

## UNIVAC I Computer System



22346

Net Zero Dial Up Provider

This web page was put together by Allan Reiter in 2004. I worked on UNIVAC I (serial #10) at Wright-Patterson Air Force Base (WPAFB) near Dayton, Ohio. The web page is in two parts. The first part, shown on this page, has a blue background. It contains technical information of how UNIVAC I really worked. This includes the mercury delay line main-memory and the computational and control circuits that also used mercury delay lines. The second part has a yellow background. It mentions various people who worked with the UNIVAC I. Both parts are optimized for a screen display setting of 1024 by 768 pixels.

**Enhancement information, the blue page:** This part is complete except when I think of something that would be useful. I have included a date and the enhancement on the contents shown in the blue part, which is the technical part. The last enhancement made was on 6/30/09 to "Programming and FLOW-MATIC."

**Enhancement information, the yellow page:** This part was enhanced in 01/08/07 to give additional information of the Simulator, which I used in the program example.

I am still waiting for information on the hard code of the Music Program that came with UNIVAC I. I would like to hear it play music. After looking at this blue page go to the [yellow](#) page to read about some of the people who worked with UNIVAC I.

My email address and two links are shown below.

### Some Links

The UNIVAC [Video](#)

The UNIVAC [Simulator](#)

reitory2k@q.com My email address for copy and paste.



**The UNIVAC I SYSTEM (1951).**

From the top clockwise: Central Computer, UNISERVOs, Off-line Uniprinter, On-line Typewriter, Supervisory Control (SC), and Off-line Unityper. Just to the right of the Unityper stands a Tektronix oscilloscope used for maintenance functions. There were 46 UNIVAC I's built.

I have included three pieces of music that somewhat reflects the operation of the UNIVAC I. They are midi files and you can start them by using the small consoles below. As you read about how it worked and listen to the music you will often notice pauses followed by beautiful music. Think of the pauses as UNIVAC I waiting on a word from a long mercury tank (main-memory) and then quickly processing the word.

Daisy

Faithfully

My favorite

*This is a request for information from those of you who may have knowledge of the music program that was used to enter melody notes into UNIVAC I that would then generate a program that would play that music. Some of you may have heard the Christmas music that was played by UNIVAC I. This program was used to play any tune and came with a set of Christmas music. I am having a lot of trouble finding any source code or an old tape of it. It was written by a young USAF Lieutenant by the name of Herb Finney at the Office of the Army Air Comptroller, which used UNIVAC I (serial #2). He recognized that by coding certain computer instructions, he could play music on the speaker system of UNIVAC I. Every UNIVAC I installation had a copy of this program tape. We (Ed Lankford, Peter Ingerman and I) have a copy of the code for "Ansero" and are trying to get a copy of the hard code for the music program so Peter through a reverse engineering and programming process can play the diagnostic program "Ansero" on his Simulator as well as other music. Please contact me if any of you know Herb Finney. My email is shown above.*

This part is rather long. For viewers familiar with this page I have added headings to various subjects and a direct way to get to them via this table of contents. Return here by using the browser back function. I occasionally make enhancements when I think of something that would be useful. I have included a date and the enhancement on these contents.

<p><b>Introduction</b> 05/24/08 The XS-3 character set</p> <p><b>Long Mercury Tanks (Main-Memory)</b> 05/15/06 11.25 Mhz carrier</p> <p><b>Short Mercury Tanks (Comp and Control)</b> 03/04/06 The fish</p> <p><b>Starting and Operation</b> 09/05/06 The Booting process</p> <p><b>Memory Switch and Instruction Execution</b> 07/22/08 Univac II</p> <p><b>A Few Words about UNISERVOs</b> 07/14/08 The Erase Head</p> <p><b>Inside a Long Mercury Tank</b> 07/02/06 More data</p> <p><b>A Trip through the Computational Circuits</b></p> <p><b>Circuit Operation and Trouble Shooting</b></p>	<p><b>Programming and FLOW-MATIC</b> 06/30/09 Instruction Update</p> <p><b>Subroutines and Libraries</b></p> <p><b>Cooling and Maintenance of UNIVAC I</b></p> <p><b>UNIVAC I Instruction Set</b></p> <p><b>Photos</b> 09/22/05 2 photos of Graces work</p> <p><b>High Speed Printer (600 LPM)</b> 05/26/09 Hardware note</p> <p><b>UNIVAC I Block Diagram</b></p> <p><b>Main-Memory Whiffletree Diagram</b></p> <p><b>Flowchart Template and Code Card</b></p>
---	---

**Introduction:**

In operation with several tape drives rapidly moving backwards and forwards, console neon lights flashing, console typewriter typing program messages automatically from time to time, an operator keeping an eye on things, the UNIVAC I System was an impressive sight. Many dignitaries visited serial #10 at WPAFB. One of the most notably was retired 5 Star General Douglas McArthur (1880-1964) who was Chairman of the Board of Remington Rand Inc. Every one present was at attention that day.

Watch the UNIVAC [Video](#) to see it in action, you will need a high speed connection to the internet because it takes about 3 hours to download it using a 56K dial up connection).

It was big, called garage sized, a walk in computer, the alpha computer, a genius. It was all of these and more. It weighed nearly 15 tons, had 5,200 electron tubes, 18,000 germanium diodes, consumed 125 Kw of electrical power, and had an elaborate water cooling system with large fans to keep it from burning up. Wiring and connecting cabling alone weighed more than half the overall weight of the system and often required false flooring to hide and protect it. This increased its height and made it look even bigger. People in its vicinity knew when it was being powered up. The lighting would dim momentarily. It really had character and in a way took on a life of its own.

According to electron tube failure rate tables some analysts suggested it would be down most of the time if it ran at all. UNIVAC I, however, proved to be a beautiful piece of computing equipment, much beloved by many of the people who were fortunate enough to rub shoulders with it.

The UNIVAC I was a stored program computer with the ability to modify its program instructions. Ten UNISERVO tape units were used for input and output data. It contained a 1,000 word re-circulating memory called an acoustical delay line where data, program instructions, and constants were stored serially bit-by-bit in columns of mercury. Each word held twelve 6-bit characters. Each character held 6 binary bits containing ones and zeros. An extra bit position for each character to accommodate an automatic odd parity check made for an 84-bit word. The words were separated by a 7-bit space between words, which provided circuit setup time during instruction execution cycles. Data words and constants contained any combination of 12 characters. Numeric data words contained 11 numeric characters with a sign in the left most character position for the 12th character. Program instruction words contained 2 instructions in each word, one in the upper (left) 6 character positions and the other in the lower (right) 6 character positions. Instructions that addressed the 1,000-word memory were used in three lower characters of each of these instructions and were numbered 0 thru 999.

The character set was comprised of 63 Excess-3 (XS-3) binary characters. The 51 printable characters included the 10 numeric characters 0 thru 9 and 2 characters + - for their signs, 26 characters A thru Z of the alphabet, and 13 characters #) \$ \* : / % ' & ( , . ; for punctuation and for special characters. Of these 63 XS-3 characters there were 12 non printable characters that were used to control functions of the High Speed Printer and Uniprinter. These are explained in the heading "High Speed Printer (600 LPM).

John W. Mauchly (1907-1980) and John Presper Eckert Jr. (1919-1995) were the brains behind UNIVAC I. Mauchly was a Physicist and Eckert was an Engineer. They both had already developed a computer machine called the ENIAC at the University of Pennsylvania in 1946, which was used for calculating artillery-shell trajectories for the Army. In 1944 they envisioned building an automatic business computer and started their own company in about 1947 called the Eckert-Mauchly Computer Corporation (EMCC). This computer turned into the UNIVAC I. Mauchly was an applications expert and went on trips to convince customers about the new computer they were building and how they could use it in their businesses. Eckert was back in the plant directing the work that needed to be done to make it work. He was a real hands-on engineer and could do anything and everything with electronics. EMCC was forced to raise money to stay in business before they had completed it, though. Remington Rand Inc came to their rescue in 1950. Both Mauchly and Eckert continued working for Remington Rand for many years. Remington also brought their programmers with them. Their names are mentioned in Morgan Huffs input that is included in the yellow page. The programmers were called computers rather than programmers in those days.



*UNIVAC I was truly the first modern computer. All subsequent computers are merely improved versions based on ever continuing advances in micro electronics, memory, and communication technology. It entered a market dominated by punched card tabulating equipment. It came with supporting equipment to convert or translate card data to and from metallic tape reels. Now a card tabulating system at a large company is about as rare as a stagecoach. Remington Rand Inc made a bold move by bringing the dream of John W. Mauchly and John Presper Eckert to fruition and beyond. It changed the way we live. The very computer you are using to view this page is a*

*direct descendant of the alpha computer UNIVAC I.*

### Long Mercury Tanks (Main-Memory):

The mercury delay line memory was contained in 7 rather large mercury tanks called long tanks. Each tank contained eighteen 10-word channels. Each channel was a horizontal column of mercury with sending and receiving piezoelectric quartz crystals in physical contact with each end of a column of mercury. The channels were separated by metal tubes, which acted as wave guides as the data bits moved through the mercury. Each 10-word channel held 910-bits, 840 for the 10-words plus 70 for the 10 spaces between these words. The 910-bits would re-circulate every 404  $\mu$ -seconds. The frequency of the carrier wave for the 910-bits as they moved through the mercury column was 11.25 Mhz, which is the 5th harmonic of the 2.25 Mhz main clock. The main clock was exactly in sync with the 910-bits of a 10-word channel and provided timing for all operations of UNIVAC I.

