Historical perspectives on computers—Components

by J. H. POMERENE

IBM Corporation
Armonk, New York

INTRODUCTION

The technological base for computers was laid just prior to and during World War II. Military requirements led to a general upgrading of components. Television and radar pushed the development of high performance vacuum tubes, particularly the twin triode. Radar shifted the attention of engineers from frequency to time and timing. The microsecond became familiar.

Against this background a singular project began in 1943: An electronic calculator called ENIAC, planned to use almost 20,000 vacuum tubes. This was an unprecendented number, three orders of magnitude greater than state-of-the-art electronic products and ten times the size of anything else being considered. There was no assurance from past experience that the machine could ever work. Some observers predicted a tube failure every few seconds.

As it turned out there were only two or three tube failures per week and ENIAC was very productive. Actual experience was five orders of magnitude better than the worst predictions. The difference was due to design rules which minimized the probable causes of failure and to a less rigorous principle which says that reliability is often better than careful calculations show. Whatever—ENIAC established that large vacuum tube systems would work.

The ENIAC was primarily intended to compute ballistic tables. It was designed by analogy to electro-mechanical calculators and like them used decimal arithmetic and handled the digits of a number in parallel. Operations were timed by a 100 KHZ clock; addition time was 200 microseconds and multiplication required 2.8 milliseconds. Internal memory was very limited and consisted of 20 numbers of ten decimal digits held in vacuum tube accumulators. Instructions were not stored in memory; the machine was set up for each problem by means of pluggable wiring.

The speed of ENIAC attracted interest in solving problems outside of ballistics—hydrodynamics calculations, for example. Some of this was done but the limited memory and the manual labor of re-programming were severe limitations. It was recognized that a considerably larger memory ought to be provided for the basic data of the computation and that the same memory could also store the program to be followed. A useful capacity was estimated to be 4,000 words (i.e., either numbers or instructions). From this point forward memory technology was to dominate system design.[1,2]

EARLY MEMORIES

The choices for a 4,000-word memory in 1945 were not many. A word size on the order of 40 bits was required so that 160,000 total bits would be needed. Although a memory could in principle be made from the logic technology (i.e., vacuum tubes) this would require perhaps four tubes per bit or a total of 640,000—rather too many. It made better engineering sense to look for bulk memory effects, ways to store a large number of bits in one physical device.

One possibility was to launch pulses representing bits along a path having a large propagation delay, receive the pulses at the far end then re-amplify them and re-launch them. The memory would be the number of pulses in transit through the delay. Acoustic delay lines had been developed for radar with delays on the order of a millisecond and capable of transmitting one microsecond pulses, with consequent memory capacities on the order of 1000 bits per line. This kind of memory was used in one immediate successor to the ENIAC which was called the EDVAC. Since the memory was inherently serial EDVAC was organized on a serial basis, that is the bits of a number were handled in time succession rather than all at the same time.

Another possibility was very obvious in principle: the iconoscope which could store and retrieve a television picture with over 200,000 resolvable elements. For

rigorous digital storage this capacity would have to be much derated, though by how much was not known. The basic attraction of the approach was that an electron beam could be used to lay down a pattern of bits on a suitable surface at very high densities and that any of these bits could be selected for retrieval by the same beam. This random, rather than serial, accessibility to bits permitted (though did not require) parallel handling of a number. A second immediate successor to the ENIAC was built, using this parallel approach, at the Institute for Advanced Study.

Although suggesting the approach the iconoscope itself was not used for electrostatic memory. Several alternates were suggested or tried but the most widely used system was based on a development by F. C. Williams using ordinary cathode ray tubes. This scheme was based on small area differences in secondary electron redistribution and may be one of the few utilizations of a third-order effect.

It was possible to store 1024 bits per tube without getting unacceptable coupling between adjacent charge distributions. The worst case of this coupling, picturesquely called spill, occurred when one bit was read many times more than its immediate neighbors. All electrostatic memories required some explicit or implicit measures to control spill.

