Ken Romain Knowledge Dump

Remember only the MUX card with the crystal oscillator near the 4th UART chip. 4 position dip switch must be set for MUX board 0 meaning at I/O address HEX- F200-F20F. The second MUX board Ports 4 - 8 are at F210-F20F.

The 1702 EPROM on the DSK/AUTO card is disabled then the p/n 1001 DSK/AUTO card is used in a CPU5 or CPU6 system, should be a etch jumper short near the 1702 EPROM. The 1702 EPROM was the BootStrap program for the CPU4 system and only supported booting from the 9427H. The CPU5 & CPU6 PROM on the backplane lets you boot from the 9427 Hawk , 9448 CMD or 8" / 5-1/4" Floppy Disk. The CPU5 / CPU6 boot PROM could send the { ERROR } message to the CRT on MUX Port-0 @ HEX F200. The 1702 EPROM used on the CPU4 could not send any error messages, the system just booted or it did not boot at all.....!

The CPU6 card had to have read the text string [ERROR] from the backplane boot PROM and then wrote a number of bytes to HEX F200 - F20F [ MUX - 0 base I/O address ]. 1st. - setup WRITE - MUX Port 0 control reg. byte value for 9600,8,N,1 (should be a single byte write , do not recall the byte value it's been 40 years). 2nd - READ - MUX Port 0 status reg, for CTS and DSR status bits (should be a single byte read). 3rd - WRITE - MUX Port 0 the ASCII byte value {like E} to UART on port 0 to be sent out port zero to the terminal. Then keep repeating 2nd and 3rd steps till the other ASCII values for { R R O R } are sent to the terminal. Terminal then show display the ASCII data sent out MUX Port 0 at I/O base address HEX F200 - F20F. In this case { ERROR } should be on the screen. I think the reg. mapping for the first 4 Port MUX card is something like this: HEX I/O Address F200 - F203 are Port-0 Write-ControlReg byte, Port-0 Read-StatusReg byte, Port-0 Write-XmitData byte, Port-0 Read-RecvData byte. Then HEX I/O Address F204 - F207 are Port-1 Write-ControlReg byte, Port-1 Read-StatusReg byte, Port-1 Write-XmitData byte, Port-1 Read-RecvData byte. The I/O mapping repeats for Port-2 @ F208 - F20B then Port-3 @ F20C - F20F .

I recall the CPU crystal being 5 MHZ and the cycle being 200 ns. The CPU5 & CPU6 (AMD2901 bit slice) only used the backplane PROM to bootstrap the system. Do not recall any bootstrap code on the CMD card or other cards other than the old Disk-Auto card of the first Warrex computer call the Centurion CPU4 ( all TTL & ECL chips ). CPU4 = CPU1, CPU2, CPU3 and DMA card with bootstrap EPROM (oldest rev) or two PROMs (newer rev) on the Disk-Autoload card. You may have noticed the metal Centurion card cage slots 1, 2, 3 & 4 missing some plastic card guides. The same card cage was used for the CPU4 systems just change the backplane, had cards in slots 1,2,3 CPU bus & 4 DMA - I/O bus and no logic on the backplane. For the CPU5 & CPU6 the backplane had the bootstrap logic & PROM (left of CPU slot, slot-2) and no card guides in slots 1 & 3 all the other slots are just I/O - Memory bus slots. You may also notice two jumper wires on the backplane between slots 2 & 4 (no connector 3). This is because the CPU5 card was to wide (stacked cards into single slot , design issues). The two jumpers are to pass the serial interrupt signal pass the missing slot-3 connector. Peter, did you happen to see a old post explaining the I/O Mapping, 16 Interrupt Levels and System Memory layout? Key points are the system is 8 bit ALU & Registers with 16 bit PC (program counter) and 16 Interrupt Levels 0-F. The first memory addresses area, the CPU registers 0000-000F for Interrupt Level-0. By adding the Interrupt level to the register based address to you are at that interrupt level like 0040-004F is interrupt level-4 and 0050-005F is interrupt level-5 and 00F0-00FF is interrupt level-F. All the systems I/O cards (and boot strap PROMs) are between F000 & FFFF ( like the Hawk disk controller is a F140-F14F & MUX-0 is F200-F20F ). The first program storage was written into HEX 0100 and the top end of RAM was EFFF for the CPU4. ( The CPU5 & CPU6 used the same opcodes and basic memory design but added paging of 60K byte RAM blocks to get pass the 16 bit PC and memory limits. ) The old DMA card ( Direct Memory Access card for the CPU4 ) was also in the F000-FFF space forgot the address, sorry. The DMA function is part of the CPU5 & CPU6 card. Key DMA points are: DMA was only used for hard disk Read & Write functions. The disk sector size is 400 bytes ( HEX - 190 bytes ). HEX - 3B2F sticks out for some reason in my mind. A normal DMA disk I/O was something like this.... the DMA was pointed to a memory address to move data to or from, the DMA byte count was setup FFFF max count less HEX - 0190 (one DEC - 400 byte sector), each DMA i/O read or write increase the memory address and the DMA byte count till the DMA byte count = FFFF and the DMA operation is over. The Warrex opcode ( 01 = no-op or one CPU cycle ), opcode ( 0E = delay the stops the CPU System clock for 4.5 ms and even stops DMA functions ). Having a 0E opcode execute at anytime during a disk read / write DMA operation with cause disk data corruption... the disk drive does not stop spinning to wait for the 0E 4.5 ms delay. This 4.5 ms delay stops the DMA also and can cause data wrong data to be read into memory or written into the wrong disk sector. ((( This was a real bear to figure out the OE opcode and disk data corruption bug. )))