The photo shows a long tank that was mounted in bay position MV and contained sixteen 10-word channels for memory locations 800 - 870 and 900 - 970. It



used another channel for temperature control and the eighteenth channel was a spare. This is an original photo of one taken in Ed's attic in July 2004. The tank was from serial #24, the first Nashville L-C Ins computer. They were heavy and it took 2 or 3 people to replace one. It was difficult enough just to get three people inside "the box", which is what the inside of UNIVAC I was called. There were also 6 more of these tanks in the way. The other photo shows 5 of the 18 wave guides inside of each tank.



The seven long tanks of UNIVAC I were harnessed together in an electronic circuit called a whiffletree, which is a word not often used today and it somewhat fascinated me that it was used in describing the mercury memory linkage of UNIVAC I. You can see what that whiffletree looks like in a diagram near the end of this page. It could be valuable information if you ever come across a long tank.

Storing data in a delay line was accomplished by exciting (vibrating) the sending crystal with the data bits of a word, ones and zeros, at a rate controlled by the 2.25 Mhz main clock. This data signal was mixed with the 11.25 Mhz carrier wave. It was generated as the 5th harmonic of the 2.25 Mhz main clock. The resulting data signal would then begin moving through a column of mercury to the receiving crystal where circuitry picked up the data signals. They were amplified, reshaped, retimed, and then routed back to the sending crystal for another trip through the mercury. It was only when the data bits emerge from the receiving crystal end of the mercury column that word access was possible.

The average access time of a memory word was 222  $\mu$ -seconds. This was slightly more than half of the 404  $\mu$ -seconds re-circulation time of these delay lines so there was a fair amount of time used up waiting for word access. Averaged into this access time were wait cycles for switching between memory channels and memory tanks, which occurred during normal program operation.

In addition to the 100 channels comprising the 1,000 word memory 26 other 10-word channels were used for related activities: 6 for input buffering, 6 for output buffering, 7 for mercury temperature control (one for each tank), 1 for the 10-word register (rY), and 6 spares. According to the whiffletree diagram the temperature control channels were strategically placed in physical position 18 that put them in the right center of the mercury pool, and the spares were in physical position 14 that were located on the bottom of six of the tanks. Tank MX contained the 10-word register (rY) in physical position 9 and had no spares. Each tank was meticulously tuned at the factory and they were very reliable in operation. Once in a while a crystal would crack causing a mercury leak and that was a real mess.

On a cold start it often took about a half an hour before the system would hold memory. This would occur when the long tanks were at operating temperature and able to hold zero characters in each of the 1,000 words. The temperature control channel circuitry on each mercury tank checked output-timing pulses to control a two-stage tank heating system. Alternating current for initial quick heating and direct current to hold temperature. The mercury temperature affected the transit time of the bits and had to be tightly controlled to avoid bit creep. When all 7 tanks were "holding memory" a green light on the SC console would come on and UNIVAC I was ready to go to work.

*Mercury delay lines, electron tubes, and associated circuit technology of the 1950's were essentially pushing the state of the art at the frequency of 2.25 Mhz. Increasing the clock speed and packing more bits into a fixed delay line would just not work reliably. Even if it did work the word wait time delay would still be present. i.e.: If your program wanted word 5 and it had just passed the word access gate you would have to wait for all the other words in that mercury column to go by before word 5 again came into access position. Quantum leaps in technology for magnetic core memory, transistors, and circuit miniaturization with low power requirements solved all these problems and more in modern high-speed computers.*

*Programmers, however, can be very innovative and some took steps to minimize this word delay of UNIVAC I. Peter Zilahy Ingerman, PhD, the developer of the UNIVAC Simulator, was one of these and I am including his words on this subject. "When we were coding programs where there were long computational loops that ran often enough to make the effort worth while, actually programmed in minimum latency coding. For example, if you needed to execute a "B" instruction to get a constant to rA, you put the constant in memory so it would be immediately available when the "B" instruction wanted it. It was a slow and tedious process, but it was possible to get significant speed-up by doing it."*

### **Short Mercury Tanks (Computation and Control):**

Short tanks were mercury delay lines used for the computation and control operations. They were designed to work with a 12-character word, which included the 7-bit space between words for a total of 91-bits. The frequency of the carrier wave for the short tank bit streams as they moved through the mercury columns was 11.25 Mhz, the same as the long tanks. The short tanks differed from the long tanks in that several of them included external distribution delay lines of various lengths in electronic circuitry. This permitted multi-bit access to characters for computational purposes. The bit lengths of the distribution delay lines of short tanks were varied according to what they were electronically designed to do.

The four Computational registers, (rF), (rL), (rA), and (rX) worked closely with the adder, comparator, and multiplier-quotient circuitry (computational circuits) to perform the arithmetic, comparison, and transfer operations. These four registers were duplicated and compared to each other bit by bit to insure accuracy of computation results. Register rA was the dominant register. Its distribution delay line circuitry held 21-bits. This permitted access to several bits across characters during the time new numeric characters were being developed. For shift instructions its distribution delay line circuitry was increased or decreased by 1 character (7-bits) and the register contents circulated "n" times to obtain the desired shift. Arithmetic operations were performed using the XS-3 codes where a binary zero (000000) was expressed as a binary three (000011). The XS-3 codes handled binary carries like decimal arithmetic and could be implemented with less electronic hardware than other binary systems.

The four Input/Output Synchronizers (rSYI1), (rSYI2), (rSYO1), and (rSYO2) were used with the 60-word read and write buffers. rSYO2 had an additional function of providing programmers and operators a way to utilize the SC console keyboard and typewriter for 1-word input and output operations.

The three Control registers, Control Counter (rCC), Control Register (rCR), and Control Register 2 (rCR2) were used for controlling the flow of the program instructions. Short tank rCR2 was the shortest short tank. It was used only as a half-word delay line for timing the loading of the right instruction of rCR into the Static Register (SR).

The two-word register (rV). It was used as a holding area during a two-word move. It was also the longest short tank.

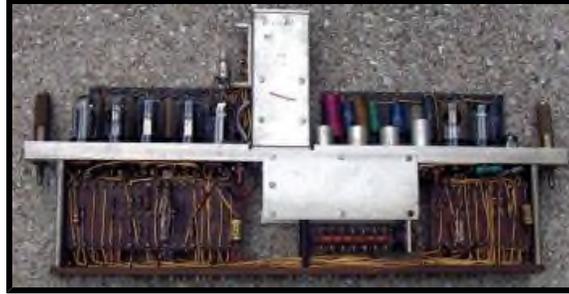
The short tanks held from 41.5 to 88-bits in mercury. The remaining bits of the 91-bits were held in external delay line circuitry. Two exceptions to this were rV a two-word tank that held 179-bits for a total of 182-bits, and rCR2 a half-word tank that held 41.5-bits. The pulse times within the mercury of each short tank was stamped on the nameplate of each tank: rF=88, rL=88, rA=67, rX=81, rCC=79, rCR=79, rCR2=41.5, rSYI1=88, rSYI2=88, rSYO1=88, rSYO2=83, and rV=179. The time it would take for the 91-bits of a 1-word register to circulate (40.4  $\mu$ -seconds) was called a minor cycle. The external delay line circuitry performed two functions.

1. Kept the bit stream alive and in sync with the 2.25 Mhz main clock when it emerged from the receiving crystal. This was done with a series of detectors, amplifiers, and pulse formers that took 3 pulse times to accomplish. This operation also kept the bit streams alive in the long tanks.
2. Multi-bit access to the registers that needed them for computations: rA(21), rX(7), rCC(9), rCR(9), and rSYO2(5).

The short tanks were mounted on several tube chassis through out the system. Each short tank was heated, but did not require the elaborate temperature control timing mechanism of the long tanks because their bit transit time was about 10 times less than bits in the long tanks. Counting the 4 duplicated registers rF, rL, rA, and rX there were 16 short tanks in UNIVAC I. The chassis that contained the duplicated registers were interchangeable, rF for rF, rL for rL and so on. This was an important aid during difficult troubleshooting sessions.

The 2 photos are of a rCC short tank chassis. The one with the nameplate reads Subsidiary of Remington

Rand Inc. The mercury tank is shown on the photo of the chassis as white with it's pulse times marking on the top. The second photo identifies the pulse times as 79 PT within the mercury. Note the type of screws that were used to assemble the unit. They did not have Phillips screws in those days. The rCC was a



straightforward register, nothing fancy. It kept track of the memory address of the program being executed and provided the address of that location. It basically had two functions that changed its contents.



1. Advanced the memory address of the program by one so the next two instructions were ready for the rCR.
2. Received the memory address of a transfer control instruction so the next two instructions for the rCR would start at another point in the program.

Many vital circuits of the computational area, including the 4 mercury registers rF, rL, rA, and rX, were duplicated to provide accuracy of the highest order. Both operations were constantly being compared and had to be exactly the same. UNIVAC I was never known to have made an error. It did what ever the programmer instructed it to do that often resulted in a programming error or bug. This brings to mind an occurrence that happened at US Steel in Gary, Indiana. That UNIVAC I (serial #14) had a custom cooling system that was cooled from input piped in from Lake Michigan that I find somewhat amusing. A fish blocked the intake causing the computer to overheat. That computer bug turned out to be a fish.

### Starting and Operation:

When the UNIVAC I long tanks were warmed up and holding main-memory the operator would latch a program tape onto a UNISERVO. This tape was produced off-line by a Unityper operator from coding sheets prepared by programmers. Data tapes referenced by the program instructions were on separate tapes and latched onto other UNISERVOs.