Delay line and electrostatic memories provided capacities from 512 to 2048 words, short of the estimated 4000 but enough to get computing well started.[3,4]

EARLY LOGIC AND PACKAGING

Basic considerations

It was clear that at least the following points had to be considered in designing a digital computer:

(a) Large numbers of tubes and other components would be used, all of which would have to work very reliably.
(b) Electrical signals representing the numbers would have to be kept within operable limits throughout the machine and over long periods of time.
(c) Since all possible bit patterns could occur within the machine the DC component of signals would have to be taken into account.

These overall considerations entered into many of the more specific decisions.

Derating and tolerances

The major components to be used were vacuum tubes, composition resistors, and in some machines pulse transformers and crystal diodes also. Tubes could fail catastrophically by heater failure or internal shorts and gradually through loss of cathode emission. Resistors could drift away from initial values. Diodes probably did not wear out but could be destroyed by overload of very short duration.

Heater failures were controlled in the ENIAC by never turning the heater off, so that thermal cycling did not occur. IAS designers felt that most heater damage came from the shock of rapid heating or cooling and provided for gradual turn on and off. Both techniques worked very well. Internal shorts were minimized by rigid pretesting including vibration, choice of tube types, and careful control of heater—cathode potentials. Emission loss was allowed for by designing circuits which would still work at half the nominal operating current. Power dissipation was derated usually to 50 percent and sometimes more.

Resistor deterioration was known to be accelerated by high power dissipation so all were derated to half power. Initial values were held to a 5 percent tolerance, or better in critical circuits, but all circuits were designed to be operable with all resistors off nominal by 10 percent in the worst direction.

Diodes were protected by designing circuits in which overload conditions could not occur or at least required unlikely failures in other components.

Vacuum tube choices

The basic logical "AND" operation was provided in two ways. With a multigrid tube the control grid and the suppressor grid were used. With triodes two or more were used together either with common plates or common cathodes. The triode circuit was easily extended to more than two inputs and was widely used. It was common to employ the "long tailed pair" in which the cathodes were connected together and returned to a negative voltage large compared to the grid-cathode voltage, typically—150 or—300 volts. In this arrangement the tube current was largely determined by the cathode resistor and the return voltage; an important technique to minimize effects of tube deterioration.

Unfortunately the signal out of a vacuum tube circuit is always at an appreciably more positive level than its input and must be translated back down to be used as an input to the next circuit. The capacitor coupling common in communication circuits could not in general be used because it blocked the DC component of the signal. DC translation required a resistor divider returned to a large enough negative voltage to minimize signal attenuation.

This translation network was a major limitation on speed. In order to have an adequate output signal under worst case tolerance it was necessary to have a large nominal signal swing at the plate. Since tube current was limited by derating, the network impedance had to be pushed up and circuit delay along with it. As crystal diodes became available with ratings compatible with tube signal swings (e.g., 30 volts) this problem led most designers to use tubes mainly for re-amplification and powering and to do the logical operations with diodes.

Pulses vs. DC coupling

One way to handle the DC component of signals was to use direct coupling, as described above. This allows the system to run at any speed up to its maximum which need not be known in advance of design. Another way was to represent information bits by pulses and restore the DC component when necessary by clamping diodes. Adequate DC restoration required some advance decisions affecting final speed, such as a standard pulse width. This difference afforded some interesting debate but final speeds turned out about the same.

Pulse logic systems were timed with explicit clock signals. Though most direct coupled systems were also clocked the timing latitude available permitted an asynchronous mode of operation. This mode had the advantage, in principle, of being insensitive to timing changes—as tubes degraded the machine would still run albeit more slowly. One form of asynchronism was used in the IAS machine. Timing was determined by circuits analogous to those being controlled, and affected in the same way by supply voltages and control wave forms. This compensated for several kinds of deterioration which could occur but not all.

A more complete concept of asynchronism was self-timing logic. In one version a signal would be propagated separately in both true and complement form, the arrival of one or another at the end of the chain would signal completion. The Philco S-2000 embodied logic of this kind.

