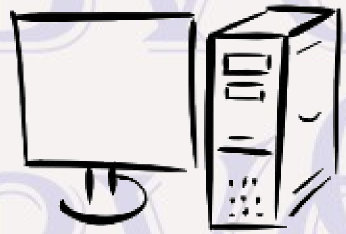
Bits & Bytes



Arkansas' Premier Computer Club

August 2025

The Bella Vista Computer Club - John Ruehle Center

Highlands Crossing Center, 1801 Forest Hills Blvd Suite 208 (lower level), Bella Vista, AR 72715

Website: http://BVComputerClub.org **Email: BVCCeditor@bvcomputerclub.org**

MEETINGS

Board Meeting: August 11, 2pm, in John Ruehle Training Center, Highlands Crossing Center.

General Meeting: August 11, 3pm. Program: Election of officers for next fiscal year (Sept-Aug), and a video on "8 Ways AI Can Help You Every Day".

AI tools like CoPilot, ChatGPT, and Perplexity can streamline everyday tasks such as summarizing lengthy content, simplifying complex texts, answering technical questions, and improving communication. They help users manage information more efficiently by providing clear, concise answers and tailored advice, making learning and problem-solving quicker and easier. Whether for work or personal use, AI is a versatile tool for boosting productivity and understanding.

We will meet in-person in John Ruehle Training Center, Highlands Crossing Center, lower level, 1801 Forest Hills Blvd, Bella Vista, or you may attend the meeting on-line via Zoom. Zoom access information is published on our website.

Visitors or Guests are welcome.

Consider attending by Zoom if you are unable to attend in-person.

HELP CLINICS

August 2, 9am - noon at John Ruehle center
August 20, 9am - noon at John Ruehle center
Members may request Remote Help on our website at https://bvcomputerclub.org at menu path Member Benefits ►Remote Help .

MEMBERSHIP

Single membership is $30; $15 for each additional family member in the same household.

Join on our website at https://bvcomputerclub.org at menu path Get Involved ►Join/Renew, by mailing an application (from the web site) with check, or complete an application and pay in person at any meeting.

CLASSES

(At BVCC Training Center)

"Basic Computer Knowledge: What is a Computer?", Thursday, August 21, 1pm - 3pm, with Joel Ewing.

Advance sign up required for each listed class: For reservations: email to bvccedu@bvcomputerclub.org, or sign up at the General Meeting. Classes are free to Computer Club members.

Check the monthly calendar and announcements for any last minute schedule changes at https://bvcomputerclub.org .

NEW OR RETURNING BVCC MEMBERS

We are pleased to welcome the following new members or members returning as BVCC members after an absence:

Judy Davis	Viah Carlson	Johnny Konkler

OFFICER CANDIDATES FOR AUGUST ELECTION

It is once again that time of year when The Bella Vista Computer Club elects officers. At the August General Meeting each year the membership elects for a one-year term a President, Vice President, Secretary, and Treasurer, who all also serve on the BVCC Board. In addition, we also elect either one (even years) or two (odd years) additional Board Members, who serve on the Board for a two-year term., so that there are three additional Board members serving overlapping terms.

Throughout the year, we periodically ask members who may be interested in serving BVCC in some capacity to let one of our officers know. Although elected officers normally transition in September, there are at times a change in circumstances that require a transition in mid-term; and it is always nice to have some available names in reserve. The descriptions of the various positions at BVCC can be found on our website under the menu options **Get Involved ►Help Us** .

Those who have agreed to run as candidates this August are

President:	Joel Ewing
Vice President:	Woody Ogden
Secretary:	Barbara Maybury
Treasurer:	Russ Ogden
Board (position 1):	Carmen Greenup
Board (position 2):	Geri Hoerner

Nominations from the floor of other members as a candidate for any of the above positions are permitted at the August meeting. The only conditions are that the member must be present and willing to serve in the office, if elected.