The operator would then clear the main-memory and registers and began the "Booting process". Usually the program tape was assigned to UNISERVO number - and the program was loaded starting at main-memory address 0. The "Booting process" was initiated from the SC console by setting the Interrupted Operation Switch (IOS) to the "One Instruction" position. Then two bits of the SR switches were set by "locking in" the 3rd bit from the right of the 7-bit instruction by placing that SR switch in the "down position" and then "locking out" the 1st bit in the "up position". This prepared the 1 instruction (in XS-3 code) for reading the first block of the program tape into the 60-word input buffer memory. Then the number of the UNISERVO program tape (in XS-3 code) was set (not locked) into the 5-bit SR character next to the instruction character. By hitting the Start Bar that operation took place and then stopped showing the 30 instruction in the first two characters of SR and the remaining three characters reading 0 (all in XS-3 code). The 3 character of the 30 instruction would show up because of the locked positions of the SR. By hitting the Start Bar the second time the 30 instruction would transfer the 60 word input buffer memory into main-memory at addresses 0 thru 59 and stop. The operator would then clear the locked positions of the SR and some other counters, and set the IOS to "Normal". Then by hitting the Start Bar the UNIVAC I would begin the automatic execution of the program instructions. The Start Bar was located on the bottom of the SC console.

The first instruction of the first block of the program just entered would then be executed to start the program. All UNIVAC I operations, including reading portions or all of the program tape into main-memory, would then be in the automatic mode. Remember the "A" of the word UNIVAC was for automatic. If everything was in order UNIVAC I could run for hours with the operator adding, removing, labeling, and stacking the 5-pound metallic data tapes as indicated by automatic responses via the on-line SC typewriter.

Production runs at WPAFB were really long and operators were vital links in keeping the whole operation in order. Faulty recovery from minor problems, like dropping a tape loop on a UNISERVO, could wreck several hours of work. Long runs almost always had breakpoints or rerun points built in for recovery.

### **Memory Switch and Instruction Execution:**

This is how the UNIVAC I worked. A Cycling Unit (CU) produced the 2.25 Mhz clock pulses that were in sync with the bits traveling through memory and provided timing for all operations. A memory switch provided access to words in the re-circulating main-memory. The circuitry of this switch included the Time Selection Counter (TSC) that kept track of the words (0 thru 9) moving in unison through all 10-word channels, except for the 10-word register (rY) that operated independently as a holding area for a 10-word move in the same manner as the 2-word register (rV) in the short tanks.

In order for TSC to open the memory switch it needed additional information. This was supplied by the SR, which contained the program instruction with the specific memory word location. When that memory word came into position for access (exiting from the 10-word channel) the switch opened and that word flowed either in or out of memory depending on the instruction being executed. The word also continued to re-circulate unless the switch was receiving a replacement word for that location.

Words traveled to and from memory and registers on a multipart High Speed Bus and Amplifier (HSBA) system. All 12-characters of a word traveling on the HSB were always checked for an odd number of bits. Every five seconds the entire main-memory (1,000 words) was cycled through the HSB for an odd character check. The HSB system was duplicated because it was attached to the duplicated computational circuits. There was a speaker attached to the HSB for optionally listening to the sound the program produced. One could always tell when a program was in a loop.

The sequence of instruction execution began with the rCR being loaded with the 2 instructions from the memory address specified by the rCC that kept track of the current memory address of the program. They were copied from the rCR register into a circuit distribution delay line that contained the 6 seven bit characters (42 bits) of an instruction. When the instruction was in position, one pulse was gated to the 27-bits that UNIVAC I required and copied into the SR. The SR carried out the actual instruction execution. The left instruction was executed first, then the right.

Only 27-bit positions of SR contained useful information for UNIVAC I. The 1st instruction character contained a 7-bit character. The 2nd instruction character contained 5-bits for various instruction options including the numbers of the 10 UNISERVOs (- 1 2 3 4 5 6 7 8 9). This character was also part of the six instructions (00, 10, 30, 40, 50, 90) that required a second instruction character as a 0. The 3rd instruction character was not used. The 4th, 5th, and 6th instruction instruction characters were each 5 numeric bits for addressing long tank memory for instructions that required access. All five characters in the SR included a check bit and were in XS-3 code.

The SR was the gatekeeper. Six bits of the instruction character (the 7th bit was only used for the odd even check) supplied signals to a Function Table circuit, which opened and closed electronic gates consistent with the instruction being executed. A Cycle Counter (CY) produced Time On and Time Off gates called Alpha, Beta, Gamma, and Delta. These signals were closely associated with the operation of rCC, rCR, rCR2, SR, TSC, Program Counter (PC), and the Function Table during execution cycles of the 2 instructions in the rCR. Briefly the four stages of the CY worked like this.

1. Alpha time copied the contents of rCC to SR.
2. Beta time then used SR to copy the 2 program instructions from long term memory to rCR. It then copied the 27 bit instruction of the left instruction of rCR to SR, and increased by 1 the contents of rCC using a bit stream of a 1 generated by the CU to the adder.
3. Gamma time executed the left instruction in SR. It then copied the 27 bit instruction of the right instruction of rCR to SR by running it through the 1/2 word mercury delay line of rCR2. This delay was necessary for timing the setting up of the right instruction in SR.
4. Delta time executed the right instruction in SR.

Time Off gates were set in each of these four stages. During that time signals were built up in the Function Tables to perform the operations but were blocked until they were stable. The length of the Time Off gates would be from 1 minor cycle to many minor cycles depending on the memory being accessed. Long tank memory took longer than short tank memory. Short tank memory access took 1 minor cycle in Alpha time by accessing rCC, Beta time by accessing rCR and rCC, and Gamma time by accessing rCR. Long tank memory access also took place in Beta time, Gamma time, and Delta time. This could take many minor cycles because they were accessing long tank memory, which could be any addressable word in a 10-word channel. This averaged out to 5 minor cycles of wait time when programming in average latency coding (no

regard to access wait time). When a long term memory location that was waiting in the SR became available the Time On gate in conjunction with the TSC would open the memory switch. Experienced programmers who wanted to reduce this access wait time to speed up their programs often programmed in minimum latency coding by positioning their codes to make immediate access available. During the Gamma and Delta stages there were additional Time Off gates under control of a Program Counter for the several steps needed to build up the various Function Table signals used by many of the instructions. This four stage sequence is then repeated for the next 2 program instructions.

The SR was the only place in UNIVAC I where circulating memory content was stopped and used. All other memory content, including the word or words being changed by the instruction being executed, continued to circulate or travel in mercury delay lines. There were a lot of moving "parts" in UNIVAC I memory.

The UNIVAC II was built in St. Paul, MN from a UNIVAC I (serial #16). There were 28 UNIVAC IIs built. Some of the major changes that were made to UNIVAC I to make it a UNIVAC II are shown below because they fit into this section quite nicely.

>There were literally hundreds of wiring changes that were made and soldered into place.

>Removed the 7 Long Mercury Tanks from UNIVAC I and replaced them with the new magnetic core memory.

>Removed the old UNISERVOs and replaced them with new updated ones. The tapes included both the old metallic tape and a new plastic tape. It also used a much-improved design by two electronic servos that sensed the positions of the large tape loops by changes in vacuum pressure. It was easier on the tape because it eliminated the two tape loop pulleys on the bottom of the loops that were used by UNIVAC I.

>Removed the old black Supervisory Control unit (SC) and replaced it with a new white one. Since the Static Register (SR) is contained in the SC it had to show the missing 3rd instruction character that was not in the old one.

>Deleted Alpha time, and extensively modified Beta time, Gamma time, and Delta time. This sped up UNIVAC II considerable because each word in the new magnetic core memory was immediately accessible.

>Added the unused 3rd instruction character in the SR to accommodate the new magnetic core memory that contained addresses from 0 up to 9999. This took an additional 5-bits (in XS-3 code) in the SR. The 2nd instruction character in the SR was changed from 5-bits to 7-bits for additional instruction options including 16 UNISERVOs. The 6 additional UNISERVOs for UNIVAC II were numbered A B C D E F. The SR for UNIVAC II now contained 34-bits rather than the 27-bits that were used for UNIVAC I.

>Modifications were made to the HSBA because UNIVAC II still used mercury in the Short Tanks for Computation and Control operations. The word in the new magnetic core memory was being accessed in parallel (or all at once) and had to be processed serially into and out of the applicable Short Tanks.

>This is the end of major changes made for UNIVAC II.

### **A Few Words about UNISERVOs:**

There actually was another place where memory content was stopped and used. Each 7-bit character of a word and a timing sprocket pulse were written together across the width of a metallic tape by a UNISERVO in one pulse time. During reading each of these 7-bit characters would then have to be put back into circulation again. The Input/Output Synchronizers, rSYI1, rSYI2, rSYO1, and rSYO2, which were short tanks, were used for this during reading and writing operations. The 7-bit space between words was ignored during writing and added during reading by these Input/Output Synchronizers.

Each of the metallic tape reels had the capability to hold 2,000 blocks or 1,440,000 seven bit characters. There was also an instruction to write an output at 22 characters per inch (CPI) for use with the Uniprinter data. The normal block size was 60 words (720 characters) written at 128 CPI at a tape speed of 100 inches per second. Each block was 6 inches long and the space between blocks was 2.5 inches that provided time for the Read/Write motor to get up to speed. Each block for reading and writing required the tape to be started and stopped. The data order in the 60-word input register rI when reading a block in either the forward or backward direction was identical. Writing a block from the 60-word output register rO, could only be done in the forward direction. When writing a tape there was an Erase Head that was located before the Read/Write Head. This cleared out the entire width of the tape so that any previous tape could be re-written.

Two Reel Drive motors were powerful devices to perform their functions with the 5-pound metallic tapes at

these speeds. When a tape reel was latched to the left drive it was fastened to the UNISERVO leader with a connector on it that was shaped like a small square "U" with two prongs facing each other. This leader was already threaded through the tape path to a fixed tape reel on the right drive as a result of rewinding. The connection to the tape was made by stretching the two prongs apart and inserting them into a small cylinder at the beginning of the tape. Fastening the tape to the leader was somewhat of a challenge and it was best to let an operator do this. Programmers occasionally injured themselves by getting the prongs caught on a fingernail or on a hand, which could lead to painful bleeding. Ed Lankford said this prong connector was changed in later stages of development to eliminate the two prongs by replacing them with an improved connector.

