CHAPTER 1

Classic Machines: Technology, Implementation, and Economics

1.1 Introduction

The technology for implementation of computing systems has varied widely over the last one hundred years. Computers and their architecture at any point in time have been a consequence of the capabilities and limitations of the technology of their day. In this section, we illustrate this principle with a brief tour of computer history from a technological standpoint. It is essential to have a good understanding of technology, the trend of its advancement, and its limitations in order to have a good understanding of future developments in computers and their architecture.

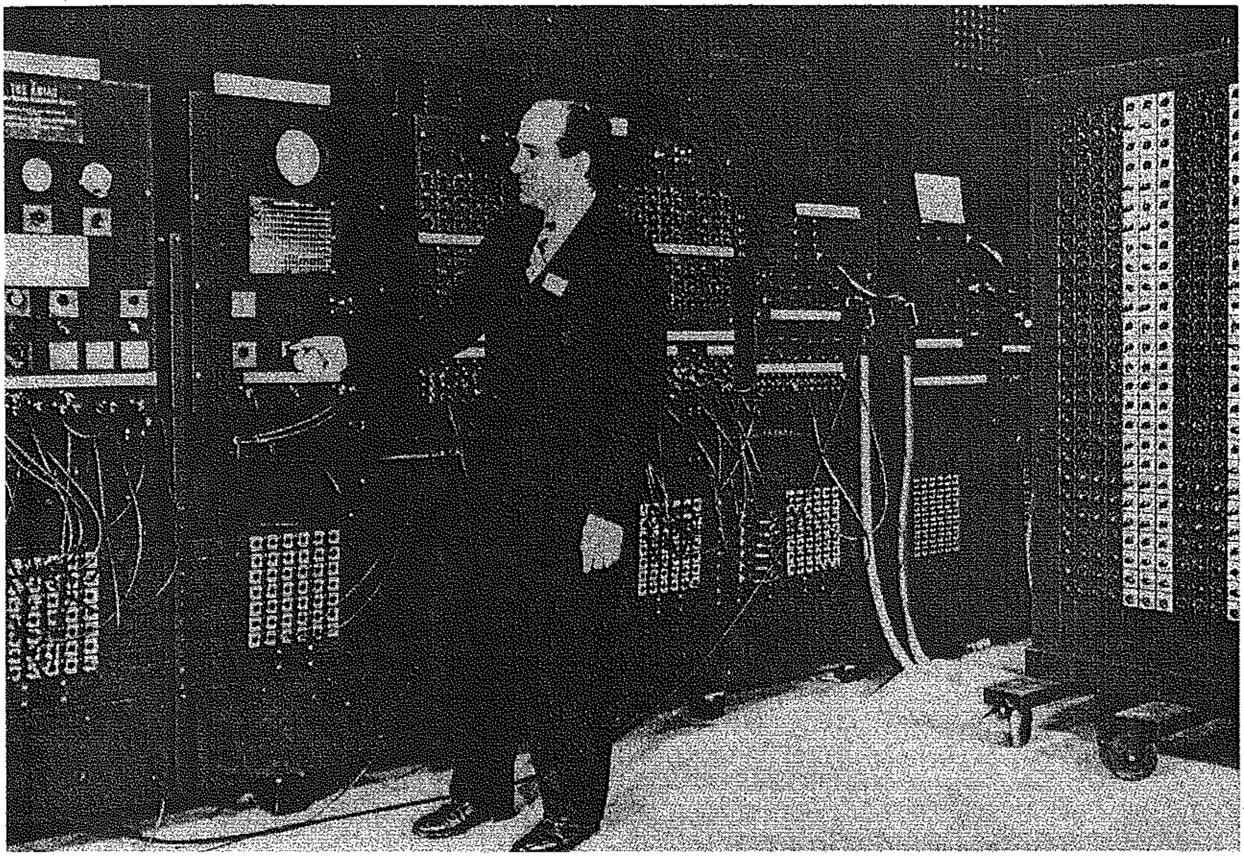
1.2 Early Computer Technology

During the first third of the twentieth century, technology was limited to mechanical and electromechanical calculators.[1] During the 1930s, several experimenters began building electronic calculating machines with vacuum tubes. From 1937 through 1942, Atanasoff and Berry at Iowa State University constructed a computer (the ABC) using binary circuits built with vacuum tubes and a capacitive drum memory. This machine had almost become fully operational when the demands of World War II caused its development to stop. The next significant vacuum-tube computer development was the ENIAC, which became operational in 1945 at the Moore School of Electrical Engineering of the University of Pennsylvania.

1.2.1 ENIAC Computer Technology

The ENIAC had over 18,000 vacuum tubes. Vacuum tubes require a heater element in order to displace electrons from their cathodes. This requires a significant amount of power. The heater element also has a tendency to burn out after a period of operation. Thus, based on the total power consumed by all the tubes and the expected time between tube failures, there is a maximum practical size of a vacuum tube computer. At the time of ENIAC, many vacuum tubes had lifetimes of only a few thousand hours. Worse yet, tube failures were spread over the lifetime of the tube as described by probability distributions. Given these handicaps, one would think it would be impossible to build a computer with 18,000 tubes given these characteristics.

[1] Information about these early calculators and computers up until 1965 can be found in the excellent reference of Williams [26].



ENIAC: Courtesy of the Hagley Museum and Library

Fig. 1.1. A close-up view of the ENIAC with Pres Eckert. The large panel of rotary switches mounted on wheels is one of the three constant tables. Programs were entered using the banks of switches below waist level.