Wiring and connections

Wiring presented no problems unique to computers. Layout tended to be generally planar but all three dimensions were used for wire routing. Wiring impedance was largely uncontrolled, except for being kept as high as possible (i.e., a thin wire in free space) in some direct coupled machines. Even so, circuit impedances were higher yet and capacitive loading was a problem. Pulse logic machines represented a low enough source impedance to largely avoid capacitance problems.

The tubes and circuitry associated with a logical grouping, such as a register position, were generally packaged together as a plug-in unit. Layout within the unit was three-dimensional and followed signal flow where possible to minimize connection lengths. Signals going outside the unit were driven by cathode followers.

Two non-commercial machines, the SEAC and the IAS, probably represented the packaging extremes. Signals in the SEAC were driven at low impedance from pulse transformers and wiring could be any reasonable length along any route. As a result a convenient rack and panel construction was used, logic and crystal diodes on the outside for accessibility and hot tubes and resistors on the inside. Wiring going any length ran in longitudinal trays.

The IAS machine was direct coupled and hence high impedance. To minimize capacitance loading the physical layout followed the logic flow very closely so that all signal wires were very short. Chassis were curved so that all intra-chassis wiring could be point-to-point away from the chassis (like chords of a circle).

Power and cooling

The heater of a computer tube required between 0.45 and 0.9 amperes, or 450 to 900 amperes per 1000 tubes. Supplying and distributing such large currents presented no basic difficulty but heaters were always a nuisance and certainly not less so in large numbers. DC power was more of a problem. Loads aggregating to 30 amperes at 300 volts were common and similarly for other voltages. Commercial supplies of this size were not initially available, so early groups had to plan their own. In some cases large storage battery banks were floated across a DC generator. This had some advantage during construction, when voltage levels and loads were subject to change, but it was not operationally convenient. Subsequently, very satisfactory commercial supplies were produced. These used thyratrons in a 3-phase full wave circuit and had filtering capacitance of nearly a farad. Regulation was easily held to a few percent.

Cooling was done with forced chilled air. The only real difficulty, at least for the earlier machines, was learning how to operate air conditioning in the winter.

FERRITE CORE MEMORIES

Origin

As computers became operational and were put into use attention turned to increasing the memory capacity. For the serial delay line machines an auxiliary magnetic drum memory could offer considerably increased capac-

ity at nearly the speed of the delay line memory. For the faster parallel electrostatic machines the same kind of drum was too slow to be other than a secondary memory.

At about this time (1951) the ferrite core memory was developed at MIT and at RCA. The MIT memory replaced the electrostatic system on Whirlwind and provided 2048 words; the RCA memory was a 10,000-bit plane. At the outset the ferrite memory offered speeds equal to or better than electrostatic and capacities appreciably greater. Further, it seemed likely that speed and capacity could be increased by an order of magnitude with further development.

General characteristics

By earlier standards the ferrite memory was unlikely. Instead of storing 1000 bits in one delay line or 1024 bits in one cathode ray tube it required a ferrite core laboriously threaded with 3 or 4 wires for each and every bit. The assembly labor for one early core plane was 240 hours. Probably only a firm conviction that production could be automated gave courage to proceed.

Apart from the labor of core-stringing the prognosis for ferrite looked very good. The magnetic properties were a bulk effect susceptible to tight control and having no likely time deterioration. Although the drive circuits were fairly expensive it was expected that much larger arrays could be handled. Since the drive cost for a $n \times n$ array was proportional to n and the capacity was n^2 the drive cost per bit would be proportional to $1/n$. This is a strong economy of scale.