THE ORIGINS OF PROGRAMMABLE DIGITAL COMPUTERS (PART 1 OF 2)

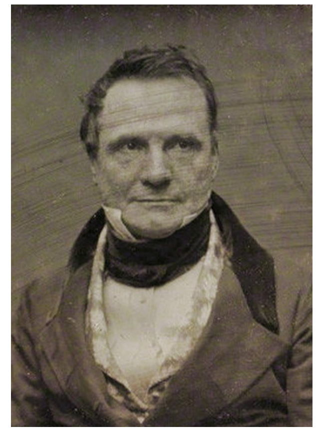
By Joel Ewing, President Bella Vista Computer Club
Bits & Bytes, August 2025
htttps://bvcomputerclub.org
president (at) bvcomputerclub.org

Much of the general public's direct experience with computers didn't begin until personal computers became widely available, which started in the 1980's, but didn't really grow exponentially in the United States until the ARPANET evolved into the public Internet with the passage of the *High Performance Computing and Communication Act of 1991*, often called the Gore Bill.

The First Programmable Digital Computer Design

If you ask ChatGPT who "designed" the first programmable digital computer, its answer is Konrad Zuse's Z3 electro-mechanical computer in Germany in 1941. This is an example of the subtle errors that are frequently present in AI responses. This answer is not correct unless the question is: Who is credited today with designing **and building** the first practical programmable digital computer.

The credit for **designing** the first programmable digital computer belongs to Charles Babbage (1791-1871) of England, who designed a mechanical general purpose programmable (by punched cards) Analytical Engine around 1834. The first known computer program, for calculating Bernoulli numbers on Babbage's Analytical Engine, was published around 1843 by Ada Lovelace, an English mathematician familiar with Babbage's design and daughter of poet Lord Byron. Although pieces of the Analytical Engine and of Babbage's special-purpose digital computers Difference Engine I and Difference Engine II were built as proof of concept, none of his machines could be built during his lifetime.



Charles Babbage

Babbage's goal was to create a machine to calculate and publish the mathematical tables needed to perform many other manual calculations, while avoiding the human errors that were common in many published tables of the time. These tables were an essential aid in scientific calculations and for accurate determination of position at sea. Unfortunately the large number of metal parts with precise tolerances made his machines economically impractical to build using the hand-manufacturing techniques of his time. Ada Lovelace is also credited as being the first to understand that a machine designed to work with numbers could also work with letters or any other symbols. just by encoding each symbol as a number.

Babbage was 100 years ahead of his time. It took new technologies developed in the 20th century to make computers with the features envisioned by Babbage practical. The wartime designs during the 1940's were made possible by availability of electric power utilities, electro-magnetic relays first developed to support telegraphy and automated telephone exchanges, and by vacuum tubes first developed to support radio and audio applications.

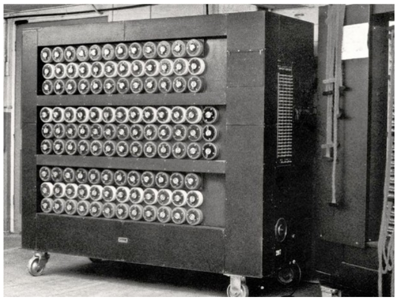
As an exercise of historical interest, in 2002 a fully-functional Difference Engine II was finally built from Babbage's original design using modern manufacturing techniques. The only change from his original design was to change the power interface from steam-driven power to a hand crank with a 4:1 gear reduction, making it possible for one person to power the machine. It is possible to view this machine in operation at https://www.youtube.com/watch?v=BlbQsKpq3Ak .

Theoretical Work On Computability

Independent from the design and construction of physical computers, in the 20th century mathematical research progressed in logic and automata theory. An "automaton" is a conceptual abstract device or machine for which "rules" can be defined that when followed for a given "input" will result in some computational result as "output", giving the solution of some problem involving numbers or symbols. The goal was not to build some physical machine, but to understand the potential capability and class of problems that could be solved with a very simple design. A number of theoretical designs were being studied and it was unclear what their differences in capability might be. In 1936 Alan Turing published a thesis, which along with works of several others effectively showed that all the various abstract machines were logically equivalent in what they could compute. A process could be described as "computable" if it could be computed by any of those abstract machines, and described as not computable if it could be proved it could not be computed by any of those abstract machines.

Any digital computer that can be programmed to emulate a Turing Machine (which includes all of what we now call general purpose computers) can compute anything that can be computed -- provided you give it enough time and resources.