J. Presper Eckert solved the tube reliability problem with a number of techniques:

- Tube failures were bimodal, with more failures near the start of operation as well as after many hours and less during the middle operating hours. In order to achieve better reliability, he used tubes that had already been burnt-in and had not suffered from "infant mortality." Also, the machine was not turned off at nights because power cycling was a source of increased failures.
- Components were run at only one half of their rated capacity, reducing the stress and degradation of the tube and significantly increasing their operating lifetime.
- Binary circuits were used, so tubes that were becoming weak or leaky would not cause an immediate failure, as they would in an analog computer.
- The machine was built from circuit modules whose failure was easy to ascertain and that could be quickly replaced with identical modules.

As a result of this excellent engineering, ENIAC was able to run for as long as a day or two without a failure! ENIAC was capable of computing 5,000 operations per second. It took up 2,400 cubic feet of circuitry, weighed 30 tons, and consumed 140 kW of power.

The architecture of ENIAC was like an electronic version of a collection of mechanical adding machines. It had no main memory as we think of it today but only had a collection of 20 ten-digit accumulators. The digits in the accumulators were stored in rings consisting of ten flip-flops, with a digit being signified by the corresponding flip-flop being set and all of the others in the ring being clear. A ten-digit accumulator required a total of 550 vacuum tubes! Programs were entered on plugboard panels and numerical constants could be entered into three function tables consisting of cabinets of rotary switches (see Fig. 1.1).

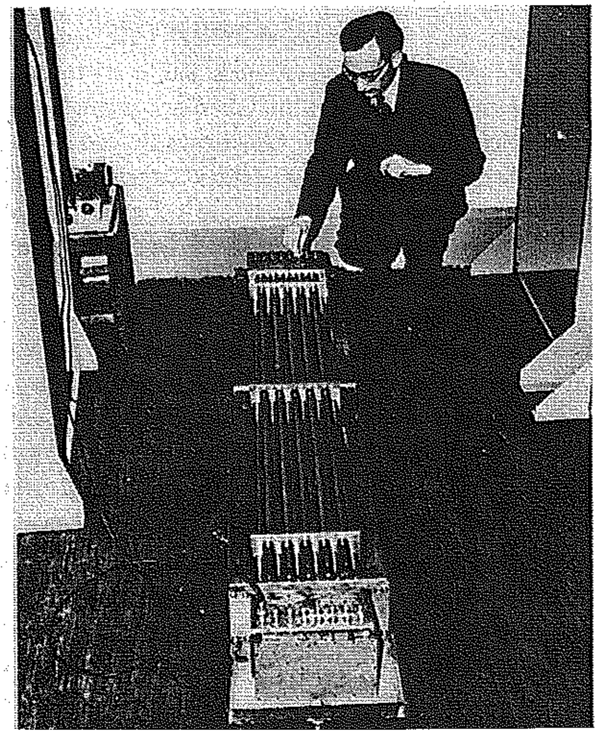
Toward the end of the ENIAC project, Eckert, John Mauchly, John von Neumann, and others working on the project developed the idea of the stored-program computer. This was published in a report titled, "First

Draft of a Report on the EDVAC" [24] in June 1945, with von Neumann listed as the only author. Later, in the summer of 1946, a special summer institute was held at the Moore School. This institute was widely attended by researchers from around the world and led to a burst of computer construction at many places.

As can be imagined, the primary limitation of the ENIAC was its lack of a memory (especially for implementing a stored program). Even if a more-efficient binary representation was used for the memory, tube flip-flop circuits would be too expensive and unreliable to be used to build a stored program computer. What was needed was a reliable, inexpensive, and fast technology for memory. In the late 1940s and early 1950s researchers experimented with several different technologies for implementing computer memories. These included electrostatic memories (such as the Williams tube) and various types of acoustic delay lines (such as mercury delay lines).

1.2.2 EDSAC Computer Technology

Maurice Wilkes attended the 1946 summer institute at the Moore School and immediately returned to Cambridge University and embarked on implementing a stored-program computer. First, Wilkes attacked the problem of providing a memory for the computer. By early 1947, he had designed and built the first working mercury acoustic delay memory (see Fig. 1.2). It consisted of 16 steel tubes, each capable of storing 32 words of 17 bits each (16 bits plus a sign) for a total of 512 words (roughly 1 Kbytes). Now that a suitable memory technology had been demonstrated, design and construction of the machine could proceed. Because the mercury acoustic delay lines were a bit-serial memory technology, this dictated a bit-serial machine organization.



Courtesy of The Computer Museum of Boston

Fig. 1.2. Maurice Wilkes looking at the first set of mercury delay line memories for EDSAC. This set contained 16 tubes, of which the top row of five can be seen [25].

Wilkes was more interested in having a machine he could program than pushing the state of the art in hardware technology. As a result of this conservative hardware design, he settled on a machine cycle time (500 kHz) that was slower by a factor of two than other machines being discussed at the time. Partly as a result of this conservative hardware design and early success with practical memory technology, Electronic Delay Storage Automatic Calculator (EDSAC) became the first large-scale operational stored program computer in May of 1949. This lead to the project leading the way in many software innovations. One crucial innovation from the early experience of the EDSAC project was a subroutine call instruction that saved the calling address for later use as a return address. At the time, this was called the Wheeler jump, after David J. Wheeler who invented it while he was working on the project as a graduate student. Another innovation from Wheeler was the world's first relocating program loader.

Because of the clock cycle time of the machine and the bit-serial machine organization, simple commands took 1.5 ms. However, a bigger limitation was the speed of the I/O devices. EDSAC used paper tape readers and teleprinters for I/O devices. The teleprinters could print 10 characters per second, whereas an optical paper tape reader could read 50 characters per second (each character consisting of only 6 bits)!

