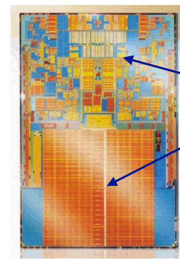
Lecture 17a
A short review

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?

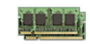
2.0 GB

$200.00

Apple Memory Module 2GB 667MHz DDR2 (PC2-5300) 2x1GB SO-DIMMs

Estimated Ship: Within 24 hours

Free Shipping



750GB SATA Hard Disk Drive Kit for...

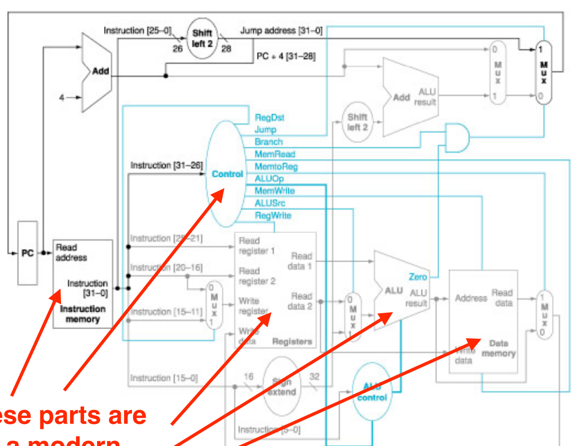
Ships: Within 24hrs

Free Shipping

$299.00

We're halfway there…

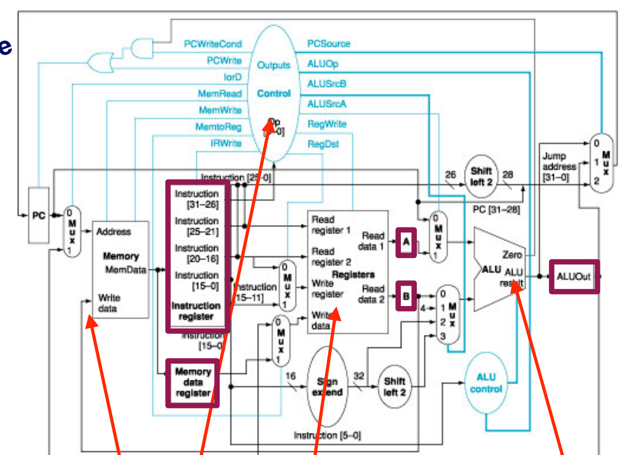
Single Cycle



All of these parts are part of a modern processor's datapath

We're halfway there…

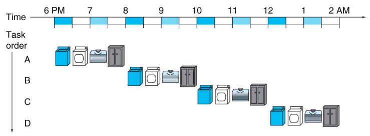
Multi Cycle

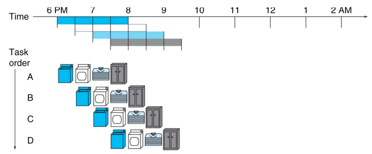


However, they may be organized slightly differently…

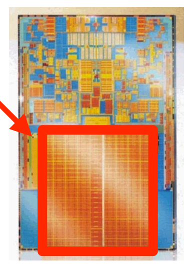
Next…

Pipelining…





…and memory…



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

AMD Smarter Choice

Processor	AMD Athlon™
Model	3200+
OPN Tray	ADA3200AEP5AR
OPN PIB	ADA3200BOX
Operating Mode 32 Bit	Yes
Operating Mode 64 Bit	Yes
Revision	[illegible]
Core Speed (MHz)	2000
Voltages	1.50V
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 Number¹	Architecture	Cache	Clock Speed	Front Side Bus	Dual-core	Enhanced Intel SpeedStep® Technology²	Execute Disable Bit°	Intel® 64Φ
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	[illegible]	533 MHz	✓	✓	✓	
T2370	65 nm	1MB L2	1.73 GHz	533 MHz	✓	✓	✓	✓

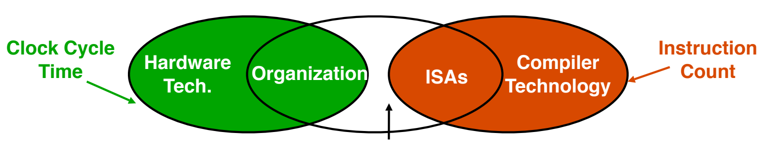
If you want to do X, which processor is best?

