

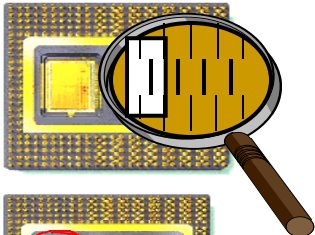
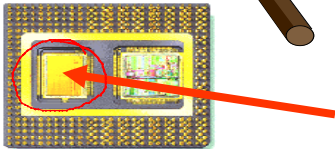




# Compilers

---

## Managing Caches

# Managing Caches

	Registers	<u>1 cycle</u>	<u>256-8000 bytes</u>
	Cache	<u>3 cycles</u>	256k-1M
	Main memory	<u>20-100 cycles</u>	<u>32M-4G</u>
	Disk	0.5-5M cycles	4G-1T

- Power usage limits
  - Size and speed of registers/caches
  - Speed of processors
- But
  - The cost of a cache miss is very high
  - Typically requires 2 caches to bridge fast processor with large main memory
- It is very important to:
  - Manage registers properly
  - Manage caches properly

- Compilers are very good at managing registers
  - Much better than a programmer could be
- Compilers are not good at managing caches
  - This problem is still left to programmers
  - It is still an open question how much a compiler can do to improve cache performance
- Compilers can, and a few do, perform some cache optimizations

- Consider the loop

a[1] b[1]

→ for(j := 1; j < 10; j++)  
→ for(i=1; i < 10000000; i++)  
→ a[i] \* = b[i]

a[1] b[1] a[2] b[2] x x x

all  
misses

→ a[1] \* = b[1]  
→ a[2] \* = b[2]

- Consider the program:

```
→ for(i=1; i<1000000; i++)  
    → for(j := 1; j < 10; j++)  
        a[i] *= b[i]
```

- Computes the same thing
  - But with much better cache behavior
  - Might be more than 10x faster
- A compiler can perform this loop interchange optimization