1.2.3 Manchester/Ferranti Mark I Computer Technology

The University of Manchester was an early leader in the implementation of stored-program computers. A simple testbed machine with only five instructions and 32 words of memory ran the first stored program on June 21, 1948. Because of this early success, they decided to build a full-scale computer. The design work for this machine was contracted out to the Ferranti Corporation. The machine used Williams tubes invented at Manchester by Frederic C. Williams and Tom Kilburn for electrostatic main memory. But because the Williams tube only stored 32 words of 32 bits, there was a need for a larger storage device.

Andrew D. Booth at Birkbeck College in London was an early innovator of storage devices. He developed thermal memories, mechanical memories, delay lines, and rotating magnetic memories. After early experiments with disk memories failed, he developed the first drum memory with help from his father, a mechanical engineer.

The Manchester team decided to build a storage hierarchy with Williams tubes and a magnetic drum. By using eight upgraded Williams tubes, they provided a total of 256 forty-bit words of main memory in the Ferranti Mark I of 1951. This was backed up by a drum containing 16,384 words of storage. The Mark I only took 0.03 seconds to transfer a subroutine from the drum to main memory. (In comparison, EDSAC could only read 1.5 characters from a paper tape in this amount of time.) The Mark I was also the first computer with index registers. Their early experience with transferring blocks of memory from primary to secondary storage and expertise in addressing led to the Manchester team's invention of virtual memory in the Atlas machine in the late 1950s.

The Mark I was still a bit-serial machine (as dictated by the bit-serial access of the Williams tubes), so its operation was again rather slow compared to its cycle time. A typical operation such as addition required 1.2 ms. In order to further improve computer performance, a large-scale reliable main memory with reasonable cost and parallel access was needed so that a bit-parallel computer could be built.

1.2.4 Whirlwind Computer Technology

The Whirlwind machine was designed and built in the late 1940s and early 1950s at MIT [18]. It was intended to be a bit-parallel machine and so much faster than previous machines (hence its name). In addition, it was designed for a much faster maximum 5-MHz clock frequency. Initially it was designed using a rank of sixteen Williams tubes, making for a 16-bit parallel memory. However, the Williams tube memories were a limitation in terms of operating speed, reliability, and cost.

To solve this problem, Jay W. Forrester and others developed magnetic core memories. When the electrostatic memory was replaced with a primitive core memory, the operating speed of Whirlwind doubled, and the maintenance time on the machine fell from four hours per day to two hours per week, and the mean time between memory failures rose from two hours to two weeks! Core memory technology was crucial in enabling Whirlwind to perform 50,000 operations per second in 1952 and to operate reliably for relatively long periods of time.

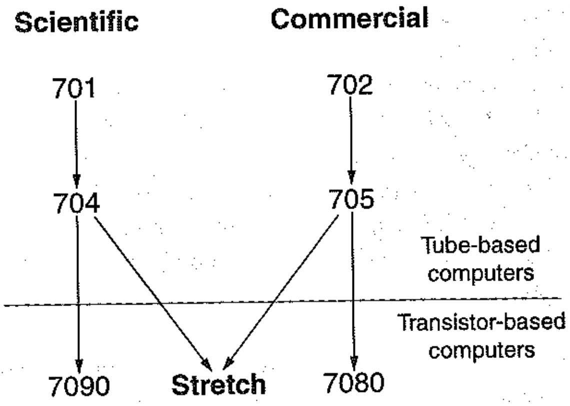


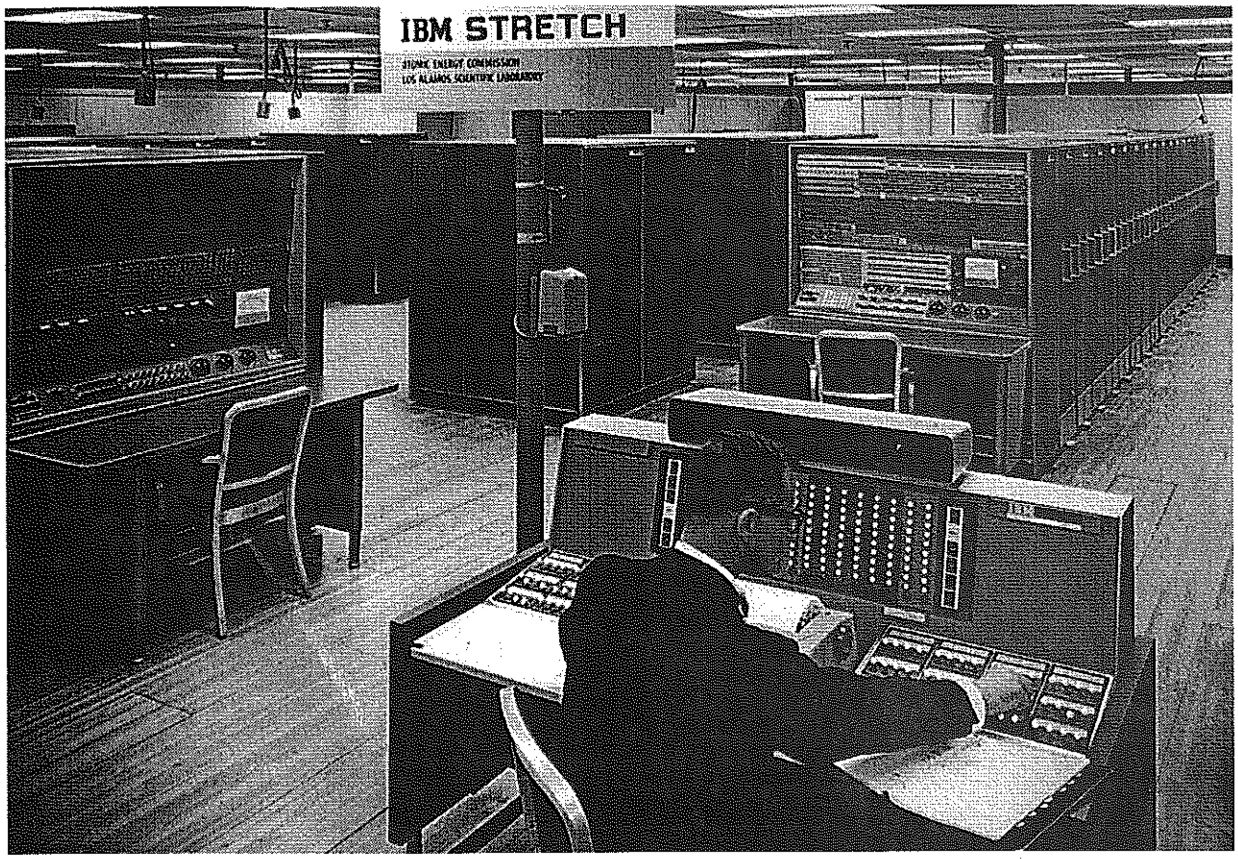
Fig. 1.3. The IBM Stretch's relations.