Alan Turing's theoretical work, and his work on the Bombe machine that broke the German Enigma code at Bletchley Park during WWII laid the groundwork for the British Colossus project that broke other German codes, saving Allied lives and shortening the war. Those who have seen "The Imitation Game" movie will better appreciate Turing's wartime contributions.



Bombe Machine

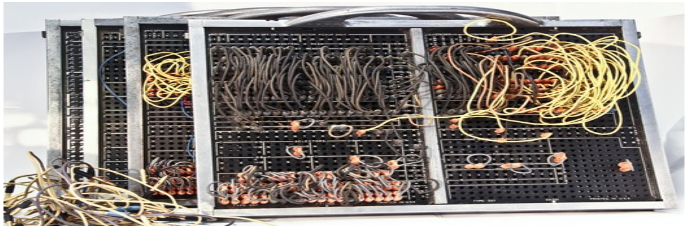
Developments Before World War II

During this period there were no general purpose computers, but there a number of specialized electro-mechanical machines, referred to as "unit record equipment" or "electric accounting machines". The 1890 census,which required 8 years to process, was accelerated by using a punched card tabulating system invented by Herman Hollerith. The impetus for this invention was the US Constitutional requirement to complete a national census and perform a Congressional reapportionment every ten years -- impossible if it took more than a decade to process the census data after data collection.

Several companies using different designs for processing punched cards evolved over the next several decades, and by 1930 there were three major players left: IBM and Remington Rand in the United States, and Groupe Bull in France. By the mid-1930's IBM was the dominant player in the US unit record equipment market.

The standard IBM card format introduced in 1928 had 80 columns in 12 rows with rectangular holes and multiple punches to represent letters and special characters. This was called a Hollerith card even though it differs from the cards invented by Herman Hollerith. Some of the peculiarities present in the Extended Binary Coded Decimal Interchange Code (EBCDIC) in use on IBM mainframe computers today are the direct result of that code being designed as an extension of the old Hollerith Code used on IBM unit record equipment, so that IBM mainframes would continue to be compatible with the keypunches, unit record equipment, and data cards already in use by their customers.

Many of IBM's unit record equipment and accounting machines could be customized for different applications by plugging wires into a removable plug board to change the ways the various components of the device were interconnected. Sometimes this was called "programming" the machine, but this was not programming in the current sense of the word.



Several IBM 407 Application Plug Boards

The first really massive use of unit record equipment in the United States was the Social Security Administration in 1935.

Computer Developments During World War II

WWII encouraged a major investment in computer research and development in the United States, Great Britain, and Germany. There were applications related to the war effort that would benefit from faster and error-free calculations: artillery firing tables, calculating missile trajectories, nuclear research, and code breaking.

Computer development in the various countries proceeded independently. In the United States there was collaboration between several universities and the military, and of course during the war this work was classified. Because work in Britain was related to code breaking, it was even more highly classified and remained classified for decades after the war -- the mere existence of the Bletchley Park Colossus project was classified until 1975, and the "full" story wasn't revealed until 1996 when the United States finally declassified its documents related to Bletchley Park. Karl Zuse's work in Germany was only partially classified, but isolation of Germany during the war kept his work from being well known. It took several decades after the war before some of the accomplishments of those working in other countries were known and acknowledged.

There were two main computer construction projects started during WWII in the United States: A computer proposed by Howard Aiken of Harvard, funded by IBM, based on electro-mechanical relays; and a computer designed by John Mauchly and J. Presper Eckert using vacuum tubes, built with others at the Moore School of Electrical Engineering, University of Pennsylvania, for the US Army.

Relay technology was the safer choice. It was a known technology with two states, on/off, which was easily adapted for the true/false binary logic required for a computer design, but inherently slower because it depended on physical movement of relay contacts. Vacuum tubes were faster, but using these continuously variable devices first required inventing circuits that would reliably assume only one of two states to represent on and off, and which would continue to work reliably as a tube's characteristics changed with age.

The Harvard Computers

The Harvard Mark I relay computer was built at the IBM Endicott, NY facility and reassembled at Harvard in February 1944., in time to run simulations of the implosion detonation of the first nuclear bomb for the Manhattan Project to a greater precision than previously possible. After that it was used by the Navy.