No passwords, just the Radio Shack AC Power Switch key on the front panel which David picks in about 2 seconds. On much a later version of the @OSN Centurion operating system we did have a optional password to format the disk platters but rarely used.

They very first spin of the CPU4 had the 256x8 bit CPU registers that are bytes 0000 - 00FF in core memory. The system normally had two 4K Byte core memory card each two slots wide. The total of 8KB of core mapped like this: 0000-00FF (16 sets of CPU4 registers of which 14 of the register are 8 bit wide ALU & Status with the PC being 16 bits with. One set of registers per interrupt level 0-F. ) 256 bytes of static RAM cost thousands of dollars in 1974 & core was cheap. Memory 0100 - 2FFF = Opsys & program memory . Memory Mapped I/O registers F000 - FFFF ( each I/O card only needed a hand full of address ). So the max Opsys & program memory of only about 60KB. (16 bit PC = 65K less 4K I/O less 100 bytes CPU register for total of 60K bytes of RAM.) I think the most core memory we ever shipped in a customer's CPU4 system was 12K bytes. Later CPU4 systems had a 32KB DRAM card made up of 32chips 4K x 1bit TI DRAMs and two 32KB DRAM card max per system still only let you address 60KB of DRAM. The I/O cards at F000-FFFF had a address "Preempt Signal" that stopped the upper (2nd) 32K DRAM memory from responding the Read / Write cycles in the I/O range. Also by this time the newer rev CPU4 cards had high speed SRAM 256x8 registers in the CPU and used the "Preempt Signal" to kill memory cycles to the 1st 32K DRAM memory address 0000 - 00FF and only allow read / write to address 0100 - 7FFF or 8000 - EFFF. When the CPU5 was designed to run the same opcodes of the CPU4 with the DMA card logic now in the double wide CPU5 board stack ( two cards stacked in plugged into a single backplane slot ). The CPU5 added a few opcodes to lay the ground work for memory page swapping really put to use in the CPU6 about a year later. With the single board CPU6 and new Opsys code we could address 16 memory pages of 60K bytes ( still have to account for the 4KB of I/O space in each 65K memory block ). Most CPU6 systems had one128K or two 128K parity enabled memory cards and two to four 4-Port serial MUX cards ( 4 to 16 terminals attached meaning 4 to 16 users ). When we tried to run 16 to 32 or more users the system got very very very slow. Doing some digging the problem was found that all the system software written in CPL expect all the terminals to be the older ADDS model 580. When the currently shipping ADDS terminals like the model 520 and ADDS ViewPoint are used the Operating System had to load "terminal pre-drivers" into system memory based on the terminal model attached to a give 4-Port MUX card to adjust cursor position control codes changes between the ADDS terminal models over the years ( burning up CPU cycles). In order to get more than 16 terminal to run well on the larger CPU6 systems we designed a new 16-Port MUX again using the AMD 2901 bit slice ALUs (just like the CPU5, CPU6 and CMD disk controller). This very smart ADM2901 based16-Port MUX let the Opsys load the terminal pre-driver control software directly into the 16-Port MUX and off-load the task from the CPU6....! Now we could run 32 terminals at a reasonable speed with a 8 bit ALU running at 5 MHz and a 10 year instruction set doing memory mapped I/O, not to bad...! Then the IBM PC hit the market with Lotus-123, Word Perfect and QuickBooks.... bye bye to the Centurion minicomputer and many the minicomputers of the 1970s-1980s-1990s... ( Wang, NCR, Lition, Katoe, Computer Automation, Data General and even DEC years later ). Centurion really did not have many problem with 4K or later 16K DRAMs. I did burn up (32) 4K TMS 4060 chips checking out the 2nd proto type 32K DRAM card we ever build. The DRAM Refresh timer design was wrong and doing DRAM refresh at a 200 ns rate and not at 2 ms rate. All 32 chips let the magic $10,000 smoke out even before I picked up the 465 scope probe, very bad day for a small startup company in Richardson, TX. At this time in the early - mid 1970s DEC did not offer a 32K DRAM card an I burned up the 2nd proto type. Okay why did we have a "0E" 4.5ms delay instruction / opcode? It was used for very simple timing loops for paper tape readers / punches and 110 / 300 baud teletype I/O terminals every byte counted when you only had 4K. The "0E" got into the ADDS terminal driver code in a oversite (error) and bit us in the but, years later it we found it screwing up disk files a few time per year. The very core instruction set / opcodes comes from a CD200 CPU dating back to the late 1960's as I understand the CPU4 history (before my time). Good luck find docs on that CD200 system on the WWW even with Google.

