# HP 2100

From Wikipedia, the free encyclopedia

[](https://en.wikipedia.org/wiki/File:HP_1000_E-Series_minicomputer.jpg)

HP 1000 E-Series minicomputer with a 9895A dual 8 inch "flexible disc memory" drives.

The **HP 2100** was a series of [minicomputers](https://en.wikipedia.org/wiki/Minicomputer) produced by [Hewlett-Packard](https://en.wikipedia.org/wiki/Hewlett-Packard) (HP) from the mid-1960s to early 1990s. The 2100 was also a specific model in this series. The series was renamed **HP 1000** by the 1970s and sold as [real-time computers](https://en.wikipedia.org/wiki/Real-time_computing), complementing the more complex [IT](https://en.wikipedia.org/wiki/Information_technology)-oriented [HP 3000](https://en.wikipedia.org/wiki/HP_3000), and would be the starting point for a line of [desktop computers](https://en.wikipedia.org/wiki/Desktop_computer). They would eventually be phased out in favor of [UNIX](https://en.wikipedia.org/wiki/Unix)-based [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing) workstations.

HP entered the [minicomputer](https://en.wikipedia.org/wiki/Minicomputer) market in 1966, along with [Varian Data Machines](https://en.wikipedia.org/wiki/Varian_Data_Machines). Later, [General Automation](https://en.wikipedia.org/wiki/General_Automation), [Computer Automation](https://en.wikipedia.org/wiki/Computer_Automation), [Data General](https://en.wikipedia.org/wiki/Data_General), [Micro Systems](https://en.wikipedia.org/w/index.php?title=Micro_Systems&action=edit&redlink=1), and [Lockheed](https://en.wikipedia.org/wiki/Lockheed_Corporation) would also be competitors. The 2116A was the first model of the series, demonstrated November 7-10, 1966 at the [Joint Computer Conference](https://en.wikipedia.org/wiki/Joint_Computer_Conference) in San Francisco.[[1]](https://en.wikipedia.org/wiki/HP_2100#cite_note-1)[[2]](https://en.wikipedia.org/wiki/HP_2100#cite_note-2) It was designed by HP's Dymec division, after absorbing Data Systems Inc. (DSI), a subsidiary of [Union Carbide](https://en.wikipedia.org/wiki/Union_Carbide). DSI had designs for a 16-bit minicomputer called the DSI-1000, which would eventually evolve into the 2116A through HP's involvement.

## 2116A overview

[](https://en.wikipedia.org/wiki/File:HP_2647A_terminal.jpg)

An HP 2647A graphics terminal connected to an HP 1000 E-Series

The 2116A is a [16-bit](https://en.wikipedia.org/wiki/16-bit) [word-addressed](https://en.wikipedia.org/wiki/Word-addressable) [general purpose computer](https://en.wikipedia.org/wiki/Computer). [Main memory](https://en.wikipedia.org/wiki/Computer_data_storage) is 4096 words (4K), expandable to 8K of [magnetic core](https://en.wikipedia.org/wiki/Magnetic_core_memory) in the mainframe, or 16K with a memory extender. The 2116A features 16 [I/O](https://en.wikipedia.org/wiki/Input/output) slots in the mainframe, a 10 [MHz](https://en.wikipedia.org/wiki/Hertz) clock and a [memory cycle time](https://en.wikipedia.org/wiki/Memory_timings) of 1.6 [microseconds](https://en.wikipedia.org/wiki/Microsecond).

The 2116A had two subsequent revisions: the 2116B added support for up to 32K with a memory extender, and the 2116C incorporated a more compact model of [core memory](https://en.wikipedia.org/wiki/Magnetic_core_memory), allowing the full 32K to be housed within the computer mainframe. (32K 16-bit words, equivalent to 64K 8-bit bytes of memory.)

The HP 2116A’s [software](https://en.wikipedia.org/wiki/Computer_software), with a [FORTRAN](https://en.wikipedia.org/wiki/Fortran) [compiler](https://en.wikipedia.org/wiki/Compiler), [assembler](https://en.wikipedia.org/wiki/Assembly_language), [linker](https://en.wikipedia.org/wiki/Linker_(computing)), [loader](https://en.wikipedia.org/wiki/Loader_(computing)), [operating system](https://en.wikipedia.org/wiki/Operating_system), and [I/O drivers](https://en.wikipedia.org/wiki/Device_driver) were ready at the same time as the hardware. This was quite unusual, at a time when most computer vendors would roll out the [hardware](https://en.wikipedia.org/wiki/Personal_computer_hardware) first with little software. The 1967 issue of the [Hewlett-Packard Journal](https://en.wikipedia.org/wiki/Hewlett-Packard_Journal) called the HP 2116A "an unusual new instrumentation computer".