Now that a good memory technology had been found, the tube circuits used for arithmetic and control became the major limitation.

1.2.5 Germanium Transistor-Based Computers: The IBM Stretch

By 1955, quite a few manufacturers were building computers based on vacuum tubes and magnetic core memories. Because the number of components available for design of the CPU's was limited by the reliability and cost of vacuum tubes, computers tended to be quite specialized in their targeted market segments. For example, a machine targeted toward scientific users could not afford to have circuitry for packed-decimal arithmetic for accounting; similarly, machines targeted for commercial users could not afford to support floating point. Moreover, machines targeted at high-performance users tended to have larger word lengths, which lower performance and lower cost machines could not afford. Coupled with programming in assembly language, this led to considerable development costs for all of the different models as well as incompatible software bases.

IBM had a scientific and a commercial tube-based computer line, starting with the 701 and 702 models, respectively. These models were significantly redesigned and improved and sold as the 704 and 705. In 1955, IBM initiated a research project to produce a computer one hundred times faster than both the 704 or the 705 models and incorporating both scientific and commercial functionality. This machine was



Stretch (IBM 7030): Photograph courtesy of IBM

Fig. 1.4. An overall view of Stretch, which was first shipped to customers in 1961.

formally called the 7030, but most people knew it by its project name "Stretch" [2, 5, 6]. Its place in the IBM product line is shown in Fig. 1.3. A photograph of Stretch is shown in Fig. 1.4.

By switching to early germanium transistor technology, the circuitry in Stretch ran ten times faster. Also, by using a high-speed core memory, memory access became about six times faster than the core memory in the 704. However, the project goals of two orders of magnitude speed improvement went far beyond the speedups obtainable from new technologies. This would require many architectural innovations which would, in turn, require many more circuits than previous computers, but this was made possible by the increased reliability, smaller size, and lower cost of transistors compared with vacuum tubes. This resulted in a theme that is still being exploited today: using additional circuits provided by technology to exploit instruction-level parallelism.

With Stretch, the exploitation of instruction-level parallelism took many novel forms. First, the processing of instructions was pipelined (this was called "overlapped" or "lookahead" then), so that six instructions were in some phase of execution at any point in time. Second, in order to provide higher bandwidth to memory, memory was split into multiple banks. Because successive words were stored in different banks, new data could often be fetched every 0.2 μs even though the core memory had a 2-μs latency. Third, an autonomous transfer engine was provided to transfer data between the memory and peripheral devices (e.g., DMA or a primitive channel).

The designers realized that improving the performance of the CPU without improving the performance of the I/O devices would result in low overall system speedup. Thus, as part of the Stretch project the first disk drives with multiple read/write arms were developed. Their capacity (2 Mbytes) and transfer speed were far in excess of anything else available at the time and resulted in the abandonment of drums for secondary storage.

Although Stretch did not fully meet its goal of 100 times increase in performance (especially on commercial codes, which had byte-serial data storage—see Table 1.1), it resulted in many major advances in architecture and technology. Moreover, in combining both scientific and commercial capabilities, it was a precursor to the IBM System 360 architecture.

	Time in Microseconds		
Operation	**IBM 704**	**IBM 705**	**Stretch**
Floating add	84	–	1.0
Floating multiply	204	–	1.8
Floating divide	216	–	7.0
Five-digit add	–	119	3.5
Five-digit multiply	–	799	40.0
Five-digit divide	–	4828	65.0

Table 1.1. Key Operation Latencies in the IBM 704, 705, and Stretch

1.2.6 Large-Scale Planar Silicon Transistor Computers

> There is a striking chronological relationship between the appearance of the silicon planar transistor and the Control Data 6600. —J. E. Thornton, *Design of a Computer: The Control Data 6600* [21], p. 19. See also Fig. 1.5, reprinted from p. 21 of Thornton [21].

Stretch was designed with germanium transistors. By the early 1960s, planar (double-diffused) silicon transistors had been developed, with significantly improved reliability and performance characteristics over germanium transistors. "Planar" refers to the fact that those transistors were made by diffusing impurities into a flat wafer of silicon to form the base and emitter.