Regarding the gap in the 1702 Bootstrap EPROM. I do not recall how the code in that EPROM should look. That EPROM did offer booting from the Hawk drive controller @ HEX F140 - F14F or a Sykes pinch roller cassette drive & controller @ HEX F??0 - F??F. The way you selected which device to bootstrap from on the very old CPU4 system was a group of four latching push button switches on the system front panel. ( I shared some very old Warrex pictures with David one of which shows a tech checking a old 32K DRAM card on a I/O bus extender card. Above the chassis on the table is a Sykes cassette drive and in the very right most card cage slot is the Sykes tape controller. You can also see the old Warrex / Centurion front panel with four sense switch button and a few other Run, Halt, Reset, InterR. and ... control buttons. All but the Load Select, Load Opsys and the R/F (Fixed Disk / Removable Disk) which is the "Sense Switch #4 button are removed in the later shipping CPU4 systems. The CPU opcode offer a "Branch / Jumper" command on Sense Switch 1, 2, 3 or 4 test status. The the 1702 EPROM code has the lower data field for the Hawk drive @ F140-F14F and the upper data field at F??0 - F??F for the Skyes tape control bootstrap. The buttons removed from the CPU4 front panel later reappear as the 8 position dip switch on the CPU5 and CPU6 cards. When it comes time to rev. engineer the opcodes David does a few extra 1702 rev 116 EPROMs I gave him to check for bit rot in the 1702's. The other 3 EPROM positions never used on the DSK-Auto card.

Back in the 1970's so few people had access to the minicomputers just locking the computer room door / office door was all you needed. No Internet to cause problems with hackers and few even had a MODEM on the computer. Centurion did offer a 300 baud MODEM that connected to 4-Port MUX port 3 at 300 baud for dial-in remote works to do time sharing on the system ( all other MUX ports are 9600 baud for local terminals ).