An important idea…

A common denominator

$$\frac{Instructions}{Program} \times \frac{Clock\ cycles}{Instruction} \times \frac{Seconds}{Clock\ Cycle} = \frac{Seconds}{Program} = CPU\ time$$

- We can see CPU performance dependent on:
  - Clock rate, CPI, and instruction count
- CPU time is directly proportional to all 3:
  - Therefore an **x %** improvement in any one variable leads to an **x %** improvement in CPU performance
- But, everything usually affects everything:



More in Lab 05

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…

Example

- Calculate cycle time assuming negligible delays except:
  - memory (2ns), ALU and adders (2ns), register file access (1ns)
  - R-type: max {mem + RF + ALU + RF, Add} = 6ns
  - LW: max{mem + RF + ALU + mem + RF, Add} = 8ns
  - SW: max{mem + RF + ALU + mem, Add} = 7ns
  - BEQ: max{mem + RF + ALU, max{Add, mem + Add}} = 5ns

Looking at the multi-cycle datapath, we might optimize a state to make it faster ... even if the CPI increases for some cases

More in Final Project, HW 06

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?

Example 1

x = A(i) -- w/A(i) in memory

6 instruction processor

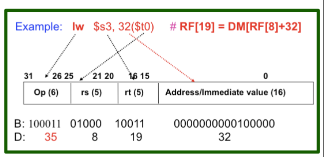
MOV R1, address

Load instruction—0000 $r_3r_2r_1r_0$ $d_7d_6d_5d_4d_3d_2d_1d_0$

Therefore address to memory comes from instruction *encoding* (or IR)

MIPS

LW $1, 0($5)



$1 <-- Mem(0 + RF(5))

Therefore the address to memory must come from a register.

Example 2

x = y + z

6 instruction processor

Add R1, R2, R3

Therefore ALU operation operates on data in registers and stores the result back to a register.

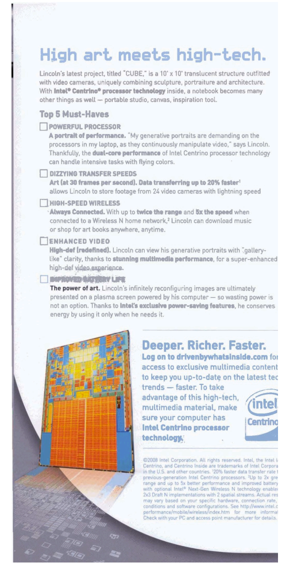
MIPS

Add $1, $2, $3

Therefore ALU operation operates on data in registers and stores the result back to a register.

So, what are the goals of this course?

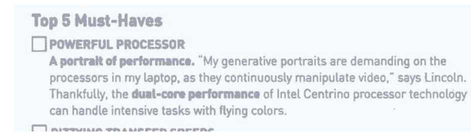
- At the end of the semester, you should be able to…
  - …explain and articulate why modern microprocessors now have more than 1 core…
- Why?
  - For 8, 16 core chips to be practical, we have to be able to *use them*
    - Students in this class should go on to play a role in making such chips useful…



Solution?

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

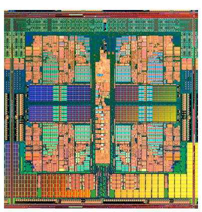
Parallelism



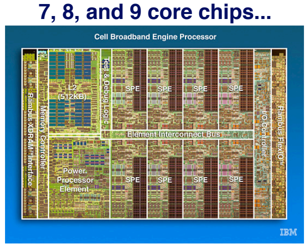
(i.e. 1 processor, 1 ns clock cycle
vs.
2 processors, 2 ns clock cycle)

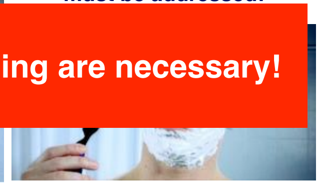
This idea has been extended…

Quad core chips…



7, 8, and 9 core chips…





Practical problems must be addressed!

Advances in parallel programming are necessary!

We'll start this today!