When the UNISERVO was turned on after a tape was loaded, it would be in a "first block condition". The tension on the tape at rest was held by two electronic servos that sensed the positions of the two tape loops in the tape path. When the first block was read or written the tape leader moved into position beyond the Read/Write Head. This would take about 1.5 seconds.

A third motor was called a Capstan motor (or the Read/Write motor). It was a smaller unit that was used for advancing the tape over the Read/Write Head at the speed of 100 inches per second for the actual reading and writing operations. The Capstan motor changed the position of the two tape loops so it did not have much resistance and got up to speed rapidly. The changed positions of the two tape loops also affected the two electronic servos that controlled the Reel Drive motors. They responded quickly and moved the heavy metallic tapes to equalize the two tape loops. Ed called these Reel Drive motors "really big horses" because he replaced several in Nashville and said they were really heavy to handle. The Capstan motor had a pulley that was coated with cork to keep the metallic tape from slipping.

A fourth motor was necessary to advance a small Mylar tape reel slowly between the metallic tape and the metal Read/Write Head to protect the Read/Write Head from wearing out.

UNIVAC I's were equipped with a fish line with springs and pulleys to control the two electronic servos arms. The photo shows a UNISERVO that was used by UNIVAC I. The cover is removed so you can see the entire tape path and the two electronic servos arms.

A plug board to switch between physical UNISERVOs was provided. This feature was used because UNISERVO numbers were hard coded into each tape instruction. When a particular UNISERVO gave trouble it could be fixed off-line by easily switching to another one.



When characters were written there was great care in selecting the placement of bits to insure accuracy of numeric computations. Character skewing was less a problem the nearer the Read/Write Head was to the timing sprocket pulse. The sprocket pulse (position S) was in the center or near center as there were 8 bits in total with the sprocket pulse. The check pulse to insure the bit count was odd (position 1) and the 4 lower bits (positions 4,5,6,7), which included the vital numeric characters, were crowded next to the sprocket pulse. The remaining 2 bits (positions 2,3) were for the alphabetic and special characters and were located on the outer edges of the tape. The bit numbering positions on tape were 3,5,1,S,6,7,4,2.

Another feature that was quite valuable was bad spot detection on tape reels. In an off-line process the tape was read and when a defect was found holes would be punched and the UNISERVO would bypass them. In addition when an error was detected in any part of the tape, error detection circuitry would force multiple backwards and forwards reading at different gains to bring the data back into conformity. A reading and writing operation, once initiated by the Central Computer, took place independently. This allowed the computer to continue executing program instructions during the time the UNISERVO was operating. This is an example of an asynchronous operation of UNIVAC I.

Even though the tape reels moved at a very fast rate they were slow compared to the UNIVAC I main clock rate of 2.25 Mhz. UNIVAC I could keep several UNISERVO's busy during reading and writing operations such as sorting. The UNISERVO's were the center of interest to visitors because of their very quick movements both forward and backward. Several of them in motion created a unique shuffling type of sound and a sense of a lot of things happening fast.

When troubleshooting read and write problems the bits on the tape could actually be visually seen. The tape in question was then put into the Uniprinter mode and written at 22 CPI rather than at 128 CPI. The tape was un-spooled and a portion of it was dipped into a solution containing iron filings. One could then read the bits clearly. A weak showing in a particular bit position or other problems could then be detected. This procedure was called vis-a-maging the tape.

When the High Speed Printer (HSP) and the Tape to Card Converter (TCC) came available there was a need to subdivide the 60-word block into increments of 10 blockettes. For the HSP the increments were 1 inch and for the TCC .1 inch. These were selected by 10 Block Sub-Divider Buttons on the SC, with UNISERVO 1-7 for the HSP and 8, 9, and - for the TCC.

### Inside a Long Mercury Tank:

Each 10-word channel contained twelve 7-bit data characters (84 bits) per word plus a 7-bit space between each word for a total of 91 bits per word. A 7-bit data character of UNIVAC I was entered serially by seven 2.25 Mhz pulses from the Main clock into a 10-word channel at the sending piezoelectric quartz crystal associated with each 10-word mercury channel. This character data was also mixed with a 11.25Mhz carrier wave at the sending crystal. The character data moved through the mercury enclosed in anti-cross-talk tubes called wave guides and look somewhat like the examples below. When the character data was picked up by the receiving piezoelectric quartz crystal it entered circuitry that amplified and reshaped the data for another trip through the mercury. It also made the character data available to and from the High Speed Bus for computing use by the UNIVAC I. This circuitry took about 2 clock pulse times. The time to complete 1 trip of the 910 bits (840 for the 10-words plus 70 for the 10 spaces between these words) in the 10-words of a channel was 404  $\mu$ -seconds.

A 2.25 Mhz clock pulse source is shown below each of the 3 numeric characters that are in XS-3 code. Seven clock pulses examine each 7-bit data character through the circuitry on each 10-word channel as the bits exit the mercury. Visualize the shapes below moving to the right in 7 increments. When the shape is positive a clock pulse sees a 1, when negative a 0.

A zero (1000011)  a five (0001000)  and a nine (1001100)   
 2.25 Mhz clock pulse

### A Trip through the Computational Circuits:

UNIVAC I's arithmetic instructions operated within the range of +99,999,999,999 to -99,999,999,999 with an implied decimal point just to the right of the sign. The least significant bits of the least significant characters of the 2 numeric words being processed were presented to the computational circuits where they were examined and changed to produce the new numeric characters of the new word. This processing was consistent with the instruction being executed and in accordance with rather complicated XS-3 arithmetic (+ - \*  $\div$ ) rules.

The computational circuits briefly worked like this. Two 12-character numeric words involved in the addition or subtraction instructions would each be circulating in short tank mercury registers and their distribution delay line circuitry, rA and rX, of the rF, rL, rA, rX register set. A round trip through these registers was called a minor cycle, which was 40.4  $\mu$ -seconds. This was the time it would take to perform an addition or subtraction instruction. The processing took place in the distribution delay line of rA, which provided access to multiple bit positions and was required for developing the new numeric characters. Many strange and wonderful things happened during this processing, which included bit addition, bit subtraction, and 4-bit character complementation operations. Some of the work done by the computational circuits at this time included:

1. XS-3 base correction. Processing numeric bits of two XS-3 characters changed the base of the new character.
2. XS-3 carries. Carries to the next character position jump 3-bit positions including the parity bit.
3. Parity bit generation. Required for each new character developed.
4. Sign character generation. A plus was a 1110011 and a minus was a 0000010.

Activities like these really kept the 25L6 tubes busy opening and closing required gates. The result of this bit-by-bit processing produced the characters of the new word, which would be stored directly in rA right behind the characters in rA that still had to be processed. All this was done without missing a single clock pulse. Now you may be able to better visualize what I mean by strange and wonderful things. In UNIVAC I classes, this process was difficult to explain and to be understood. An example of a garment zipper usually helped with the zipper handle mechanism in operation being the computational circuits.

The multiply and divide instructions took many minor cycles since multiple processing operations were required but the processing was much like described in the previous paragraphs. Register rL was brought into play as one of the operands and the product and the quotient stored into rA where it's value was rounded. The divide instruction also stored an unrounded quotient in rX. One of the 3 multiply instructions (P) was used for a precision multiply and stored a 22 character product in rA and rX. The multiply instructions reduced the number of minor cycles by loading rF with the value of 3 times the multiplicand.

Each time this value was used saved 2 minor cycles.

If a XS-3 carry into the sign character position occurred during this operation the dreaded overflow light on the SC console would come on and the computer would stop unless the programmer expected this condition and programmed it to continue. If the computer stopped, the operator would then print out or record certain conditions displayed on the console like the instruction in the SR and the memory address in the rCC to help programmers recover from this and other error conditions. During the initial coding phases of a project, programmers were often present at the computer with coding sheets and flowcharts in hand. By watching their program run they could see if a tape moved, if a loop occurred, or if other conditions germane to their program happened. Debugging then was much easier.

### Circuit Operation and Trouble Shooting:

I mentioned the 25L6 tubes, three paragraphs above, because their plate rise times had to be extremely fast (.08 to .1  $\mu$ -seconds) when they changed states. Each word bit was .445  $\mu$ -seconds wide and represented either a 1 (90 volts) or a 0 (60 volts). These voltage levels were used to enable or disable gates probed by main clock pulses (.08 to .1  $\mu$ -seconds spikes of about 37 volts). Doing the arithmetic with these numbers show how very tight UNIVAC I tube circuits behaved in the 1950's. They were going as fast as they possibly could.

When plate rise times of the 25L6's became sloppy or too slow, a single error in one gate of the duplicated computational circuits would cause the checking circuits to immediately stop program execution and display a neon indicator with the general location of the error. This condition then would be isolated and observed with a dual sweep oscilloscope. We called this tube condition sleeping or cathode sickness and replaced it. Other reasons for failures in these gates, occurred when germanium diodes used for clamping the plate circuits of the 25L6's between 60 and 90 volts failed (opened or shorted). This made for a wild looking gate and was not good. Their replacement required a soldering iron. Conditions like this were often caught during maintenance while running "margins", explained in the heading "Cooling and Maintenance of UNIVAC I". Persistent circuit problems in a particular area could result in a Field Change Order, which usually tweaked an already functional but marginal timing circuit.