The HP 2116A had an oversized cabinet with 16 empty card slots for interface cards. Up to 48 could be fitted using one or more add-on I/O extender chassis. At introduction, HP engineers had interfaces for more than 20 instruments including "counters, nuclear scalers, electronic [thermometers](https://en.wikipedia.org/wiki/Thermometer), digital [voltmeters](https://en.wikipedia.org/wiki/Voltmeter), ac/ohms converters, data amplifiers, and input scanners." The HP 2116A's introduction began the age of modern [automated test systems](https://en.wikipedia.org/wiki/Automatic_test_equipment).

[](https://en.wikipedia.org/wiki/File:Hewlett_Packard_2115A_Computer_(1968).jpg)

HP 2115A Computer pictured without its power supply.

When HP discovered it sold more HP 2116A minicomputers for business applications than for instrumentation, HP introduced the short-lived 2115A in 1967, a cost-reduced variant of the 2116A with only 8 I/O slots, a bulky external power supply, and a 2116-style front panel. The HP 2116A of 1968 was stripped of [DMA](https://en.wikipedia.org/wiki/Direct_memory_access) and extended arithmetic. The 2114A featured a redesigned front panel, with reduced register displays and illuminated [touch switches](https://en.wikipedia.org/wiki/Touch_switch). The 2114 saw two further revisions: the 2114B added single-channel [DMA](https://en.wikipedia.org/wiki/Direct_memory_access) and [HSIO](https://en.wikipedia.org/w/index.php?title=HSIO&action=edit&redlink=1) options at the expense of a single I/O slot, and 2114C supported up to 16K maximum core in mainframe, at the expense of yet another I/O slot. The 2115A and 2114A/B/C have an 8 MHz clock and a 2.0 µs cycle time.

HP's Data Systems Division, initially based in [Cupertino, California](https://en.wikipedia.org/wiki/Cupertino,_California) and later moved to nearby [Santa Clara](https://en.wikipedia.org/wiki/Santa_Clara,_California), produced a long series of successful HP 21xx minicomputers that HP would not be able to retire from production despite five serious attempts to introduce successors, including the [HP 3000](https://en.wikipedia.org/wiki/HP_3000). By 1978, HP was the fourth largest minicomputer manufacturer, trailing only [DEC](https://en.wikipedia.org/wiki/Digital_Equipment_Corporation), [IBM](https://en.wikipedia.org/wiki/IBM), and [Data General](https://en.wikipedia.org/wiki/Data_General). The 16-bit instrumentation-oriented HP 21xx architecture continued and evolved for more than 20 years.

## 2100 series architecture

[](https://en.wikipedia.org/wiki/File:ESO_Hewlett_Packard_2116_minicomputer.jpg)

HP 2100 computer, shown with two 7970 tape drives, 7900-series hard disk, 2748 paper tape reader and a 2767A impact printer.[[3]](https://en.wikipedia.org/wiki/HP_2100#cite_note-3)

The HP 2100 is one of many 8- and 16-bit machine architectures said to be inspired by the [PDP-8](https://en.wikipedia.org/wiki/PDP-8). These can be characterized by use of RAM instead of registers, page-oriented addressing, and a small number of [accumulators](https://en.wikipedia.org/wiki/Accumulator_(computing)) (such as A and B) rather than a relatively large number of regular [registers](https://en.wikipedia.org/wiki/Processor_register) (such as R0-R7 found on the [PDP-11](https://en.wikipedia.org/wiki/PDP-11)). This philosophy can save money when RAM is less expensive than registers.