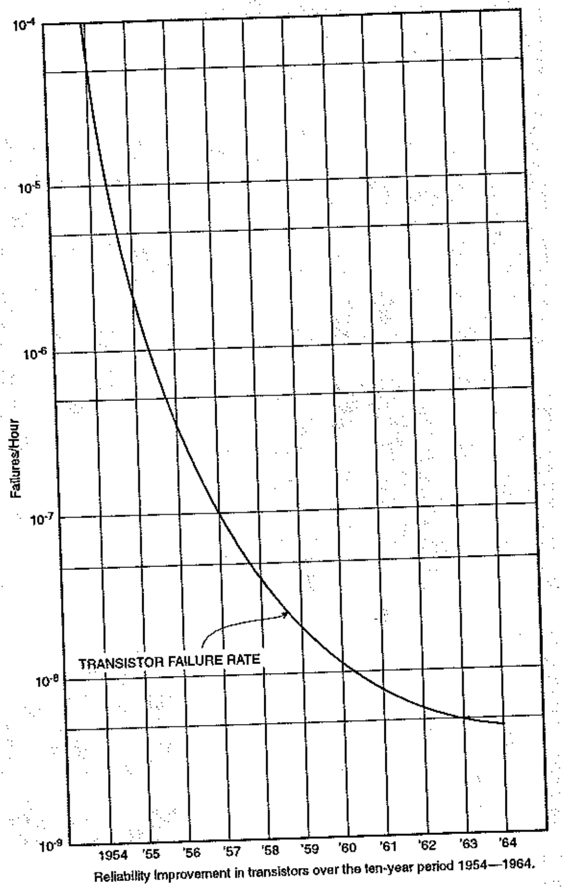
The main benefits of silicon planar transistors over germanium are:

- higher junction temperatures allowed (and hence higher reliability at lower temperatures)
- higher current and power levels available
- higher speed (by using a n-p-n configuration instead of p-n-p)
- lower cost

The combination of all these advantages was a key enabler of a generation of machines in the mid 1960s with greatly improved performance, reliability, and cost. These machines included the CDC-6600 and the IBM 360 family.

1.3 The Rise and Fall of Supercomputers

Until the late 1990s, high-end CPUs were a breed apart. Historically, high-end computers were made using bipolar transistors. This is in contrast to the MOS transistors used in microprocessors today. Bipolar transistors in the 1960s were relatively easy to make, because their active portion consisted of two p-n junctions that could be grown by diffusion processes. They were also more than ten times faster than early MOS transistors, which were not produced in practical numbers until the late 1960s.



Transistor reliability graph from page 21 of Design of a Computer: The CDC 6600 *by Jim Thornton.*

Fig. 1.5. Reliability improvement in transistors over the 10-year period from 1954 to 1964.

Because bipolar transistors were relatively fast, much of the delay in an 1960s- or 1970s-era high-end computer resulted from the time required to send signals through the wires of the computer. Some high-end computers (such as the CDC Cyber 205 series) used coaxial wiring that allowed transmission of signals at 70% of the speed of light. However, the Cray machines used slightly slower twisted-pair wiring but relied on denser packaging and circuitry to reduce the size of the machine, hence reducing the delays caused by communicating from one part of the machine to another.

There were many models of supercomputers built by different manufacturers. In this section, we focus primarily on computers designed by Seymour Cray for three reasons. First, most of them were the highest

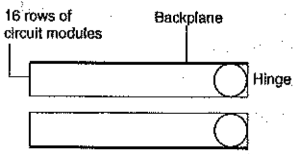


Fig. 1.6. A 1604 logic "book."

performance computers of their day. Second, they introduced a number of important architectural and implementation features. Third, by concentrating on a single lineage of computers, the underlying trends in technology become clearer.

The shape of machines built by Seymour Cray is an interesting study in itself. Seymour Cray was the champion of computer packaging and computational bandwidth. His first large scale machine, the CDC 1604, used a single "book" configuration. Two backplanes stuffed with sixteen rows of small circuit modules were flexibly mounted to the rest of the machine by hinges. This allowed the boards to be spread apart during construction or testing, like the pages in a young child's board book (see Fig. 1.6).

1.3.1 The CDC 6600

The CDC 6600 combined four books of four pages each with their spines facing the center of the machine. This formed an plus-shaped "+" center of the machine, as shown in Fig. 1.7. This configuration was later widely used by large mainframes and Japanese supercomputers. For photographs of the 6600 and other Cray machines, see the Web companion to this book for links to online photographs.

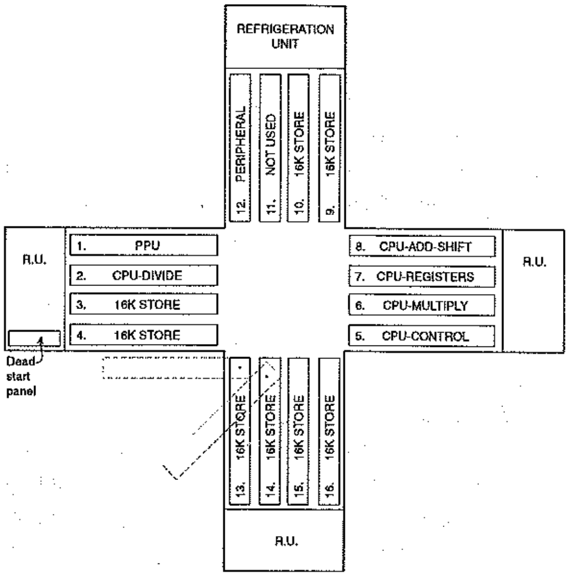


Figure 26 from page 35 of Thornton's Design of a Computer

Fig. 1.7. The classic 6600 plus-shaped machine configuration, widely used in later supercomputers by other makers. In the 6600, each page of the four books consisted of up to 756 "cordwood" logic modules.

1.3.2 The CDC 7600

A major advance in the 7600 over the 6600 was the introduction of fully pipelined units. This gave it a peak performance around seven times faster than the 6600, even though the clock was not quite four times faster (27.5 ns). It also reduced the need to keep the components in such close proximity to each other. Instead of large blocks of unpipelined logic which had to be close to each other, pipelining the functional units allowed each individual functional unit pipestage to be smaller physically and faster even though the whole pipelined functional unit was larger. Fig. 1.8 shows the machine was constructed in the shape of a large "C", with four panels on three sides and two panels and an opening on the fourth side.

