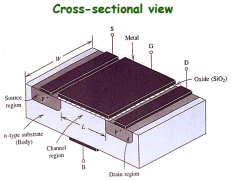


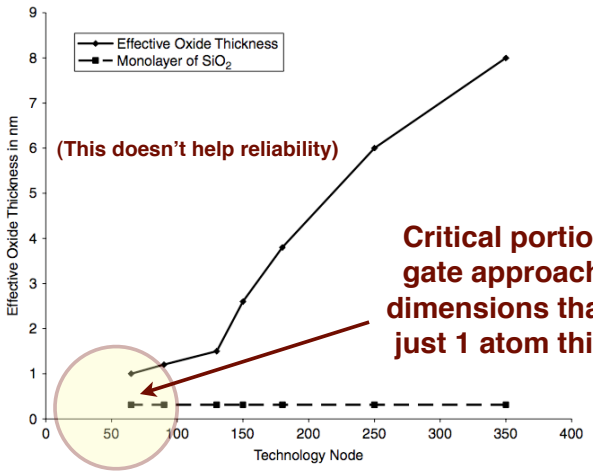

# A funny thing happened on the way to 45 nm

- Reliability increases with scaling...

Remember these!

One quick example:




(This doesn't help reliability)

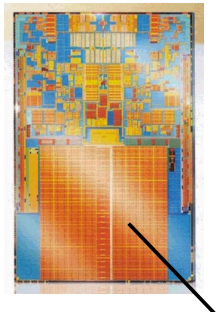
Critical portion of gate approaching dimensions that are just 1 atom thick!!!





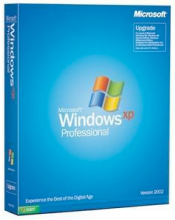
# Other reasons too, but this should give you a good feel for technology...

# Transistors used for memory too...



Saw earlier, that transistors used for *on chip* memory too...

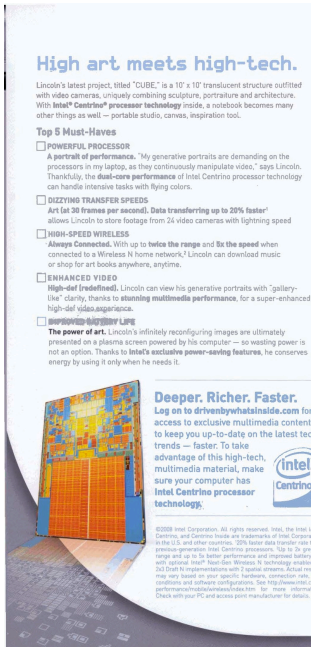
Problem: Program, data = bigger + power problem isn't going away...




- Why? Put faster memory closer to processing logic...
  - SRAM (logic): density +25%, speed +20%
  - DRAM (memory): density +60%, speed +4%
  - Disk (magnetic): density +25%, speed +4%



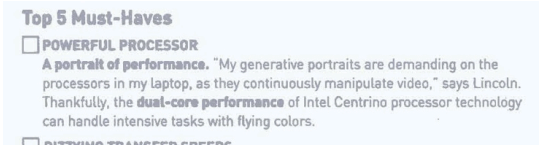





# Solution?

Motivation:
Processor complexity is good enough,
transistor sizes scale,
we can slow processors down,
manage power,
and get performance from...

**Parallelism**



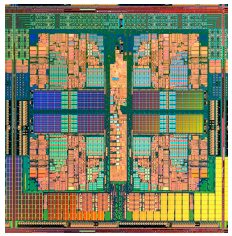

(i.e. 1 processor, 1 ns clock cycle
vs.
2 processors, 2 ns clock cycle)



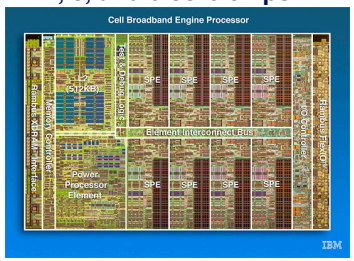


# This idea has been extended...

Quad core chips...



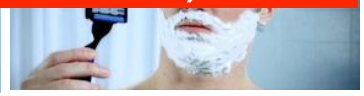
7, 8, and 9 core chips...





Practical problems must be addressed!

**Advances in parallel programming are necessary!**
**(We'll get into later in the semester)**







# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - ...explain and articulate why modern microprocessors now have more than 1 core...
- Why?
  - For 8, 16 core chips to be practical, we have to be able to *use them*
    - Students in this class should go on to play a role in making such chips useful...

# So, what are the goals of this course?

- At the end of the semester, you should be able to...
  - Apply fundamental knowledge about single core machines, dual core machines, performance metrics, etc. to design a microprocessor such that it (a) meets a target set of performance goals and (b) is realistically implementable.





# Now, let's look at the syllabus