There were two 16-bit [accumulators](https://en.wikipedia.org/wiki/Accumulator_(computing)), called A and B which could do most [instructions](https://en.wikipedia.org/wiki/Instruction_(computer_science)) such as load or add, although the A register had a few more instructions, and two 1-bit flags called Overflow and Extend. The [program counter](https://en.wikipedia.org/wiki/Program_counter), 15 bits, was called P. All 68 instructions in the standard [instruction set](https://en.wikipedia.org/wiki/Instruction_set) were 16 bits long and executed in one memory cycle (1.6 microseconds), though indirect addressing would add more cycles.[[4]](https://en.wikipedia.org/wiki/HP_2100#cite_note-4) [Conditional branching](https://en.wikipedia.org/wiki/Conditional_branch) was done with a conditional skip over one instruction, which was usually a [jump instruction](https://en.wikipedia.org/wiki/Jump_instruction). There was no dedicated stack register.

The smallest addressable unit of memory was a 16-bit word (unlike the later [PDP-11](https://en.wikipedia.org/wiki/PDP-11), which was addressable by byte), and the maximum possible address was 32,767, which would fit in a word with one bit left over. The most significant bit of any memory-reference instruction indicated [indirect addressing](https://en.wikipedia.org/wiki/Addressing_mode#Memory_indirect): The word addressed by the instruction, instead of being the operand, contained the operand address. The most significant bit of *that* word could be set to indicate an additional level of indirect addressing, and this could be iterated any number of times, potentially in an infinite loop that could not be interrupted.[[5]](https://en.wikipedia.org/wiki/HP_2100#cite_note-5)

Since most instructions were 16 bits long, there was no room in an instruction for both a full 15-bit memory address and a meaningful instruction code. Memory was therefore addressed in terms of "pages" of 1,024 words each. Instructions that reference memory had 10 bits for a page offset (supporting values from 0 through 1023 decimal) and a single bit for a "zero/current" page indicator. If the page indicator bit was set the 10 bits were interpreted as an offset within the page that contained the instruction being executed; in other words, the high five bits of the operand address were taken from the P-register, the program counter. If the page indicator bit was clear, the offset represented an address within page zero. Thus 2,048 words could be addressed at once: 1,024 of them within page zero and another 1,024 within the same page as the instruction performing the reference.[[6]](https://en.wikipedia.org/wiki/HP_2100#cite_note-6) If necessary to access other memory locations the instruction could have its high bit set, denoting indirect addressing as described previously; the location addressed by the instruction would then contain the full 15-bit operand address (unless it too had its indirect bit set, etc.).

With no stack for saving procedure return points, indirect addressing was used to implement procedure call and return: The first word of a procedure was reserved for the return address, and the jump to subroutine (JSB) instruction would store the return address there. The return to caller was performed via a jump indirect through that word.[[7]](https://en.wikipedia.org/wiki/HP_2100#cite_note-7) This design also appeared in other machines of the era, such as the [CDC 3000](https://en.wikipedia.org/wiki/CDC_3000) series, [PDP-8](https://en.wikipedia.org/wiki/PDP-8), and [IBM 1130](https://en.wikipedia.org/wiki/IBM_1130), and makes it impossible for a function to be called [re-entrantly](https://en.wikipedia.org/wiki/Reentrancy_(computing)) unless the function maintains its own stack of return addresses.

The early machines in the series (including the 2116) were direct-execution machines but the 2100 and later machines were [microprogrammed](https://en.wikipedia.org/wiki/Microprogram). The 2100 offered a [writable control store](https://en.wikipedia.org/wiki/Microprogram#Writable_control_stores) allowing the user to extend and change the [vertical microcode](https://en.wikipedia.org/wiki/Microcode#Vertical_microcode).

The 2100-series of processors is one of the systems which the [SIMH](https://en.wikipedia.org/wiki/SIMH) multi-system emulator is able to run.

## Descendants and variants

[](https://en.wikipedia.org/wiki/File:HP9830A-HP9866.png)

HP Model 9830A desktop computer with optional Model 9866 thermal printer.