The complete pipelining enabled the use of a large "C"-shaped chassis, but may have also required it. Because the machine was fully pipelined, the bandwidth required between units was greater than in the 6600. The restriction of edge-only access to each page in a book would provide much lower bandwidth than if the complete face of every page were accessible for

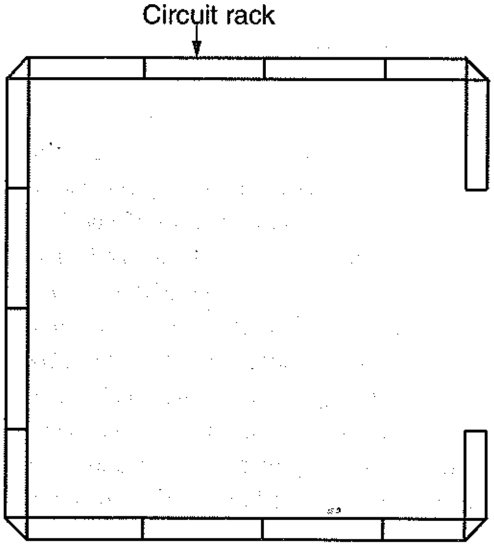


Fig. 1.8. The 7600 large "C"-shaped chasis configuration.

wiring. Whatever the case, the 7600 was certainly much easier to wire than the confined space between the spines of the four 6600 books. And as a final benefit, when seen from above the machine is the first letter of Seymour Cray's last name. Later machines at Control Data Corporation (CDC) under Jim Thornton elongated the classic plus-sign shape into a "t" shape.

Another major improvement in the 7600 over the 6600 was a significantly upgraded I/O system. Unlike the 6600, which implemented ten virtual peripheral processor units (PPUs) on a single physical PPU by time multiplexing, the 7600 had up to fifteen physical PPUs. This provided much higher performance I/O capabilities.

Once the functional units were completely pipelined, the limited number of registers provided in the 6600 architecture (of which the 7600 was backwardly compatible) would be a major limit. Moreover, the 7600 was limited to issuing one instruction per cycle, a factor of nine less peak-instruction bandwidth than that provided by the fully pipelined functional units. Both these limitations were overcome by the addition of vector registers and operations in the Cray-1.

The 7600 had a 100 × 100 Linpak performance of 7X the 6600. It first shipped in 1969, five years after the 6600. This was an annualized performance improvement of 48%. For comparison, note that the annual scaling of device count from Moore's Law is 59%, and the annual scaling in MOS device speed from scaling theory [7] is 26%. The maximum potential performance increase for MOS devices would be given by the product of the device count and the device speed growth rates.

1.3.3 The 8600 Program

After the 7600, Seymour Cray moved back to his home town in Wisconsin with a group of engineers to work on the 8600. The 8600 tried out tightly coupled multiprocessors (it had four processors accessing a common memory), while a team in Minnesota under Thornton tried out vectors. The 8600 implemented an architecture similar to the 7600 but it was a bold packaging departure. The machine was much smaller and more of a true circular shape than the 7600. But the most challenging aspect of the 8600 was that its boards were full of tiny discrete components (e.g., transistors covered only by a drop of epoxy 2.5 mm in diameter instead of the relatively large metal cans of the day) that were to be tightly interconnected in truly 3-dimensional (3-D) bricks. This provided many more wires in the vertical direction between parallel board faces than that available only on the board edges. This can be seen as the next logical step from the progression of the 6600 and 7600:

- The 6600 had modules in racks, with wiring between racks only at the rack edges.
- The 7600 had modules in racks, with wiring from the parallel face of the racks but only from the edges of the modules.
- The 8600 was to have wiring between the faces of the modules themselves.

The dense packaging would allow the machine to have a 8-ns cycle time.

Unfortunately, there were three problems with the 8600 program. First, the tight stacking of boards and components generated a lot of heat, which was difficult to remove from the board stack, resulting in overheating. Second, the 3-D connections between boards (and there were a lot of them) were very small, and they were difficult to test and assemble. This resulted in poor manufacturing yield and unreliability. But the third problem was that before these issues could be resolved, financial difficulties at CDC mandated the canceling of the 8600 project. As a result, Seymour Cray and a small team left CDC to form Cray Research, but they did get initial funding from CDC.

1.3.4 The Cray-1

At Cray research, the initial emphasis was to quickly develop a risk-free supercomputer to begin providing income for operations of the company. So, instead of attacking the relatively hard problems facing the 8600, a less-aggressive approach was called for. The result was the Cray-1 [19], which physically resembled the 8600 but was much larger, because it only used communication between module edges instead of module faces. This also made it easier to cool by using cold plates at the edge of each module (with racks placed side by side in 270° of a circle).

The Cray-1 was a more-conservative design than the 8600 in terms of clock rate and packaging. Also, the vector architecture was an extension of the full pipelining of the 7600, with the addition of more registers (eight 64-element vector registers plus the 64 register B and T banks behind the A and S registers). Vector registers were "architectural firsts" that led to much-improved performance.