Manufacturing improvements

The investment in manufacturing automation depends on the expected volume of production. The great success of the ferrite memory came from the expectation by manufacturers that the volume would be very large. The first impetus came at MIT shortly after the development of the ferrite memory. In planning for the SAGE air defense system MIT worked out some basic techniques for core plane fabrication. IBM continued the work, both for SAGE and its own commercial production. IBM made a large and continuing investment in automated fabrication and also in automated pretesting of cores. The story of this has been well told elsewhere. The outcome was that memory changed from something rather special and difficult to something that was commonplace, easy to use, and could be as large as need and purse allowed.[5]

Speed improvements

The first ferrite memories used cores of a size that could be easily seen and through which wires could obviously be pushed. The resulting speeds were equal to or better than electrostatic memories and quite compatible with vacuum tube computers. For example, the IBM 701 with a 12 microsecond electrostatic memory was replaced by an improved IBM 704 with a 12 microsecond core memory, both matching the cycle of the arithmetic logic. However, the switch from vacuum tubes to transistors made arithmetic logic cycles of 0.3 microseconds seem possible: an improvement of 40 to 1. Note that this came from a change of kind in logic technology, not just degree. (Ferrite memories could never produce a change of degree to match this change of kind.)

Increasing the speed of a ferrite memory involves a number of factors but in any case the core size must go down. From early cores about the size of an aspirin pill the size went down toward the almost invisible. Problems of handling, threading and testing were solved on the way down but a barrier of sorts was reached at a cycle time of 0.5-1.0 microseconds. These were produced in quantity but the next step, seen as 250 nanoseconds, would involve a massive tooling effort to be practical. Semiconductor memory technology had meanwhile moved to where it could predictably offer higher speed and lower cost. The next ferrite step was not implemented.

THE ROLE OF PROGRAMMING

It took special skill, motivation, and patience to program the early machines and it was not obvious that use of computers would ever spread beyond a limited number of places where such expertise could beassembled. Attempts were soon made to use the computer itself to handle some of the labor of programming but the small capacity of the early memories limited what could be done. The larger ferrite memories removed this limitation and effective programming aids began to appear. Among these were programming languages which in effect re-defined the hardware computer into a new computing system. This new system could be programmed by a user in terms familiar to his discipline, he did not have to learn or deal with the intricacies of the hardware system.

One of the first languages was FORTRAN, which provided a computing system particularly easy for scientists and engineers to learn and use. As a result the entire technical community became actual or potential programmers and usage of computers skyrocketed.

This expansion of computing services to ever-widening circles of users became the driving force for the growth of the industry.

DISCRETE TRANSISTORS

The invention of the transistor attracted the immediate attention of computer designers. They were smaller and effectively faster than tubes and required no heater power. Work on transistor computers began as soon as enough transistors could be obtained and with the explosive growth of computing already apparent there was an urgency which produced rapid improvement. In short order it became possible to design machines which would be ten times faster than existing vacuum tube systems. The pace of improvement was still continuing, however, and even higher speeds seemed possible.

The STRETCH project of IBM is a well-documented example of this point in computer history. STRETCH designers set a goal to be one hundred times faster than the IBM 704, a goal to be reached by pushing hard on both component technology and logical machine organization. Though STRETCH did not meet the goal in all respects, the effort did result in a significant upgrading in all areas of computer technology.[6]

Circuits and logic

Not only did transistors have no heaters, they were also available in two complementary forms working on opposite voltages. The problem of signal voltage translation which had been such a significant limitation in direct coupled vacuum tube circuits could be handled very nicely by alternating the two kinds of transistors. The DC signal shift of one kind was compensated by an opposite shift in the other. All transistor machines used direct coupling.

The problem of saturation in transistors was avoided by the development of a circuit in which the saturation condition would not occur. This was called a current mode circuit and it was the transistor analog of the "long-tailed pair." This circuit was used in STRETCH and it is still widely used today where speed is important. Used with drift transistors it provided a circuit family with an average delay less than 20 nanoseconds.

The transistors, resistors, and diodes comprising a basic logic circuit were mounted on a small card with printed wiring interconnections. In many cases larger cards were also used to provide larger functional groupings having recurrent use. These larger cards carried on the order of 20 transistors. As with vacuum tubes these circuits were designed on a "worst-case" basis.