The HP 9810, 9820 and [9830](https://en.wikipedia.org/wiki/HP_9830) desktop computers used a slow, serialized [TTL](https://en.wikipedia.org/wiki/Transistor-transistor_logic) version of the 2116 CPU, although they did not ultimately use any of the operating system or application software, instead relying on user-friendly [ROM](https://en.wikipedia.org/wiki/Read-only_memory)-based [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) such as [BASIC](https://en.wikipedia.org/wiki/BASIC) which worked when powered up and integrated keyboards and displays rather than [disks](https://en.wikipedia.org/wiki/Disk_storage) or standard [terminals](https://en.wikipedia.org/wiki/Computer_terminal). In 1975, HP introduced the BPC, the world's first [16-bit](https://en.wikipedia.org/wiki/16-bit) [microprocessor](https://en.wikipedia.org/wiki/Microprocessor), using HP's [NMOS](https://en.wikipedia.org/wiki/NMOS_logic)-II process. The BPC was usually packaged in a ceramic hybrid module with the EMC and IOC chips, which added extended math and I/O instructions. The hybrid was developed as the heart of the new 9825 desktop computer. The later 9845 workstation added an [MMU](https://en.wikipedia.org/wiki/Memory_management_unit) chip. These were the forerunners of [personal computers](https://en.wikipedia.org/wiki/Personal_computer) and technical workstations.

The major differences between the original 2116 architecture and the BPC microprocessor are a completely redesigned I/O structure, the removal of multiple levels of indirect addressing, and the provision of a stack register for subroutine call and return. The elimination of multiple indirection made an additional bit available in a memory word containing an indirect address, allowing the maximum memory capacity to be increased from 32K 16-bit words to 64K. The BPC also added an input allowing the "current page" to be relative to the location of the current instruction, rather than a power-of-two aligned page.

The BPC was used in a wide range of HP computers, peripherals, and test equipment, until it was discontinued in the late 1980s.

[Poland](https://en.wikipedia.org/wiki/Poland) manufactured an HP 2114B clone since 1973. The Polish clones were called [MKJ-28](https://pl.wikipedia.org/wiki/MKJ-28) (prototype, 1973), [SMC-3](https://pl.wikipedia.org/wiki/SMC-3) (pilot production, 17 machines, 1975-1977) and [PRS-4](https://pl.wikipedia.org/wiki/PRS-4) (production in series over 150 machines, 1978-1987).

[Czechoslovakia](https://en.wikipedia.org/wiki/Czechoslovakia) produced its own HP 1000 compatible clones, designated [ADT4000](https://cs.wikipedia.org/wiki/ADT_(po%C4%8D%C3%ADta%C4%8D)) (4300, 4500, 4700, 4900). More than 1000 units were delivered by the vendors Aritma Prague (development), ZPA Čakovice and ZPA Trutnov between 1973 and 1990. Those computers served in power plants, including nuclear ones, other industry, military, at universities, etc., for their high reliability and real-time features. Operating systems were DOS/ADT (several versions) and Unix. The oldest hybrid ADT7000 (1974) was composed of digital ADT4000 and analog ADT3000 parts, but only the digital part was interesting for customers. ADT4316 (1976) had 16K words of ferrite core memory, the ADT4500 (1978) up to 4M words of semiconductor RAM. The ADT 4900 was designed as a [single-board computer](https://en.wikipedia.org/wiki/Single-board_computer), but its mass production did not start. Czechoslovak People's Army used ADT based MOMI 1 and MOMI 2 mobile minicomputers, built into a container carried by the Tatra 148 truck.

## Instruction overview

The instruction set contained 68[[8]](https://en.wikipedia.org/wiki/HP_2100#cite_note-8) or 70 instructions.[[9]](https://en.wikipedia.org/wiki/HP_2100#cite_note-9)

* [Arithmetic](https://en.wikipedia.org/wiki/Fixed-point_arithmetic) — Add, Increment, And, Or, [Exclusive-or](https://en.wikipedia.org/wiki/Exclusive-or)
* [Program Control](https://en.wikipedia.org/wiki/Control_unit) — Skip, Jump, Jump to [Subroutine](https://en.wikipedia.org/wiki/Subroutine)
* Shift and Rotate — Arithmetic and Logical Shifts, 16- and 17-bit Rotates
* Optional — Multiply, Divide, 32-bit Load and Store, 32-bit Shifts

## Model overview

### Early models (1966-1970)

Core memory, hardwired CPU. Similar to a [PDP-8](https://en.wikipedia.org/wiki/PDP-8) that has been pumped up to 16 bits and two accumulators.

* 2116A
* 2116B
* 2116C
* 2115A
* 2114A
* 2114B
* 2114C (advertised but never released; single prototype held by Computer History Museum)

### Second generation (1970-1974)

Core memory, [microprogrammed](https://en.wikipedia.org/wiki/Microprogram) CPU. An option allowed user microprogramming. Front panel buttons were illuminated by small [incandescent lamps](https://en.wikipedia.org/wiki/Incandescent_light_bulb) that burned out with use. Dark lights did not bother regular users, who knew the 1 and 0 sequences to load the [paper tape](https://en.wikipedia.org/wiki/Paper_tape) "loader-loader" instructions without seeing the panel's lights.