The Cray-1 also contained a number of technology firsts. It used the first commercial emitter coupled logic (ECL) integrated circuits that compensated for changes in temperature, supply voltage, and process. Machines before the Cray-1 used bipolar transistors as a switch (like in transistor-transistor logic, or TTL), which resulted in the transistors going into saturation and making them relatively slow to switch. In contrast, the ECL circuits in the Cray-1 were built from high-speed differential amplifiers whose transistors never enter saturation. Besides being fast, because ECL gates are differential amplifiers, they produce both true and complement outputs. This allows them to provide more logical functions per gate stage than other circuit technologies. ECL gates also have high fan-outs and allow construction of OR gates merely by connecting emitter-follower outputs together (wire-ORs). The high gate functionality of ECL helped enable a logic design for the Cray-1 using only eight gate delays per cycle. Combined with the high gate speed of ECL, the Cray-1 achieved a cycle time much faster than any other computer of its time. Its 12.5-ns cycle time was not surpassed by commodity CMOS microprocessors until two decades after the first Cray-1 shipped in 1976. Even IBM mainframes using a variant of bipolar TTL logic and expensive multichip modules did not surpass the Cray-1's cycle time for fifteen years. One lesson here is that the speed of a "bipolar" machine greatly depends on the circuit technology.

From a business standpoint, the Cray-1 was relatively easy to manufacture and had a significant performance lead over all other supercomputers. This headroom allowed Cray research to create several direct derivatives of the Cray-1, the Cray-1S and the Cray-1M. Around sixty-five of the family were sold. This is in contrast to a more typical number of twenty to fifty units for an economically successful multimillion-dollar supercomputer and a much lower typical number for failures.

The Cray-1 clock cycle time was one third that of the 7600. With the addition of vectors, it achieved a 100 x 100 Linpak performance of 27 MFLOPS, which was about eight times faster than the 7600. Shipping seven years after the 7600, this represented a performance growth rate of about 35% per year. This was below the 48% growth rate from the 6600 to the 7600 and significantly below MOS microprocessor performance growth rates of close to 60% [9].

1.3.5 The Cray-2

Now that the foundation of the company was assured, Seymour Cray returned to more of a research style (befitting the name of the company). Furthermore, a line of derivative machines in improved technologies (the letter series X-MP, Y-MP, etc., designed under Steve Chen) continued to support the company.[2] The next machine Seymour Cray would build was remarkably like an updated version of the 8600: the Cray-2.

Initially, to get a significant clock speed increase over the Cray-1, Seymour Cray wanted to use gallium arsenide in the Cray-2. This direction towards GaAs can be seen at the end of the Cray-1 technology paper [11]. But gallium arsenide technology was not mature enough yet to be useful, so after some delay the Cray-2 ended up being built out of 16-gate bipolar ECL gate arrays. The Cray-2 was a four-way multiprocessor (like the 8600 but with the addition of vectors), and it solved the 3-D circuit stack problem. A more reliable board-interconnection technology was found that allowed many more connections between the parallel faces of boards than from their edges. The heat problem was solved by immersing the whole machine in Fluorinert. Although Fluorinert was originally developed as an artificial blood substitute, it was not used medically. However, being a liquid, it had better heat capacity than air and was electrically and materially inert. This Cray-2 had a 4.1-ns clock cycle time (3X faster than the Cray-1 but only two times faster than the proposed 8600) and up to four processors. The vector architecture was almost the same as the Cray-1 but it did not allow chaining. The Cray-2 used DRAMs for main memory (in contrast to SRAMs in the Cray-1 and X-MP), so it had much longer memory latencies and had to be highly banked (128-way) [10]. As a result of the slower memory, the performance per processor was skewed toward long vector codes and did not scale by the clock frequency. The machine was a four-processor multiprocessor. This improved performance on parallel applications but did not increase uniprocessor performance. Combined with the project delays and the implementation of more-flexible chaining in the letter series along with multiple memory ports (the Cray-1 and Cray-2 only had one memory port per processor), this resulted in less success for the Cray-2. But it still sold thirty copies, mainly to customers needing its larger main memory capacity.

Now supercomputer development was falling significantly behind the volume MOS technology scaling curve driven by lithography. The Cray-2 came out in 1985, nine years after the Cray-1, but its clock cycle time was only 3X faster. This was only a compound annual growth rate of 13% per year.

[2] Over two hundred X-MPs were sold!

1.3.6 The Cray-3 and the Death of Traditional Supercomputers

After the Cray-2, Seymour Cray resumed work on GaAs. His next machine would remove the last vestige of two-dimensionality from his designs: the signature "C" shape. The "C" is two-dimensional (2-D) in that signals traveling from one part of the "C" to another must travel in a (curved) 2-D space of board stacks and stacked racks. The Cray-3 would be a cubic shape, with a base that was 1 foot on each side. This was an extremely aggressive undertaking, for reasons of difficulty of testing and manufacturing such a machine. Internal parts of the machine would not be accessible for testing, and the small size of the interconnections would require robotic assembly using tiny wires and precision laser welding.

Again, Seymour's parent company did not have the resources to continue development of the existing profitable line of computers and Seymour's risky new venture. This led Seymour Cray to spin off his project again as part of a new company in 1989. Because the name Cray Research was already in use, Seymour Cray called his new company Cray Computer. This is ironic since Cray Research was in the business of producing computers and Cray Computer was more of a research venture.

The Cray-3 was announced in 1993. Although the machine was supposed to have sixteen processors, only one processor was ever delivered for evaluation. It had a 2-ns cycle time. This was eight years after the Cray-2 but was only faster by a factor of two. The uniprocessor performance growth had decreased to an annual rate of only 9%, but at the same time the technology was getting much more difficult to manufacture and test. This was not economically sustainable as a business, even compared with the more traditional letter line of computers from Cray Research. The Cray Research T90, announced in 1994, used bipolar standard cells and had a clock cycle time of 2.2 ns. But it also had dual vector pipes, support for chaining, a scalar cache, and had up to thirty-two processors. This gave it an aggregate peak performance 8X more than a full Cray-3. It may be that a 3-D computer will never be economically or even technologically justifiable.