The dual sweep oscilloscope was especially helpful for isolating a component failure that stopped the computer. The computer, in effect, was only stopped from advancing to the next instruction of the program. All main-memory and register content continued to move in the delay lines and the circuits in play at the time of the error could be activated from the SC console and scoped. Comparing wave shapes at identical points in each of the duplicate circuits usually was sufficient to pin point the offending component or at least the immediate area of the component. The main reason for duplication of circuits in UNIVAC I was to provide error free results, but it also greatly reduced troubleshooting time. I still can visualize the giant circuit diagrams used for troubleshooting. They were about 3 feet wide by 2 feet high and there were a lot of them. They were laminated to extend their life and were mounted on a slant table with wheels.

### Programming and FLOW-MATIC:

Even by today's standards the UNIVAC I instruction set was quite advanced. Its 43 instructions are shown.

Twelve (B C F G H J K L V W Y Z) for loading, preparing, and storing registers.

Seven (A D M N P S X) for arithmetic.

Two (T Q) for conditional transfers.

One (U) for unconditional transfer.

One (R) for subroutine return.

Four (. : - 0) for shifts.

One (E) for masking.

One (00) for skipping.

One (90) for stopping.

One (10) for typing one word into long term memory from the SC keyboard.

One (50) for typing one word from long term memory to the SC on-line typewriter.

One (,) for a breakpoint stop.

Ten (1 2 3 4 5 6 7 8 30 40) for input/output, including backward reading from UNISERVOs.

Some experienced programmers who worked with other later computers said UNIVAC I was the "Cadillac" of the early computers. The UNIVAC I instruction set from a programmers code card is shown in table form below.

Edward Teller, in his memoirs, used UNIVAC I to do calculations to complete the details of a weapon design. These are his words.

*...Sid [Fernbach] and Chuck [Leith, Jr.] were working on the computer program before the [Livermore] laboratory officially opened. They had gone out to the Eckert-Mauchly plant in Philadelphia to oversee the final development of the first computer, a Univac (Universal Automatic Computer), one of the first computers produced commercially. I joined them there to try out the Univac while it was still on the factory floor. ...*

*...The calculations we wanted to do were needed to complete the details of Livermore's first weapon design. The Univac performed wonderfully until shortly before the work was completed. Then the computer broke down so thoroughly that there was no hope of our using it to complete the calculations in a timely manner. We had nothing but pencil and paper to assist us in that task, which sufficed...*

*...The value of the Univac, which arrived five months after the laboratory opened, can hardly be overstated. But a little less than four years later, someone tallied the operations that had been completed on it. They were equivalent to the number of calculations that 440 people, working forty hours a week, fifty-two weeks a year, without any vacation or breaks, could have completed in a hundred years. ...*

UNIVAC I (serial #5) was delivered to California's Livermore Laboratory in April 1953. It was ready to be delivered in November 1952 but was delayed because it was the UNIVAC I computer that was programmed by a statistician from the University of Pennsylvania to predict Eisenhower's landslide victory in 1952. The program was written to sample early returns of certain precincts. CBS did not believe this presidential prediction because they thought Adlai Stevenson II would win. CBS finally said that UNIVAC was right and put that out on the air waves.

In 1955 UNIVAC I produced the first concordance of the Revised Standard Version of the Bible. Two copies of the entire 800,000-word text were made. One by five people using Unitypers that produced UNISERVO tapes. The other by ten people punching cards that were copied to UNISERVO tapes by the UNIVAC card to tape converter. This preparation process took about nine months. UNIVAC I compared these tapes and corrections were made and the final set of UNISERVO tapes were ready for the concordance run.

Then UNIVAC I started the concordance run, which finished in three hundred hours. The actual computational processing by the computer was a relatively simple task. It was an I/O intensive program because of the limitation of the 1000 words of UNIVAC I memory. This required more UNISERVO tapes to be produced for recording the key-words and their references. Then UNIVAC I sorted those tapes and produced more tapes that were printed by the High Speed Printer. Typesetting by Nelson's publishing took about two years to complete and the concordance was published in 1957. A similar concordance published in 1894 had taken thirty years to complete before this one was done by UNIVAC I.

I recently bought a copy of this massive 2,157 page concordance from [abe.com](http://abe.com). Here are some of the words said about the preparation of this volume. This is in the preface, which was written by John W. Ellison on Whit Monday, 1956. John was the Episcopal priest who envisioned and supervised this work.

*...To distribute the work among a group of scholars would be to run the risk of fluctuating standards of accuracy and completeness. The use of mechanical or electronic assistance was feasible and was at hand. The UNIVAC I computer at the offices of Remington Rand, Inc., was selected for the task. Every means possible, both human and mechanical, was used to guarantee accuracy in the work. ...*

*...The editor takes this opportunity to express his personal gratitude for the unqualified cooperation which he received both from the publisher, Thomas Nelson and Sons, and from the computing staff of Remington Rand, Inc., in the many difficulties, complexities, and delays in this pioneering work. ...*

I can still remember the three-instruction set called BAC. B brings memory word (m) to register rA and rX, A brings memory word (m) to register rX and then adds rX to rA, C copies rA to (m) and clears rA. The designers of the instruction set assigned alphabetic letters that corresponded with what the instruction actually did. I believe this also began or accelerated the thought process that led to higher-level language development.

Rear Admiral Grace Hopper (1906-1992) admirably called Grandma COBOL worked with UNIVAC programmers to develop the English-statements of the FLOW-MATIC language. Then they wrote a UNIVAC I program called a compiler that would produce the UNIVAC I program codes from the English-statements of FLOW-MATIC. When the output of the compiler program was executed it would do what ever the FLOW-MATIC English-statements specified. FLOW-MATIC was the forerunner of the Common Business Oriented Language know as COBOL. I have included two photos of her work with FLOW-MATIC and COBOL.



She retired from the Navy in January 1967 and was recalled back to active duty in August 1967. She then retired again in August 1986 and was honored by the U. S. Navy in 1997 for her pioneer work in applied computer science by commissioning a new guided missile destroyer called the USS Hopper (DDG-70). She remained associated with Remington Rand, Sperry Rand, and UNIVAC for many years. I remember one such occasion where she gave her presentation on the nanosecond. She was always interested in the speed of computing and how programmers could write more efficient code and faster. She used a small coil of wire and emphasized that a length of just under a foot of it was the time a nanosecond would travel through it at the speed of light. She was proud of wearing her Navy uniform and was always impeccably dressed.

The COBOL language has stood the test of time and is still widely used. Jim, one of my friends at Northwest Airlines says he loves the COBOL language and uses it all the time. His wife, Rose, used it at USPS for several years recently. This makes me feel good since I was on the Committee on Data Systems Languages (CODASYL) where it was developed and refined.

The CODASYL committee was made up of members from various computer corporations and interested users that wanted to set standards for computer languages so that they could be more easily moved between different computers. COBOL was a main benefactor of this. There are literally thousands of older COBOL programs still running today.

### Subroutines and Libraries:

The UNIVAC I, although designed for business use, was used in both the business and scientific fields. Floating point arithmetic was not supported by the UNIVAC I instruction set but could be programmed using a combination of instructions constructed as a subroutine. Subroutines, called subprograms in those days, were supported by the R instruction. R recorded the main program return address into the subroutine exit point before leaving the main program to execute the subroutine. Subroutines were often part of libraries available for use by all programmers. This forward-looking concept is still widely used in modern computers.

### Cooling and Maintenance of UNIVAC I:

The following description is from Ed Lankford's understanding of the refrigeration system of UNIVAC I. Ed worked on UNIVAC I longer than anyone and knows everything about it. Ours at WPAFB and the one that Ed worked on in Nashville were in the same room as the UNIVAC I to keep the refrigeration system from freezing up in the winter.

This is the start of 4 paragraphs of Ed Lankford's input.

*A huge refrigeration unit called a CentraVac was used to keep UNIVAC I cool. It had a compressor turbine with a diameter of about 3 feet and stood about head high. There were pumps for circulating the water through the system and a condensing tower like you would have seen on top of buildings back then. There was a venting system to outside of the building for the fan pulling the air through the tower when called for by the system.*

*It was not uncommon to have firemen burst into our computer space asking "where's the fire" in the coldest weather of winter. Someone spotted the cloud of warm water vapor being blown out into the cold air on the alley and called the fire department! This generally happened on the midnight to eight shift.*

*According to one article it was described as being a 54 ton unit, probably more likely a 50 ton. Since the UNIVAC II had basically the same cabinetry, I would assume the cooling system was the same for the UNIVAC I.*

*The UNIVAC I had copper heat exchangers around the base with the chilled water flowing through and two large fans under the sub-floor inside the box to circulate the air over the chilled heat exchangers and outward and upwards over the hot tubes on the chassis in the bays and back inside and down to the fans again. This did cause some condensation at times and it seemed to want to collect in H-J corner that was probably the lowest point. This condensation caused oxidation on the Silver Plated contacts on the chassis and blackened them. When it dried out it could cause poor or high resistance connections which caused problems; mostly in the memory sections. To combat this Univac Factory Engineers set up a system to replace the 'backboards' that contained the male contacts with Gold Plated contacts. There were also 'clips' that could be placed on the female 'spring' contacts (after cleaning them) and using a electrical tool to 'weld' the pre-tinned clips onto the old contact to make a gold to gold contact surface. This largely solved the problem. We did the memory and some of the more critical chassis like cycling unit and timing pulse generators and such others that we found problems with.*

This is the end of Ed Lankford's input.

The cooling system was provided with a number of checks to determine and regulate its proper operation.

1. A cooling system interlock switch was connected into the main interlock circuit of the power system. This interlock prevents the application of power to any unit unless the cooling equipment is turned on.
2. The mass of the cooling equipment is sufficient to take care of limited demands for cooling, thus it provided modulation on the operation of the large compressor system.
3. The coolant was monitored for flow, temperature and pressure by gages in the intake and outlet of the system.