* 2100A
* 2100S

### 21MX (1975-1979)

The 21MX series featured a memory management unit and [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) memory expandable to 1,048,576 words (one [megaword](https://en.wikipedia.org/wiki/Mega-)). The bit displays on the front panel buttons used small red [LEDs](https://en.wikipedia.org/wiki/LEDs), instead of the incandescent bulbs used in earlier versions.

* M-series — 2105A, 2108A, 2112A (blue line on front panel)
* E-series — 2109A, 2113A (yellow line on front panel; E for Extended)
* F-series — 2111F, 2117F (red line on front panel; F for [Floating point](https://en.wikipedia.org/wiki/Floating_point) Processor in a separate 2U chassis)

The 21MX ran the HP RTE (Real Time) Operating System (OS). They started out as refrigerator-sized [rack computers](https://en.wikipedia.org/w/index.php?title=Rack_computer&action=edit&redlink=1) with lights and switches on the front panels. The last models would use a 1-chip processor and fit under a desk using a console terminal rather than a front panel.

The new L and A series models had [HP-IB](https://en.wikipedia.org/wiki/IEEE-488) interface ability, but as with all HP systems at that time, the blinking LED lights were removed from the front panel. Despite customer demands for a real-time ability and HP R&D's efforts using an installable real-time card, the RTE-A OS was not as good at real-time operations as RTE on a 21MX. This was an important reason this computer was hard to kill. Many companies use real-time operations to take a measurements and control processes — turn on or off a pump, heater, a valve, speed up or slow down a motor, etc.

### L-Series (1980)

* HP1000L SOS ([silicon on sapphire](https://en.wikipedia.org/wiki/Silicon_on_sapphire)) CPU and I/O processors

### A-Series (1981-1996)

Each addressable up to 32 MB of RAM.

1981:

* A600 - based on [Am2900](https://en.wikipedia.org/wiki/AMD_Am2900) [bit-slice](https://en.wikipedia.org/wiki/Bit_slicing) processor, 1 MIPS, 53k[FLOPS](https://en.wikipedia.org/wiki/FLOPS)
* A600+ - based on Am2900 bit-slice processor, supports code and data separation, optional ECC (error correcting) memory. Codename: LIGHTNING

1982??:

* A700 - based on AMD AM2903 bit-slice processor, optional hardware floating point processor, 1MIPS, 204kFLOPS, microprogramming, optional [ECC memory](https://en.wikipedia.org/wiki/ECC_memory). Codename: PHOENIX

1984:

* A900 - Provides [pipelined](https://en.wikipedia.org/wiki/Instruction_pipeline) data path, 3MIPS, 500kFLOPS, ECC memory. Codename MAGIC

1986:

* A400 - first single-board CPU including 4 serial lines; CPU fabricated by [VLSI Technology](https://en.wikipedia.org/wiki/VLSI_Technology) with their CMOS-40 process, 512kB RAM on board. Codename Yellowstone

1992:

* A990 - CPU implemented with two 208-pin CMOS [application-specific integrated circuits](https://en.wikipedia.org/wiki/Application-specific_integrated_circuit) (ASICs), 298 instructions, supports up to 512 MB of memory.

### Operating systems

Operating systems were RTE-II (2114-2116), RTE-III (2100), RTE-IV (21MX series), RTE-A ("A" series). Provided "Real Time Executive" with interrupt capability.

The operating system shell, even in the late 1970s, was very primitive, with a single-level [file system](https://en.wikipedia.org/wiki/File_system). For example, the command to run a [FORTRAN](https://en.wikipedia.org/wiki/Fortran) compiler would be as follows:[[10]](https://en.wikipedia.org/wiki/HP_2100" \l "cite_note-10)

ru, f77, &test,'test,%test

meaning run the f77 program, using special characters to distinguish between source file, object, and executable files for older FMGR files. A modern [Unix](https://en.wikipedia.org/wiki/Unix) command line uses an implied run, and files have dot extensions or internally stored characteristics ("magic numbers") to distinguish between different file types for a given project. It may have been the most primitive shell of any competitive minicomputer at the time.