By the time of the Cray-3 announcement in 1993, scaling of volume CMOS process technology had resulted in clock speeds of up to 200 MHz, as evidenced by the DEC Alpha 21064 microprocessor [14]. This chip was limited in performance by its I/O interface, which only consisted of pins along the edge of the chip. More recently, volume microprocessor chips have adopted bump technology, which allows interconnections over the entire face of a die instead of just at the edges. Thus, in a sense, they are following in Seymour Cray's footsteps technologically but in a fashion that must remain profitable.

1.4 The Rise of MOS Microprocessor-Based Computers

The primary reason why non-saturating bipolar technologies such as ECL were so much faster than MOS in the 1970s was that they had very narrow base widths that could be carefully controlled by diffusion (and later ion implantation) to be much smaller than the lithographic line widths of their day. For example, the transistors used in the IBM 360/91 in 1967 had a base width of 0.5 μm [12], whereas production lithographic feature sizes were around 20 μm [9]. However, there are certain limits to how narrow bipolar bases can be made, and these limits are close to what MOS transistors are achieving with advanced lithography today. For example, recent high-performance bipolar gate-array processes have base widths of around 0.08 μm, giving a compound annual rate of change of only 6.3% per year since 1967.[3] Thus, as Moore's Law [16] advanced, the performance gap between nonsaturating bipolar technologies and MOS technologies closed.

Differences in production volumes of different technologies leads to different levels of sustainable technology investment. The bipolar technology used in supercomputers was a very high-end technology, and shipped in relatively low volumes. This meant that much less money was available for investment in bipolar technology. As a result, both lithographic and device development of bipolar technology started lagging that of MOS technology by the late 1980s. This reduced even further the relative performance advantage of bipolar technologies versus volume MOS technologies, leading to even smaller relative volumes. This is obviously a bad spiral to be on. As a result, by 1998, supercomputers and mainframes being built in the United States use multiple copies of microprocessors made with volume CMOS technology.

If a computer ships in low volumes, the design cost of the machine must be amortized over a small number

[3] Obviously, there are many other important parameters that determine the performance of transistors that we have ignored for the purpose of this discussion.

of machines. This also limits the resources that are economically available for their design. As a result, supercomputers and mainframes have always been built with standardized building blocks. An extreme example of that was the Cray-1, which was built using only four different integrated circuits. How much more performance would have been achievable if full-custom techniques were used instead? As an example, multiplexors built solely from AND/NAND gates require two levels of logic, whereas in a custom multiplexor gate, the delay from the data inputs to the data outputs is reduced to one gate delay. Similarly, if a logic function really requires six inputs, a whole extra logic level will be required if only 5/4 input gates are available.

In contrast, the initial microprocessors were developed with volume markets in mind. They also had very meager device budgets. This led to a full-custom approach and innovative circuit designs that have persisted until this day in high-performance MOS microprocessors. The combination of full-custom design with aggressive circuit design probably accounts for a factor of three in system performance over a processor built from a large MOS standard-cell library.

Many young engineers believe the "best product" will win in the marketplace. Unfortunately, the marketplace is much more complicated than this. One example of this is microprocessor instruction-set architectures. One might expect the cleanest, most powerful architecture with the highest performance implementations to be the most successful. (An example with these characteristics is Compaq's Alpha, which currently has a market share of less than 0.1%.) Instead, one would expect the arguably most irregular and lowest performance instruction-set architecture (with such "bad" features as only a few registers, nonorthogonality, a single condition code register, and a segmented address space, just to name a few) not to be very successful at all. However, evolution in instruction-set architecture does not involve survival of the fittest in the biological sense, because the strength of an ISA greatly depends on the size of the installed software base. The 8008 was the first high-performance microprocessor architecture (compared to the 4004, the first microprocessor architecture). Because it was the first, it was the most successful when it started. In order to preserve its customer base, the most successful architecture at a given point in time will be extended with new features necessitated by the current technology. This will ensure that the commercially most successful architecture continues to be successful. After many generations of adding new features onto an existing instruction-set architecture that was not designed with expandability in mind, you can imagine the creature that results. Thus, in instruction-set architecture evolution, the surviving architecture will therefore be among the "least fit." This does not preclude its implementations from being among the "most fit," however, as we shall explain in the last chapter of this reader.

1.5 Discussion of Included Papers

In this section we introduce papers that describe the technology, implementation, and economics of classic machines.

1.5.1 Amdahl, Blaauw, and Brooks's "Architecture of the IBM System/360" [1]

In November 1961, IBM formed a group known as the SPREAD committee, which was chartered to set IBM's design goals for the next decade and to create an engineering and marketing plan to unify the efforts of the three IBM computer divisions. At the end of December 1961, the report of the group was released. It advocated production of a family of compatible machines ranging from the smallest commercial machines of the day to scientific computers even more powerful than Stretch. After much engineering effort, on April 7, 1964 IBM announced the 360 architecture with over 150 new products, services, and devices [17]. The family name 360 was chosen by IBM to say that it excelled at all "360 degrees of data processing."

At the same time as the announcement, an excellent paper giving the design rationale for the 360 architecture appeared in IBM's *Journal of Research and Development* [1]. Not only does it describe the architecture in detail, but it describes the tradeoffs that were evaluated leading up to the architecture. It is worthy of study by all computer architects and is included in this reader.

1.5.2 Thornton's "Parallel Operation in the Control Data 6600" [22]

> Last week Control Data . . . announced the 6600 system. I understand that in the laboratory developing the system there are only 34 people including the janitor. Of these, 14 are engineers and 4 are programmers. . . . Contrasting this modest effort with our vast development activities, I fail to understand why we have lost our industry leadership position by letting someone