Memory limitations

Early core memories were driven with vacuum tube circuits. As part of the work on transistor machines development of a fast core memory with all transistor drive circuits was begun. This was a bold effort because it sought a threefold increase in speed over the best memory then available while accepting the handicap (at that time) of not using tube drivers. The goal was 2.0 microseconds and in fact a cycle of 2.18 microseconds was achieved. This was probably the last memory specified on a rigorous worst-case basis. In a way reminiscent of the spill problem in electrostatic memories it was observed that repeated accessing of the same location at maximum memory rate would result in heating the affected cores beyond the Curie point and result in loss of information. For this reason the array was cooled in oil. Somewhat later, on the realization that the worst case was exceedingly improbable, air cooling was substituted.[7]

Ambitious though it was, the two microsecond cycle fell far short of matching transistor speeds. In STRETCH, for example, the logic cycle was 300 nanoseconds, making the memory cycle seven times greater. In order to offset the speed imbalance the concept of lookahead was introduced. The memory would be kept as busy as possible supplying the next few instructions and operands in anticipation of their use. Unfortunately the critical importance of the branch instruction was not fully recognized. At a branch the program may take one of two paths and if the lookahead had gone down the wrong path considerable unwinding was necessary. This problem proved to be quite fundamental and had a strong effect on high performance machine organization.

Wiring and connections

The first level of wiring was now handled by printed wiring on the circuit cards. These cards then plugged into sockets on the back panel. Wire wrap was used rather than soldering, a choice which facilitated the widespread use of automated back panel wiring. Coaxial cable was used for critical leads and ordinary wire for the rest. This wire, though not really controlled in impedance, tended to be about 150 ohms in the back panel environment.

Although transistors took much less space than tubes they were also used more lavishly. As a result most computers were still too large to fit in one conveniently

sized frame. Cabling between frames was conventional but on large machines like STRETCH it could also be described as monumental. Again, despite the much lower power consumption of transistors, their large numbers resulted in power and cooling requirements not much different from vacuum tube machines.

LARGER MEMORIES

Computers had grown along two diverging lines, scientific and commercial, and it was becoming apparent that this divergence was serving neither area well. Almost every installation really needed to do both kinds of work. A unification was needed but it could not be achieved without fundamental machine instruction and instruction format changes which could invalidate a majority of existing programs. Allowing the divergence to continue, however, would only let the problem grow larger. In announcing System/360 IBM opted for unification and set about to do what was technically possible to ease the transition.

Actually a more pedestrian force than the logical necessity of unification would probably have forced the same outcome. The 704-709-7090 had an address field of 15 bits, enough to address some 32,000 words of memory. Though apparently an ample allowance when the 704 was designed it had become evident that much more memory was needed to support the kind of programming service then in demand. However, the simple change of the address field to provide more bits would have caused most of the disruptive effects of the more comprehensive changes of System/360.

From our component-oriented standpoint the net effect was that much larger memory could be addressed (up to 16 million bytes or approximately 4 million words.) Once again this had a direct effect on system software, making possible a comprehensive operating system which in turn increased the utility of computers and stimulated further growth of the field.

Computer designers from the outset thought of the speed of light as a mere 1000 feet per microsecond. This was not a limitation in early machines but designers felt that it would become one. When the nanosecond speeds became possible the speed of light was then regarded as one foot per nanosecond and the limitation was more tangible. That which everyone knew was coming was suddenly at hand. The situation with integrated circuits is quite similar and also relevant to the speed question. As one looked at the physics underlying the semiconductor art one realized that there was no near-term limit to how small transistors, and their interconnections, could be made. The piece of silicon which once provided a single transistor could be made to hold a complete circuit of many transistors—an integrated circuit. The scale of this integration could be projected as quite large, and large scale integration (LSI) became the same sort of round-the-corner thing as speed of light limitations.

When it is precisely known what to make LSI seems promising indeed. But when there is less certainty and changes may have to be made after fabrication LSI becomes a problem. It is what it is and if even one detail is wrong it must either be accepted or thrown away.

