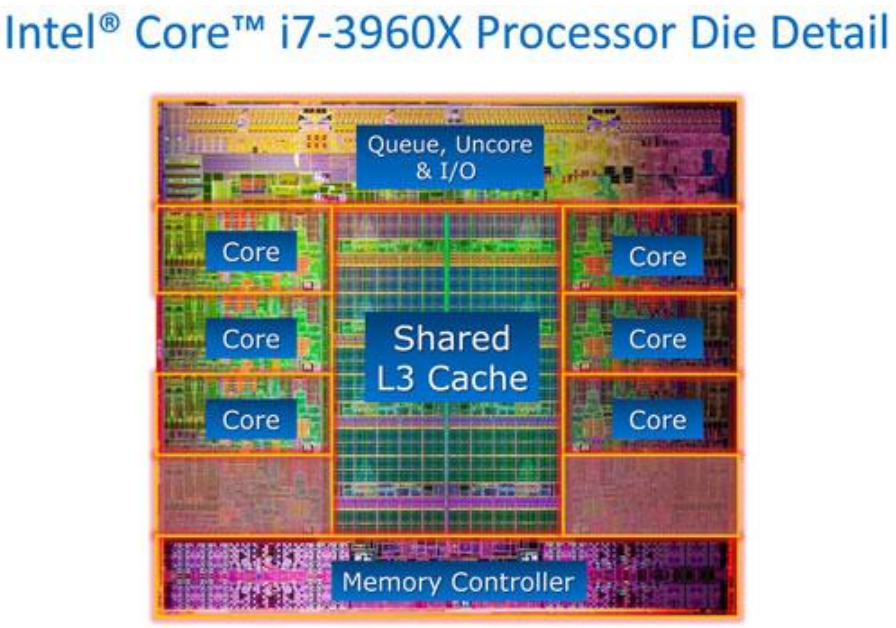
Cache

J. Wunderlich PhD Handout

(from http://www.computerhope.com/jargon/c/cache.htm):

1. .**MEMORY CACHE** is a portion of the high-speed static RAM (SRAM) and is effective because most programs access the same data or instructions over and over. By keeping as much of this information as possible in SRAM, the **CPU** avoids accessing the slower DRAM, making the computer perform faster and more efficiently. Today, most computers come with L3 cache or L2 cache, while older computers included only L1 cache. Below is an example of the Intel i7 processor **(CPU)** and its shared L3 cache.



2. **Cache**, **INTERNET BROWSER CACHE**, or **temporary Internet files** with an Internet browser, like Google Chrome, Firefox, or Internet Explorer, is used to improve how fast data loads while browsing the Internet. In most cases, each time you open a web page, the page and all its files are sent to the browser's temporary cache on the hard drive. If the web page and files contained on that web page (e.g. pictures) need to load again and have not been modified, the browser opens the page from your cache instead of downloading the page again. Cache saves lots of time, especially if you use a modem, and can also help save on bandwidth for the website owner.

3. Like memory **caching**, **DISK CACHING** is used to access commonly accessed data. However, instead of using high-speed SRAM, a disk cache uses conventional main memory. The most recently accessed data from a disk is stored in a memory buffer. When a program needs to access data from the disk, it first checks the disk cache to see if the data is there. Disk caching can dramatically improve the performance of applications because accessing a byte of data in RAM can be thousands of times faster than accessing a byte of data on a hard drive.

4. A **CACHE SERVER** is a computer or network device set up to store web pages that have been accessed by users on a network. Any user trying to access a web page stored on the cache server is sent the stored version, instead of downloading the web page again. Cache servers help reduce network and Internet traffic congestion, as well as save the company on bandwidth costs.

Computer Hardware **MEMORY CACHE DESIGN** *Intro*

by J. Wunderlich PhD

We use **PROBABILITY THEORY** to **SPEED-UP** Computer Performance
by applying the principal of **LOCALITY OF REFERENCE**

TEMPORAL Locality of Reference

*For Design of Memory Caches, which are more expensive, less-dense, but faster RAM (SRAM vs. DRAM), we want to hold copies of data or instructions that were recently used so that the processor(s) (or cores) can access data and instructions faster the next time these things are needed. (i.e., things that are **PROBABLY** needed next)*

SPATIAL Locality of Reference

*For Design of Memory Caches, which are more expensive, less-dense, but faster RAM (SRAM vs. DRAM), we want to hold copies of data or instructions in chunks, called **"Cache Lines" or "Cache Blocks"**(different names for the same thing). When we have a **Cache Miss** (i.e., when the **CPU** looks in a Cache for something, but it's not there), we grab a big chunk (not too big) of Data or Instructions because **PROBABLY** the processor(s) (or cores) will want to access things physically located next to each other when it/they look(s) in the Cache the next time, and the time after that. Examples are **Data in an Array, or Instructions in a Loop.***

***"Cache Line" or "Cache Block" size** is optimized based on the **Law of Diminishing Marginal Returns** (like in Business or Economics) where you gain speedup as you increase cache Line/Block size, but you gain a little bit less with each incremental increase in size. And at some point, you actually begin loosing speedup because you're taking up too much of the overall cache size with individual Lines/Blocks, and therefore the **PROBABILITY** of having a needed Cache Line/Block in the Cache is reduced.*

Examples from J Wunderlich Lecture http://users.etown.edu/w/wunderjt/ITALY_2009/TALK_COMPUTERS.pdf :