For what it's worth, the UNIVAC II dissipated about 115 Kw of electrical power that was 10 Kw less than UNIVAC I and required approximately 35 tons of refrigeration. This system required approximately 158 gallons of 59 degree Fahrenheit water per minute with an 80-foot head loss. The recommended maximum pressure is 45 psi, with 50 psi the absolute maximum. Because condensation does form on the heat exchangers a condensate drain is provided on the water-cooled sections of the equipment. Air conditioning the area in which the computer is located is not normally necessary; however, it is often done for the comfort of operating personnel and it minimizes condensation in the units.

There was a ripcord in the UNIVAC I that you pulled to remove power from the UNIVAC I in the event you developed an emergency while on maintenance, like removing your leg from one of the two large and powerful cooling fans before they amputated it. There was a fair amount of danger while working inside "the box" with high voltage and high drain current circuits present in abundance. You never wanted to trip because then you would be in big trouble while grabbing on to something.

Preventive maintenance was extremely important to reduce system down time. One of the steps taken to check the health of the system was running "margins". The filament voltage of the electronic tubes was reduced slowly in stages as the computer ran a diagnostic program called "Ansero". The time it took until system failure indicated rather decisively the system's health. The failure point then was often isolated and corrected, usually with the aid of an oscilloscope. Then "margins" were run again. Many of these systems ran around the clock and included scheduled maintenance periods. One of the interesting scheduled tasks was checking selenium rectifiers. This was not done often, but it was somewhat of a big deal when it was. The top of the computer was removed and we would climb up and walk around the top where these large protective rectifiers for the 60 and 90 clamping voltages for the 25L6 tubes were located in their respective bays. It was a dirty job because they were usually covered with dust. The forward and reverse resistance of the rectifiers were tested with ohmmeters and replaced as needed. Other rectifiers located in the power supply cabinet were checked the same way.

Ed, related to me the dirtiest job he ever did was to replace the lower cooling fan bearing in the power supply. This was a major job because of the close quarters and all the other electrical components in the way. It was a dirty and greasy job.

The figure of 90% up time was the target goal for UNIVAC I. Some installations achieved in excess of that number. There have been instances where an original 25L6 tube lasted for the life of the computer. The computers at Nashville L-C Ins, according to Ed, could often run "margins" without error. Paul Lawson felt good when that marginal report came in.

If anyone has the sound of UNIVAC I running the "Ansero" diagnostic program please send it to me so I can put it in this web page. It would bring this web page to life. I should have made a copy of that sound but I did not. Of course, there was no convenient way to record sound in those days anyway.

**UNIVAC I Instruction Set**  
(from a UNIVAC I code card)

<p><sup>1</sup> A m (m) → rX, (rX) + (rA) → rA</p> <p>B m (m) → rA, rX</p> <p>C m (rA) → m, 0 → rA</p> <p><sup>1</sup> D m (m) → rA, (rA) ÷ (rL) → rA (ROUNDED) → rX (UNROUNDED)</p> <p>E m rF EXTRACTS (m) → rA</p> <p>F m (m) → rF</p> <p>G m (rF) → m</p> <p>H m (rA) → m</p> <p>J m (rX) → m</p> <p><sup>2</sup> K m (rA) → rL, 0 → rA</p> <p>L m (m) → rL, rX</p>	<p>Y m (m THRU m + 9) → rY</p> <p>Z m (rY) → m THRU m + 9</p> <p><sup>2</sup> 00m SKIP</p> <p><sup>2</sup> .nm SHIFT (rA) RIGHT WITH SIGN n PLACES</p> <p><sup>2</sup> :nm SHIFT (rA) LEFT WITH SIGN n PLACES</p> <p><sup>2</sup> -nm SHIFT (rA) RIGHT EXCLUDING SIGN n PLACES</p> <p><sup>2</sup> 0nm SHIFT (rA) LEFT EXCLUDING SIGN n PLACES</p> <p><sup>2,4</sup> 1nm 60 WORDS FROM TAPE TO rI, FORWARD</p> <p><sup>2,4</sup> 2nm 60 WORDS FROM TAPE TO rI, BACKWARD</p> <p><sup>4</sup> 3nm (rI) → m THRU m + 59, 60 WORDS → rI, FORWARD</p>
--	--

<p><sup>1</sup> M m (m) → rX, (rL) x (rX) → rA (ROUNDED)</p> <p><sup>1</sup> N m -(m) → rX, (rL) x (rX) → rA (ROUNDED)</p> <p><sup>1</sup> P m (m) → rX, (rL) x (rX) → rA rX (22 DIGITS)</p> <p><sup>3</sup> Qnm IF (rA) = (rL), TRANSFER CONTROL TO m</p> <p>R m !000000 U00 (CC)! → m</p> <p><sup>1</sup> S m -(m) → rX, (rX) + (rA) → rA</p> <p><sup>3</sup> Tnm IF (rA) &gt; (rL), TRANSFER CONTROL TO m</p> <p>U m TRANSFER CONTROL TO m</p> <p>V m (m) (m + 1) → rV</p> <p>W m (rV) → m, m + 1</p> <p><sup>1,2</sup> X m (rX) + (rA) → rA</p> <p><sup>1</sup> - in second digit stops computer on overflow</p> <p><sup>2</sup> ignore m</p> <p><sup>3</sup> stops computer if conditional trans switch is on</p>	<p><sup>4</sup> 4nm (rl) → m THRU m + 59, 60 WORDS → rl, BACKWARD</p> <p><sup>4</sup> 5nm (m THRU m + 59) TO TAPE via rO, 128 PLUSES/INCH</p> <p><sup>2,4</sup> 6nm REWIND TAPE</p> <p><sup>4</sup> 7nm (m THRU m + 59) TO TAPE via rO, 22 PLUSES/INCH</p> <p><sup>2,4</sup> 8nm REWIND TAPE, SET INTERLOCK</p> <p>10m SUPERVISORY CONTROL KEYBOARD → m</p> <p>30m rl → m THRU m + 59</p> <p>40m rl → m THRU m + 59</p> <p>50m (m) → SUPERVISORY CONTROL PRINTER</p> <p><sup>2</sup> 90m STOP UNIVAC</p> <p><sup>2</sup>, m BREAKPOINT STOP</p> <p><sup>2</sup> ignore m</p> <p><sup>4</sup> n represents UNISERVO number (1 thru 9, -)</p>
--	---



**Robert Stark, Thomas Robinson, Frank Smith, George Miller, and Kevin Leenay.**

This is the front of UNIVAC I (serial #10) at Wright-Patterson AFB near Dayton, Ohio. Tom, Bob, and John Spengler (not shown) each were in charge of this installation. I worked with these gentlemen from 1955 to 1957 and several others whose pictures are not available.



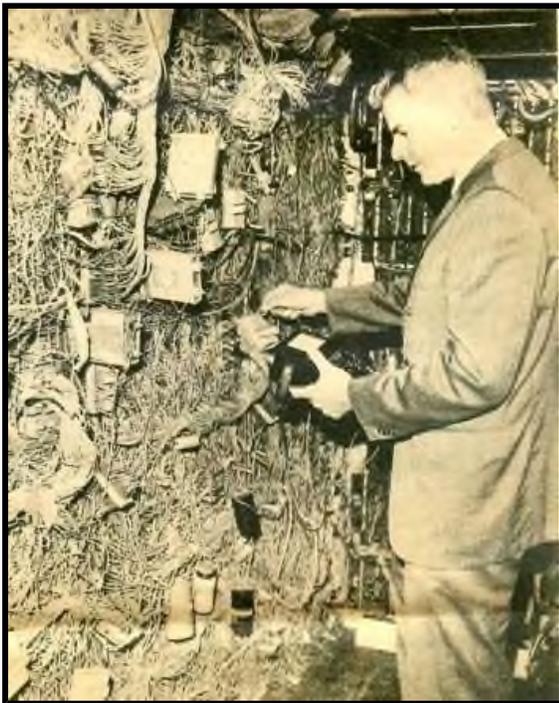
**Bob Stark conversing with associate.**

The computer shown here is the prototype of the UNIVAC II, which was built in St Paul, MN from a UNIVAC I. It was located on Prior Ave, just south of West Minnehaha in St. Paul and was called Plant 5. My late wife, Mildred, was a programmer on this UNIVAC II computer. The new magnetic core memory was suspended about the computer since it was a prototype. UNIVAC II continued to use the short tank mercury memory of UNIVAC I. The SC consoles look different too. Compare this white one with a photo of a UNIVAC I console that was black, shown a few photos below. 28 UNIVAC II's were built.



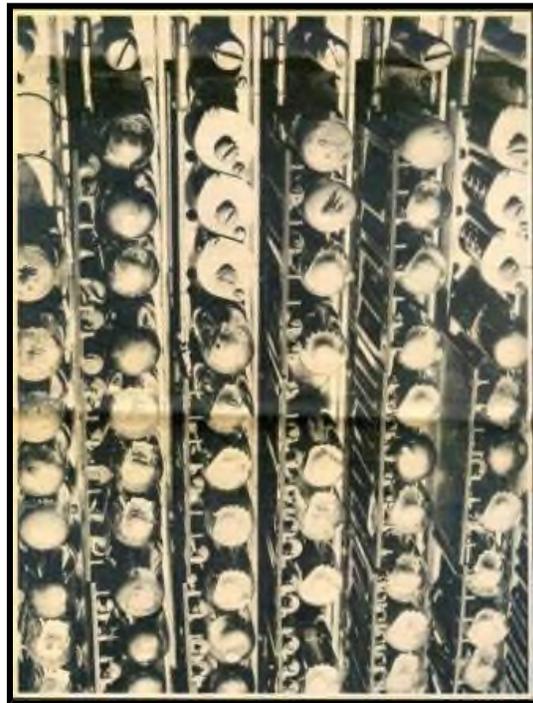
**Bob Stark, Arlin Fraedericks, Curt Leeper, Bennie Muffett, and Allan Reiter.**

This photo was taken in 2001 at Red Wing, MN. I worked with Bob, Arlin, and Curt at WPAFB in 1955-57. Bob is also shown on other photos.