The Navy representative to the project was Lt JG Grace Hopper (1906-1992, PhD Math, Yale) who was involved with programming the computer. She is now honored as a computing pioneer and major designer of the first version of the COBOL programming language, the descendants of which are still in use today. "Amazing Grace" finally retired from the Navy as Rear Admiral at the age of 79, and in 1996 a destroyer was named the USS Hopper in her honor.

The team that built the Mark I continued to develop improved versions for the military without IBM's involvement after the war (Mark II, Mark III, Mark IV), gradually replacing relays with tubes, with the Mark IV completed in 1952 being all electronic.

The Moore School of Electrical Engineering Computers

The Eckert and Mauchly team at the University of Pennsylvania began work on the all-electronic ENIAC in early 1943 and it was completed in early 1946. John von Neumann, a mathematician associated with the Manhattan Project, began consulting with the group in 1944. To simplify its design and improve speed it was programmed by manual rewiring and changing plug boards, which could take several days. Its vacuum tube logic allowed it to add about 5,000 times faster than the Harvard Mark I. The war was over before it was completed, but was used for calculations to design the first hydrogen bomb. Because of a change in the university's patent policy, in 1946 Eckert and Mauchly separated from the Moore School and established their own company to continue their work

Even before the ENIAC was finished, design of a successor named EDVAC was started to address some of the most serious deficiencies of ENIAC. When operational in 1951 its creators believed it to be the first general purpose stored-program computer in the world, with its program stored in the same memory used for data. To simplify the design it used binary instead of decimal arithmetic and stored both data and program in binary. It was actually slower than the ENIAC, but a much more useful design because the program could be quickly changed. This computer provided a general structure that most future digital computers followed. Because the EDVAC was first described in a monograph published by John Von Neumann in June 1945, this is sometimes referred to as a Von Neumann architecture; but he was not the only one to conceive of the features it included.

The First Commercial General Purpose Computers In The United States

In 1948 Eckert and Mauchly began construction of a tube computer design UNIVAC I for the US Census Bureau. By the time it was delivered in 1951, they had become a subdivision of Remington Rand. The general public got their first introduction to the UNIVAC when one was used by CBS to predict the 1952 presidential election. It accurately predicted an Eisenhower landslide, but that prediction was not initially disclosed because it disagreed with the expected results based on polling.

In 1952 IBM announced its IBM 701 scientific computer. It would be the first of a series of machines. The UNIVAC and IBM machines were directly compared and were close computationally but IBM had a distinct advantage in the ability to input and output data. Over the next decade IBM would produce a variety of different machine designs and models, some targeted for computation-heavy scientific applications and others targeted more toward commercial large-data applications.

Other companies engaged in the production of digital computers in the US and in Europe. By the close of the 1950s, a dozen US vendors had delivered approximately 450 large-scale computers, of which 350 were from IBM. During the 1950's one of the cheapest computers was the Bendix G-15, available in 1956 for a cost of around $49K. One of the fastest computers developed in the 1950's was the IBM 7030 Stretch, delivered to Los Alamos National Laboratory in 1961 for $13.5M (1950 dollars), capable of about 1 M instructions per second.

Large businesses began to realize that computers that could handle large payroll and human resource records quickly could be cost justified. The impetus for developing larger and faster computers came largely from the government, specifically to model nuclear reactions and to model and predict weather Both of these applications require computations of conditions at hundreds of thousands of points over a period of time. In theory, the closer and larger the number of points and the shorter the time intervals, the more accurate the results. In the case of weather prediction, the problem is compounded in that we can only measure the weather at specific points in time at a relatively small number of points, and in addition weather is believed to be influenced by chaos theory (a small, possibly unknown, change at one location may eventually cause a large change at different location).

IBM had the advantage of already having a customer base using IBM's unit record card equipment, plus expertise in building unit record equipment whose mechanical technology could be adapted to provide card input/output and printed output from their computers. As an option with the IBM 701, IBM also introduced magnetic tape drives that were much easier to use than the heavy metal-strip magnetic tape reels required by UNIVAC tape drives. IBM ferric oxide film tapes became the data interchange standard that other manufacturers had to support.

IBM invented the computer disk drive, first commercially used in 1957. That first disk drive, the IBM 350 RAMAC, had fifty 2 ft diameter platters, was the size of two large refrigerators, weighed about a ton, and had a single head assembly that had to be moved to the correct platter, then to the correct track on the platter. A separate air compressor was required to supply air flow around the read/write heads so they would float over the platter surfaces without contact. The total storage capacity would have been equivalent to 3.75 MB today, with a 1 sec access time.