I Was the CD200 another earlier Warrex product? I know folks at the Computer History Museum and the info network is fairly wide if it's from elsewhere. Years ago I had to step inside JOHNNIAC's open cabinet to help rotate a display case of HP calculators. LOL. ---- I never saw a 1960's CD200 computer but was told the Warrex opcodes came from the CD200. A few other computer companies like [Fedder Data or Fetter Data?] and [Century Computer] had a spinoff products that had some roots in the CD200 from late 1960's. So the Warrex CPU4 had some roots in the CD200 and the Warrex disk operating system (@OSN with JCL) was designed from ideas in the IBM 360 operation system. @OSN = Opsys or Centurion Disk Operating System, JCL = Job Control Language,( like PC DOS Batch File Commands ) and CPL = Centurion Programing Language ( I never used it. I was Assembly Language / Machine Code person when I had to write diag code ). Ken R. --- My hope is when David is done with the Warrex / Centurion system and supporting items does find a home in a computer history museum this a bit of the 1970's high tech story on the Richardson, TX history ( Texas Instruments, Collins Radio, Danray - DSC, Computer Automation, MosTek (across town) and Warrex-Centurion in Richardson ( Dallas & Allen, TX over the years). Ken R. Now I'm wondering about the cable which connects the CMD, CPU, and DSK cards? Is that a private DMA channel (edit: just found your card descriptions from 4 months ago and you indeed indicate this is DMA control), and will the CPU operate fine without both the CPU and CMD or that 26 pin cable installed? I presume that cable lets the CMD inhibit the CPU and become bus master. ---- The 26 pin ribbon cable are just for DMA control signals to/from the CPU4's DMA card and the CPU5 & CPU6 onboard DMA control logic. The 16 bit memory address and 8 bit R/W data is not part of the 26 pin cable. Ken R. I saw the interrupt request daisy-chain jumper you noted on the slot 2 position and also also see another jumper a few pins up and was wondering if that might be a similar daisy-chain for DMA? It looks like 47 or 48 active signals on that bus with 36 going to the boot PROM and monitor/control panel, and 40 terminating resistors available. ---- The two backplane jumpers just pass the daisy chain interrupt from right to left (to the CPU) from I/O cards using the same interrupt level. Only I/O cards that shared the same interrupt level was the 10MB Hawk controller board set (Dsk/Auto and DisK2 cards) Only one Warrex - Centurion system shipped with two sets of 10MB Hawk controller boards (two sets supported four Hawks on controller one and then four more Hawks on controller two - total of 80MB of disk storage). If you had two sets of Dsk/Auto & Disk2 they could not have a empty slot to the cards left before getting to the CPU. Ken R. Random note: Looking closer at the CMD board it does have a 256 x 8 PROM (a TBP18S22) marked 'CMD F12-1.1'. I also see the equivalent part on the CPU board at B13 (using a 74S471). Any idea what's in those? I can't believe I missed those along the way, but I just found the high-rez photos, now. --- No idea they both are PROMs. Any Centurion boards with AMD2910 and PROMs its a 99% shot the PROM is part of the microcode to emulate CPU4 opcodes of control sate machine logic. Do not recall any system bootstrap PROM in any I/O cards other and the AutoLoad part (1702 EPROM) on the CPU4 DSK/Auto card. Ken R. A couple more random questions.... I get the halt/run, address, and interrupt level LEDs on the front panel, but what do the 'MAP 1, 2, 3, and ABT' and single letter F, L, M, V LEDs indicate? ---- If I recall correctly the more flashing front panel LED lights the more the computer sold for in the 1970s so more $$$. I never had any real use for the LEDs other than the address, Interrupt Level, Halt & Run. MAP 1, 2, 3 and ABT would be called extended memory page address bits today. The F, L, M & V are just flag status bit from the CPU ALU logic, again no real use unless you are stepping opcode by opcode writing bootstrap code I guess. Ken R. In your card description you indicate on the CPU, 'the lower 26 pin connector is for the front panel switches and LEDs'. More blinking lights and switches are always a plus, but do you recall what they indicated and did as that panel seems to be missing and the existing panel is driven from the backplane? ---- CPU - The upper 26 pin is DMA control. The lower 26 pin goes to the small card with (3) push button switches on the front panel ( R/F, Select & OPSYS ). The backplane 50 pin cable goes the long LED card on the front panel to display address & other bit status states in the LEDs. Ken R. Last question... I see the pair of 50 pin headers below the PROMS on the CPU & CMD boards and the 26 pin header on the CMD which I'm guessing may be the analog of the 26 pin monitor/control port on the CPU. Are the 50 pin headers test ports or were they possibly for putting the microcode into different parts if the original PROMs were unavailable? --- The 50 pin headers near the PROMs in the 2901 based cards for connection the the Centurion factory test / debug station. We did not have large logic analyze scope so we build test sets to step the microcode clock by clock for board level testing / repair. Thing happen to fast to trouble shoot with a two channel 100MHz scope alone. Ken R. --- The (3) 26 pin all lined up are the B-Data-Cable (point to point) to the 96MB CDC CMD drives. The single 26 pin cable on the CMD card edge is the DMA control cable to connect all the DMA using I/O cards to the CPU. The 50 (or 60 pin?) on the edge of the CMD card is the A-Control-Cable (daisy chain CMD 1, 2 & 3) to the CDC CMD drives. Ken R. Sorry to barrage you, but you seem to have all the answers and this is so close. Thanks again, Ken! ---- Do not mind the questions. Writing down this info may help save a spot in time regarding minicomputer history and let some new people learn a bit of the good & bad ideas of the past. Maybe a Zoom call in the future if David thinks it has value, this is David's and his teams project.