The problem of changeability is increased when many different products might be brought out at about the same time. Partly for this reason IBM chose a hybrid approach to integration for System/360 production. Though the silicon chips were not initially integrated the entire manufacturing process was highly automated, affording many cost advantages. Integration of the chips was subsequently increased, moving nearer to LSI.

The IBM approach was transitional but it elaborated LSI technology. Circuit modules were mounted on boards with multiple layers of printed wiring. The characteristic impedance of this wiring was controlled at two levels: distributed transmission line runs and transmission lines lumped with successive circuit loads. Wiring between boards was conventional.

Control memory appeared as a new component in some systems, implementing the earlier idea of microprogramming. In this concept the regular machine instructions are themselves programmed from very elementary hardware operations. This eliminated the need for quite a lot of wired-in logic and illustrated one way of trading memory bits for logic circuits.

The first control memories were physically but not electrically changeable so that their contents would not be lost when power was shut off. The physical change required preparation of a new pattern corresponding to the new information content but this was still much easier than changing hard-wired logic. These memories made it feasible to alter the whole nature of a machine's instruction set after it was built. In particular, one machine could imitate another at good efficiency, a property which was used by IBM to "emulate" earlier machines on its 360 models.

Considerable experience with production of the first generation of transistor machines had also made it possible to modify the previous insistence on worst-case design. With knowledge of actual variances and distributions of component parameters it became feasible to use statistical design rules in which the concatenation of unfavorable tolerances could be made very unlikely. The higher was the degree of integration the more this kind of knowledge could be exploited in specifying the design unit.

SEMICONDUCTOR MEMORY

The advent of semiconductor memory closes a circle. ENIAC had used the same technology, vacuum tubes, for both memory and logic and now semiconductor technology provides the same commonality. In between has been the electrostatic memory and the ferrite core memory, both were important.

Semiconductor memory is now leading LSI. The regularity and simplicity of memory arrays and their interwiring allows the density of memory bits per chip to exceed logic circuits per chip by a considerable margin. In a curious inversion memory, which had been the difficult thing at the outset, has become the easy thing. Even more interesting is that semiconductor memory can be considered a fusion of its two immediate predecessors. The basic idea with electrostatic memories was to use the high resolution of an electron beam to put many bits on a small surface; and the basic idea with ferrite memories was to fabricate a structure for each and every bit but to seek maximum automation of that fabrication. With semiconductor memories the high resolution of the electron beam will be used to fabricate on a small silicon surface the bit-by-bit structure of a very large memory—and with the full automation inherent in the LSI process.

Semiconductor memory will replace ferrite memory on both cost and capacity grounds. It has already provided a less obvious but fundamental change in computer design. The old dream of infinite memory with infinite speed, fostered no doubt by the incredibly limited memory of early systems, gave way to the more realistic appreciation that a combination of a small fast memory with a large slower memory could be automatically managed to act, statistically, as a fast large memory. This change-of-kind in memory organization finally matched the logic change in going from tubes to transistors. The memory-logic speed gap which had prevailed with ferrite memories and had caused so much difficulty with high performance systems is now significantly improved.

REFERENCES

1 R SERRELL et al
The evolution of computing machines and systems
Proceedings IRE Vol 50 No 5 1962

2 N NISENOFF
Hardware for information processing systems: today and in the future
Proceedings IEEE Vol 4 No 12 1966

3 J P ECKERT JR
A survey of digital computer memory systems
Proceedings IRE Vol 41 No 10 1953

4 J A RACHMAN
Computer memories: A survey of the state-of-the-art
Proceedings IRE Vol 49 No 1 1961

5 L V AULETTA et al
Ferrite core planes and arrays: IBM's manufacturing evolution
IEEE Transactions on Magnetics Vol MAG 5 No 4 1969

6 W BUCHHOLZ editor
Planning a computer system: Project stretch
McGraw-Hill 1962

7 C A ALLEN et al
A 2.18 microsecond megabit core storage unit
IRE Transactions on Electronic Computers Vol EC 10 June 1961