The transistor was invented in 1947, and by the late 1950's all new computers were designed with transistors rather than vacuum tubes, reducing their size and heat production. This transition didn't happen sooner because transistors were initially not as reliable as vacuum tubes, not widely available, and early transistors did not function well with higher frequency signals.

A 1964 Revolution In Mainframe Computer Architecture

By the 1960's some problems with increased corporate dependence on computers were beginning to emerge. The more applications that companies wrote for their computers and the larger their business grew, inevitably a more powerful computer was needed, but within the same family of computers growth options were very limited. Every time a major upgrade was required, transitioning to a different computer architecture, even from the same company, required that all programs be revised to run on the new hardware. The effort was reduced somewhat for applications written in higher-level languages like FORTRAN or COBOL but even there subtle inconsistencies had to be resolved; and there was always some code that for the sake of efficiency had to be written in the native machine language or Assembly Language (a symbolic representation of machine language) for a specific machine model. It was clear that if this trend continued, at some point the cost of hardware upgrades would become unacceptable.

IBM 360 Model 65

In April 1964 IBM made a major gamble and announced the S/360 architecture which to a large extent isolated the physical hardware implementation from the machine-level architecture visible to programs running on the machine. A rich set of "machine instructions" were defined in the architecture, designed to support both business and scientific applications and working with data types ranging from 1 to 256 bytes in length. The architecture provided sixteen 32-bit general purpose registers and four 64-bit floating-point registers for application usage. The more-expensive and faster S/360 machine models had wider data paths, and a faster processor with direct support for more of the machine instructions. Slower less-expensive models would have a simpler hardware design and would implement complex machine instructions by having the processor run hidden read-only "microcode" that instructed the simpler hardware how to perform a complex instruction as a series of simpler steps to achieve the requested action. In addition the S/360 architecture instructions were divided into privileged vs non-privileged instructions. Only the new Operating Systems would be allowed to execute privileged instructions and access additional Control Registers, where model-dependent actions could be taken and external Input/Output devices accessed. Application code had to request hardware resources and I/O actions by issuing requests to the Operating System.

In retrospect the S/360 architecture was incredibly successful in protecting the software development investments of its users. There have been a number of major upward-compatible enhancements to the architecture since its introduction, culminating in the current z-architecture. I witnessed at least ten major hardware upgrades over a 33-year period, at least five successors of the S/360 architecture, and a larger number of Operating System upgrades from MVS/XA (a successor of the original OS/360 Operating System) through its current successor z/OS; and none of these required a mass change of the installation's application programs -- within the same Operating System family old application code just continued to run. The application interfaces to the Operating System interfaces were also enhanced over the years to support new function, but IBM was careful to do this in an upward compatible way. If an old interface could not be changed in an upward compatible way, it was left and instead a new interface was added. To take advantage of some new function or resources might require later application changes, but not because of the upgrade itself. The only time mass re-compiles of applications were required was when we changed from IBM's DOS/VSE Operating System to the MVS family of Operating Systems, which had totally different application interfaces, but that was done gradually over a three-year period as applications outgrew the resources available under the old system. Both systems were actually running on the same physical hardware during this overlap.

These upgrade issues have not yet been solved in the PC world. Under both MS Windows and Linux, it is very common that a new version of the Operating System may require corresponding upgrades to many applications that run on those PCs. For most individual users, this doesn't involve their own code, but may involve additional expense to get application upgrades. Corporate users that write proprietary applications to support their customers on PCs still have issues if they choose tools that cease to work under new versions of an Operating System.

Large mainframe disk storage has also evolved over the decades with different architectures having different track capacities, different number of cylinders per device, and different total capacity. This didn't require application program changes, but could be an issue for running an applications in that the specification of the disk space might need to be changed, either to get enough space or just to improve efficiency. That issue was finally dealt with in the 1990's when physical disk drives were replaced by large arrays of smaller drives that emulated the last physical disk architecture (IBM 3390). At that point disk architecture became stabilized and enhancements to the physical disk hardware is now only visible to applications as improved performance and increased storage capacity.

(to be continued in September)