**Rollie Arndt inside the computer.**

Every wire has to be connected to the proper place. It took two beam pentode tubes (25L6's) and associated circuitry to represent a single binary 1 or 0 state. We called this circuit configuration a flip flop. The weight of cabling for each computer was in the tons and more than half of the total computer's weight.



**Thousands of electron tubes, mostly 25L6's.**

They had 25-volt filaments and produced enormous amounts of heat. The plate circuits were clamped with germanium diodes between 60 and 90 volts to provide for consistent gate signal timing requirements. Giant cooling systems and bay fans were required for cooling. Doors on each of several component bays had to be kept closed during operation to avoid over heating.



**Bob Stark and Rollie Arndt discussing magnetic tape.**

The recording media was metallic tape and each reel weighted about 5 pounds. To protect the Read/Write Head, Mylar tape was slowly advanced between the Read/Write Head and the metallic tape. Recording density was normally 128 characters per inch and tape bad spot skipping was performed using holes punched in the metallic tape.



**Type of tubes used in UNIVAC I systems.**

Left tube: 829B, used in the Master Timing Pulse generator circuits. Right tube: 25L6, the electronic workhorse of the computational circuits.

Other tubes used: 3C23, 4D32, 5545, 5651, 5687, 5R4, 5V4, 6AH6, 6AK5, 6AL5, 6AN5, 6AU6, 6BE6, 6SN7, 6X5, 7AK7, 807, 28D7, 2050, and 8008.



**Business people attending a UNIVAC I training course.**



**A UNIVAC I at work at the Census Bureau.**



**The FLOW-MATIC manual.**

It described the English-statements used to program UNIVAC I and II in a higher language that was the forerunner of the COBOL language.

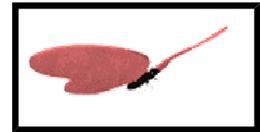


**Grace Hopper using COBOL on the UNIVAC II.**

These are the newer UNISERVOs. You can see the two vacuum loops that replaced the two-pulley arrangement used on UNIVAC I.



**UNIVAC I (serial #37), April 1964, Life and Casualty Ins. of Tennessee in Nashville.**



**In memoriam  
Paul S. Lawson  
1931 - 1984**

Paul S. Lawson, my buddy from Indiana, is at the side of the SC console. The operator is Larry Bradford. If I had to guess what he is doing I would say he is booting a program from a UNISERVO tape into memory. His left hand is setting the IOS switch to the "One Instruction" position. His right hand is setting the Static Register (SR) to the number of the UNISERVO that contains the program. Then he will hit the Start Bar twice

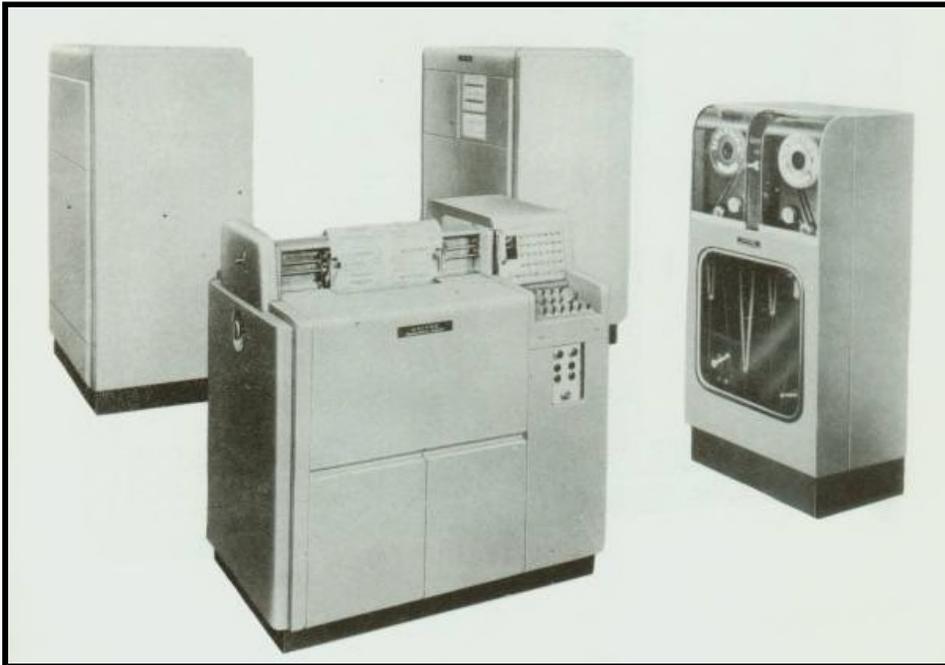
to read the program into memory. The Start Bar is located just below on the SC console. I have detailed instruction on how this booting procedure was done shown in the heading "Starting and Operation". The person shown using the phone is Holly Baker. He was an overall expert on UNIVAC I including the Unityper.

Looking carefully at the SR you can see the neon lights and switches for its 27-bits. The right most grouping of three 5-bit characters contained the number of the memory address, each 4-bit numeric character plus its parity bit. The next 5-bit group he is working with is the second instruction character, a 4-bit numeric character plus its parity bit. This character position can be used to select one of the 10 UNISERVOs (1 thru 9, and -). The left most 7-bit character, showing all 7-bits, is the actual instruction character, A, B, C, D and so on. Operators were required to memorize the UNIVAC I instructions and XS -3 character codes.

The use of the - character (000010) for selecting the 10th UNISERVO is an example of using 1 of the three characters of the XS-3 code set that had a value of less than an XS-3 zero (100011). The other two were non-printable characters, the space character (000001), and the i character (100000).

The use of the - character for the 10th UNISERVO was necessary because the UNISERVO selection was restricted to the 2nd instruction character of SR, which contained only four bits plus its check bit and could only be counted up to 9. It was thought that 8 UNISERVOs would be all that were needed in the UNIVAC I. But advances in programming techniques were really moving in those days, especially in the area of sort algorithms where additional UNISERVOs were required to increase the number of sort strings for faster sorting. In fact, UNIVAC II expanded UNISERVOs to 16.

The only character of the complete 64 character XS-3 code set not used was the all ones (1111111). The electronic circuits were overdriven with this combination of bits. Another bit series that presented similar problems was flooding the main-memory with a combination of Z4Z4 characters (1111100000111111100000111). Once this was discovered mischievous programmers could bring UNIVAC I down. Probably the first incident of computer vandalism now so rampant in modern computers. Super programmer Peter Zilahy Ingerman, PhD has incorporated this arcane UNIVAC I feature into his Simulator. It brings the Simulator down.



**The High Speed Printer (600 lines per minute). Added to the UNIVAC I System in 1954. Cabinets from left to right: Power supply, Printing, Control and Checking, and Tape.**

Customer applications of UNIVAC I were producing voluminous amounts of data that needed to be printed, many requiring multipart forms. The Uniprinter was basically a ruggedized typewriter with a tape drive and not up to the print speeds required. This High Speed Printer (HSP) was really needed. Data tapes to be printed were produced in a special 120-character printer blockette format and brought to the High Speed Printer for processing. The mechanics and electronics for this device were rather amazing.

1. A 130-column width heavy rotating printwheel with 51 printable characters engraved in each column in a

- odd even staggered manner. For example to print all the A characters in a line took two print cycles, one for the odd characters and one for the even. It was designed this way because the hardware could only print up to 65 characters in one print cycle and none of these 65 characters could be next to each other.
2. 130 print hammers (spring return actuators, four shapes to accommodate tight fitting requirements).
  3. 130 tubes (2050 thyratrons), when fired, drove the hammers to print the characters in the blockette of tape data.
  4. Multiline printing and zero suppression control via a plug board.
  5. The printing mechanism produced enough force to print multipart forms, up to 7 parts (original and 3 carbons).
  6. Computer Digit printing. For testing, 9 non-printing HSP control characters could be forced to identify themselves on a separate line just below the line that contained them.
  7. Fast feed paper stop control by punched paper loops for various output forms. Fast feed advance was initiated by operator switch for positioning paper and by 4 special non-printing characters placed in the 1st character of the blockette by the programmer.

Seeing and hearing this valuable addition to the UNIVAC I System in operation was quite a sight to behold. I enjoyed watching the bay of 130 thyratrons fire as they followed the order of the data being printed. These gas thyratrons cycled on and off with a sort of blue-white glow. Changes in the firing pattern would be accompanied by changes in the sounds of printing. WPAFB had applications that kept the printer running for hours. The only time it seemed to stop was when it needed another data tape mounted or when it ran out of paper that came in large industrial sized boxes.

