What is the difference between an accumulator instruction set architecture and a general-purpose register instruction set architecture?

These represent two points in the design-space for instruction sets. Consider what a CPU instruction needs to do. For now, just consider *compute* instructions. At a minimum, it needs to provide the following information:

What computation to perform (add, subtract, shift, compare)

What values to perform the computation on (the inputs to the computation)

Where to put the result (the output of the computation)

For example, if I wrote the high-level program statement X = Y + Z, you need to tell the processor to fetch the values held in Y and Z, add them, and write the result to X.

In an accumulator architecture, most compute instructions operate on a special register called the accumulator. Most operations, therefore, have the accumulator as an *implicit* argument to the instruction. The accumulator either provides an input to the instruction, receives the output from the instruction, or both. To perform X = Y + Z on an accumulator-based machine, the instruction sequence would look roughly like this:

Load Y into the accumulator

Add Z to the accumulator

Store accumulator to X

If I had a more complex expression, such as "I = J + K + L + M + N + O", the sequence might look like this:

Load J into the accumulator

Add K to the accumulator

Add L to the accumulator

Add M to the accumulator

Add N to the accumulator

Add O to the accumulator

Store accumulator to I

A program counter is a [register](http://whatis.techtarget.com/definition/register) in a computer [processor](http://searchcio-midmarket.techtarget.com/definition/processor) that contains the address (location) of the [instruction](http://searchcio-midmarket.techtarget.com/definition/instruction) being executed at the current time. As each instruction gets [fetched](http://searchsqlserver.techtarget.com/definition/fetch), the program counter increases its stored value by 1. After each instruction is fetched, the program counter points to the next instruction in the sequence. When the computer restarts or is reset, the program counter normally reverts to 0.

**Secondary Memory**

Secondary memory is where programs and data are kept on a long-term basis. Common secondary storage devices are the hard disk and optical disks.

* The hard disk has enormous storage capacity compared to main memory.
* The hard disk is usually contained inside the case of a computer.
* The hard disk is used for long-term storage of programs and data.
* Data and programs on the hard disk are organized into files.
* A **file** is a collection of data on the disk that has a name.

A hard disk might have a storage capacity of 500 gigabytes (room for about 500 x 109 characters). This is about 100 times the capacity of main memory. A hard disk is slow compared to main memory. If the disk were the only type of memory the computer system would slow down to a crawl. The reason for having two types of storage is this difference in speed and capacity.

Large blocks of data are copied from disk into main memory. The operation is slow, but lots of data is copied. Then the processor can quickly read and write small sections of that data in main memory. When it is done, a large block of data is written to disk.

Often, while the processor is computing with one block of data in main memory, the next block of data from disk is read into another section of main memory and made ready for the processor. One of the jobs of an operating system is to manage main storage and disks this way.

|  |  |
| --- | --- |
| **Primary memory** | **Secondary memory** |
| * Fast * Expensive * Low capacity * Works directly with the processor | * Slow * Cheap * Large capacity * Not connected directly to the processor |

Secondary Memory

Secondary memory refers to [storage devices](https://techterms.com/definition/storagedevice), such as [hard drives](https://techterms.com/definition/harddrive) and [solid state drives](https://techterms.com/definition/ssd). It may also refer to removable storage media, such as USB [flash drives](https://techterms.com/definition/flashdrive), [CDs](https://techterms.com/definition/cd), and [DVDs](https://techterms.com/definition/dvd).

Unlike [primary memory](https://techterms.com/definition/primary_memory), secondary memory is not accessed directly by the [CPU](https://techterms.com/definition/cpu). Instead, [data](https://techterms.com/definition/data) accessed from secondary memory is first loaded into [RAM](https://techterms.com/definition/ram) and is then sent to the [processor](https://techterms.com/definition/processor). The RAM plays an important intermediate role, since it provides much faster data access speeds than secondary memory. By loading software [programs](https://techterms.com/definition/program) and [files](https://techterms.com/definition/file) into primary memory, computers can process data much more quickly.

While secondary memory is much slower than primary memory, it typically offers far greater storage capacity. For example, a computer may have a one [terabyte](https://techterms.com/definition/terabyte) hard drive, but only 16 [gigabytes](https://techterms.com/definition/gigabyte) of RAM. That means the computer has roughly 64 times more secondary memory than primary memory. Additionally, secondary memory is non-volatile, meaning it retains its data with or without electrical power. RAM, on the other hand, is erased when a computer is shut down or restarted. Therefore, secondary memory is used to store "permanent data," such as the [operating system](https://techterms.com/definition/operating_system), [applications](https://techterms.com/definition/application), and user files.