

The future of computers 14765

[Technology](#), [Computer](#)



The era of the First Generation computers began around 1945 and ended around 1957. These Machines featured components such as vacuum tubes, drum memories and had to be programmed in machine code. (That is the 1's and 0's computers understand) The Second Generation computers began around 1957 and ended around 1963. These computers had components such as transistors instead of vacuum tubes, magnetic core memories and higher level programming languages called assembly code was introduced. Third Generation computers began around 1963 and ended around 1971. They included integrated circuits, semiconductor memory, magnetic disk storage, and virtual memory. Microcomputers, operating systems and time sharing software concepts were all developed during this period in the development of the computer. The following generation of computers, the fourth generation, began around 1971 and continues is drawing to a conclusion as we enter the new millennium. This group features microprocessors, very large scale integration, networking, database management systems, advanced programming languages such as Pascal, basic, C, C++, Java, etc. As we can see computers as we know them today have enjoyed a phenomenal technological growth rate from the day the first one was booted, if the growth rate in the future is to resemble anything of the dramatic changes seen in the past significant research and development must be applied to several key areas in the field. Microprocessors, parallel processing systems and other architectures, optical technologies, molecular technologies are some of these key areas. Parallel processing. Parallel processing computers probably have a place in tomorrow's world of computers. This computer architecture involves hundreds or even thousands

of processors linked to perform hundreds of tasks simultaneously. Tasks like processing hundreds of lines of code, accessing video files, or audio files, playing live media from the web, etc. The mathematician John Von Neuman who laid the foundation for serial computer architecture recognized the potential of parallel processing but put the idea aside due to the great cost of tubes and wiring. ENIAC, the first general purpose electronic computer was the first parallel computer. However, in 1948 the separate units were centralized and reprogrammed to accept serial input. ENIAC was recalibrated due to limitations of available technology, computer memories were only capable of storing several thousand bits of data and it had to access it a bit at a time. Granularity is the most important feature for the classification of parallel processing computers. Coarse-grain systems, which have the most powerful processors, contain anywhere from 2 to about 200 processors. Medium-grained multiprocessor systems contain less powerful processors but have about 10, 000 individual processors. Fine-grained multiprocessor systems contain from 100, 000 to 10 million processors. Today parallel processing research proceeds in two directions a) improving single processor speeds by applying parallel techniques of computation, and b) by building multiprocessor systems from the ground up. Bus-based Architecture. These systems use currently available CPU's that are tied together and given access to a global or common memory that processors can reach via a central communications channel. Any processor can leave data here for others. This arrangement can however lead to communication overload when the amounts processors manipulating this memory surpasses capacity, thus bus based systems are normally limited to around 20 processors.

Multiple-instruction-stream/Multiple-data-stream is how these machines get they're programming. Its is an operation where programs are broken into several pieces distributed along with data to individual processors when they run independently. The biggest drawback of share memory is the memory itself. Fighting congestion and bottlenecking (too many processors trying to access and manipulate data at the same time) is difficult and expensive.

That's why in some systems all the processors are allocated their own memory in a scheme called single-instruction/multiple data processing. In these systems data is sent at the beginning of the computations to all the processors involved. During computation however many processors will need to share data with others or receive additional input. Therefore, for efficient rapid memory accessing and data distribution, the quickest shortest connections for communication must be attained, amount of interpathways necessary must be minimized and the right scheme for distributed memory to the application at hand must be matched. N-Dimensional Cube

Architecture. The n-dimensional cube also known as the hypercube is a multidimensional shape of n dimensions, hence is other name the n-cube. To create a one dimensional plane we connect two points in space, for a two dimensional plane we take two lines and connect each of the two points to the corresponding points on the other line to form a square. For three dimensions we take the two squares and connect their corresponding points to form a cube. For the 4th dimension we do the something and so on and so forth. So in general and n-dimensional cube will have 2^n nodes because there would be a processor at each connection. So in doubling the number of connections (i. e. each additional dimension doubles the number or

connections) we get a more efficient way of connecting many processors together. In a four dimensional hypercube the longest path connecting two processors four communication lines. The unique structure precludes the need to wire every single processor to every other and still ensure an efficient data path. So in an n-dimensional hypercube the longest path between any two nodes is n. Current practical limitations place hypercubes at a maximum of 16 dimensions were 65, 536 processors are interconnected. Their theoretical peak speeds are over 262 billion floating-point operations per second.

Dataflow Architecture. Serial computers are limited to following a list of instructions step-by-step. Parallel computers free researchers to experiment, innovate and imagine radically different approaches to programming control. One of these computing strategies is the concept of dataflow, pioneered by Jack Dennis at MIT. Here data is sent to and from a processor as is needed or as solutions become available. A properly running dataflow machine should be able to maintain a constant stream of data flowing toward the solution. Dataflow machines are roughly like a railroad switchyard. All the processors, about 100 due to coarse-grained nature of most dataflow machines, are connected to this switchyard. Before computation, an instruction for each processor results. This new data is tagged with information about where it should go and how it should be used. The central yard switch will be able to read these "tags" and determine their appropriate routes. Nodes in a dataflow machine operate until all their necessary data arrives. Unused nodes remain idle. But by waiting for required data, the dataflow machine eliminates the danger of clashing with ongoing work of other processors and

creating a bottleneck. When the waited upon data arrives it is tagged and sent back through the network again. So the user only needs to specify instructions to be completed and allocate the right information to the processors once, and then let the flow of data reach a solution itself without having to be burdened with exact steps, procedures, r details of execution. As the dataflow grows, it has difficulty in dealing with its increased size and complexity. There are more communications, wiring, processors, and expenses. The cost becomes impractical which is why most dataflow machines are coarse-grained. If dataflow architecture is to become the architecture of choice for future computers it must be able to cope with the large database of real-world applications and artificial intelligence. Optical ComputersLight speed (the fastest anything currently known to us can go) is a constant one hundred eighty six thousand miles per second. Researchers the world over hope to harness this speed. Optical computers promise speeds much faster than today's computers can achieve. While silicon transistors have a projected speed limit of an operation ever 50 pico seconds, or a trillionth of a second, optical transistors might reach speeds of up to an operation ever femtosecond, or quadrillionth of a second (a million-billionth of a second). If successful these computers will have several advantages over today's silicon based computers 1) light will not interfere with itself like electrons do, 2) so computers will be able to process multiple signals at a time without losing the identity of any individual signal. This multiple processing capability is exactly what is required for parallel processing computers to be developed. The first priority for researches striving to develop optical computers is to develop devise much like a switch.

This transistor equivalent would be the basis for optical computers.

Conclusion The development of the fifth generation computer (at least as we see it now) would appear more dependent on the development and wide use of parallel processors and programs that utilize them. The attractive speeds offered by optical computers cannot be ignore though, so this avenue is likely to be the object of intense research, indeed it might be the sixth generation computer!