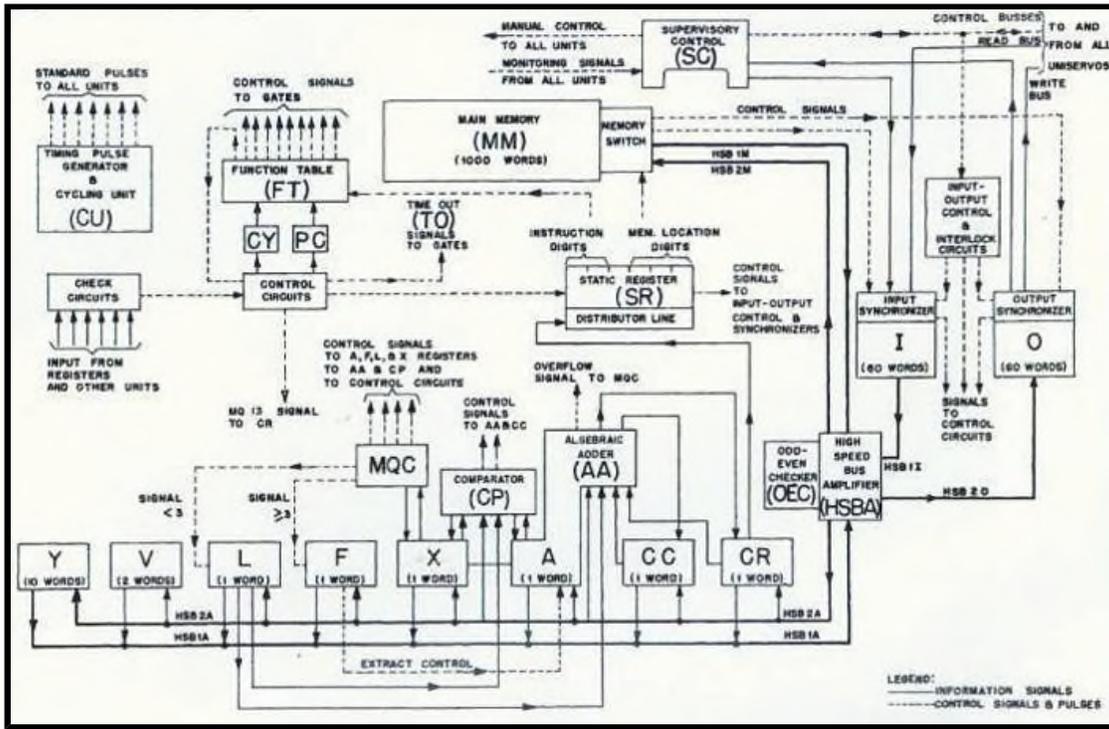
Peter Zilahy Ingerman, PhD has recently produced a Simulator for the HSP, which is part of the UNIVAC I Simulator. I have used it and found it to be a marvelous piece of work. It can print the tapes that the UNIVAC I Simulator produces and has all the features and more of the HSP including Normal printing, Computer Digit printing, Zero suppression, and Multiline printing. Since a plug board and a carriage control paper loop were used by programmers in the printing process he has made a brilliant simulation of these rather complicated features. I used the term "and more" because he added a feature that creates a Windows txt file so it can be read into the UNIVAC I and used as data. The txt file must follow the rules of the character set used by UNIVAC I, this means it must use the 26 capital letters of the alphabet used by UNIVAC I. Here is a line of the 51 printable characters that were engraved on the HSP printwheel.

**ABCDEFGHI#JKLMNOPQR\$\*:/STUVWXYZ%-0123456789'&(&#x27E8;#x27E9;**

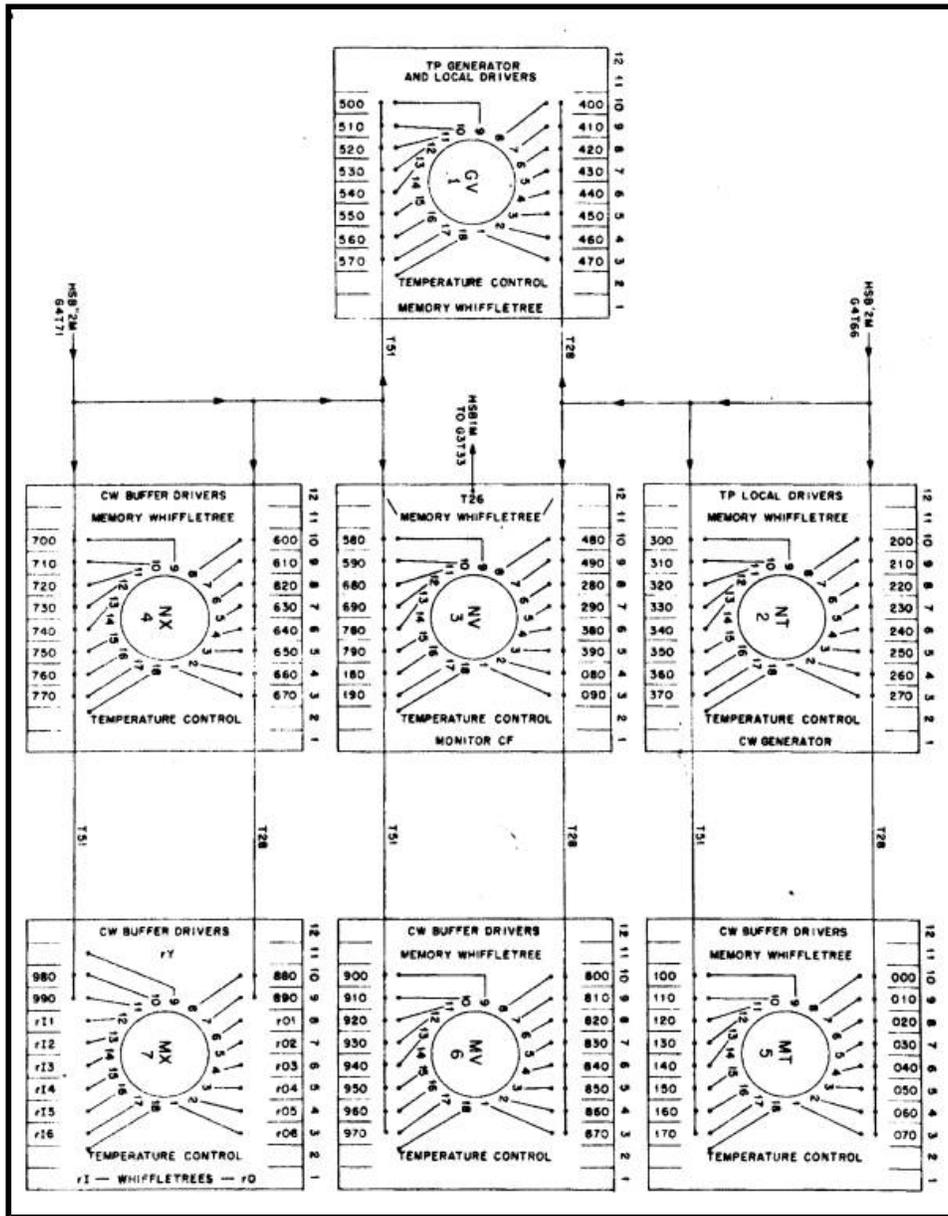
There were 12 non-printable characters that performed special functions of the 63 codes UNIVAC I recognized. The HSP used 9 of these non-printable characters; 4 for Fast Feeds FF1 (0011111), FF2 (1100010), FF3 (0101111), FF4 (0111110), 1 for Multiline printing (0010000), 1 for spacing (0000001), 1 for optional breakpoint stop (0110001), 1 for printer stop (1110000), and 1 that was ignored (1000000). Setting the HSP to print in Computer Digit mode, which was used for testing, would identify these 9 non-printable characters. Then the line that contained these non-printable characters would be printed on the following line as printable characters. The FF1 would print as a D, FF2 as a P, FF3 as a M, FF4 as a T. Multiline printing as a E. Spacing as a 6. The optional breakpoint stop as a W and the printer stop as a V. The ignore as a 5. Spacing can be used by the Windows space key.

I have included a txt file from the HSP Simulator that you can download and look at. It includes two pages of output that shows how the 4 Fast Feeds and Zero Suppression worked. The third page shows the precessing alphabet. The precessing alphabet was used for testing HSP horizontal print alignments. One or more of the 130 actuators that printed the line would occasionally wear out and had to be replaced. See the txt file [hspoutput](#). Return here by using the browser back function. Down load the UNIVAC [Simulator](#). You will need the "extensive documentation" version if you are planning to do plug boards, carriage control paper loops, and Multiline operations.

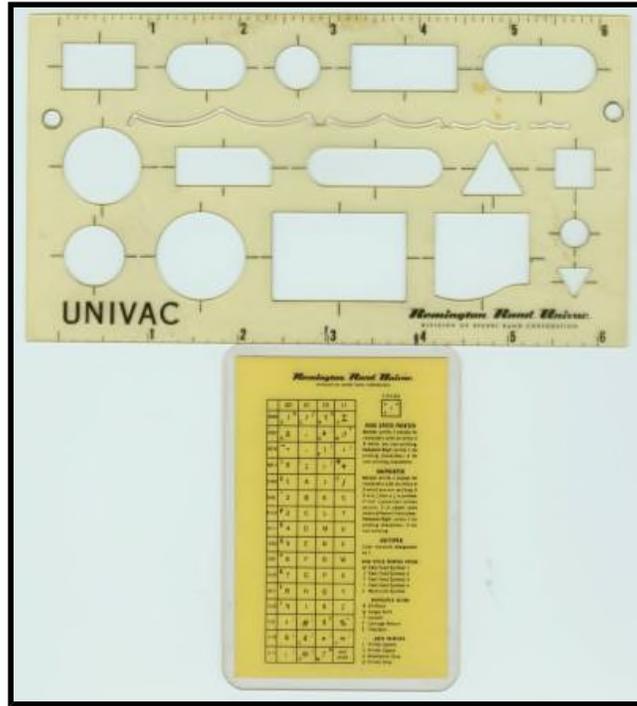
**The UNIVAC I Block Diagram.**



The 7 Long tanks Memory Whiffletree.



UNIVAC Programmers Tools.



After looking at this blue page go to the [yellow](#) page to read about some of the people who worked with UNIVAC I.

Trademarks: Unisys, UNIVAC, and UNISERVO are registered trademarks of Unisys Corporation.