The HP 1000 also was one of the few minicomputers that restricted file names to only five characters, rather than the six common at the time, which made porting and even writing programs a challenge. Newer RTE-A operating system for HP 1000 provided conventional directory structure with 16.4 file names, and made the ru command optional.

GRAPHICS/1000 was a FORTRAN 5 character name implementation of AGL, which was based on the HP 9830 graphics commands.

Alternatively, a specific [dual processor](https://en.wikipedia.org/wiki/Multiprocessing) configuration was sold (the HP 2000 series, later known as HP 2000/Access) which could run [HP Time-Shared BASIC](https://en.wikipedia.org/wiki/HP_Time-Shared_BASIC). In the original system (2000A), a well-equipped 2116 acted as the main processor while a 2114 acted as the communications multiplexer, simulating many [UART](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter) channels in software. In the last version (2000F), a 2100S and 2100A CPU were used, with the 2100A connected to up to 32 serial terminals via serial multiplexer interfaces. The HP 2000 series was introduced in 1969, sold until June 1978 and was supported until 1985.[[11]](https://en.wikipedia.org/wiki/HP_2100#cite_note-11) The HP 2000 series was the forerunner of the Tandem [NonStop](https://en.wikipedia.org/wiki/NonStop) architecture, Tandem being created when HP management stopped the HP 2000 product and its champions disagreed.[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*]

TODS (Test Oriented Disk System) was developed by a technician at the HP board repair center to improve turn around time in the center. It was used to load diagnostics from a central repository as opposed to loading individual paper tapes. TODS was also used on HP 2116 thru 21MX-F series for test systems for missiles such as Phoenix (9206B), Harpoon (9500D-354), Tomahawk and many others. Early test sets were the 9500A evolving to the 9500D, followed by the ATS (Automated Test System). Specific HP test systems such as the 9500D-A46 Minuteman launch component test set and ATS-E35 Peacekeeper/Minuteman launch component test set. The HP 8580/8542 Microwave ANA/ASA (Automatic Network Analyzer / Automated Spectrum Analyzer) ran on TODS. The TDRSS microwave transponder test set used TODS and was a very large system.

There was also MTOS (Magnetic Tape Operating System) similar to TODS.

### Introduction dates

* HP 2116A — Nov 1966
* HP 2115A — Nov 1967
* HP 2116B — Sep 1968
* HP 2114A — Oct 1968
* HP 2000A — Nov 1968 (2116-based timesharing system)
* HP 2114B — Nov 1969
* HP 2116C — Oct 1970
* HP 2114C — Oct 1970

## Notes

* 1. *History of the 2116A digital computer* <http://www.hp.com/hpinfo/abouthp/histnfacts/museum/earlyinstruments/0001/0001history.html>
  2. *50th Anniversary of HP's First Computer* <http://www.hpmuseum.org/forum/thread-5973.html>
  3. [*"Computing at ESO Through the Ages — The amazing advance of technology"*](http://www.eso.org/public/images/comparisons/potw1223a/). ESO Picture of the Week*. Retrieved 4 June 2012*.
  4. [HP 2116](http://www.hp9825.com/html/hp_2116.html)
  5. Hewlett-Packard Corporation (March 1967). "Section 2-26, Direct/Indirect". [*HP 2116A Computer Specifications and Basic Operation, Volume 1*](http://www.mirrorservice.org/sites/www.bitsavers.org/pdf/hp/2116/02116-9010_2116A_Vol1_Mar67.pdf) (PDF). p. 2-3.
  6. Hewlett-Packard Corporation (March 1967). "Section 2-24, Addressing". [*HP 2116A Computer Specifications and Basic Operation, Volume 1*](http://www.mirrorservice.org/sites/www.bitsavers.org/pdf/hp/2116/02116-9010_2116A_Vol1_Mar67.pdf) (PDF). p. 2-3.
  7. Hewlett-Packard Corporation (March 1967). "Section 2-67, JSB, Jump to Subroutine". [*HP 2116A Computer Specifications and Basic Operation, Volume 1*](http://www.mirrorservice.org/sites/www.bitsavers.org/pdf/hp/2116/02116-9010_2116A_Vol1_Mar67.pdf) (PDF). p. 2-7.
  8. <http://www.hp9825.com/html/hp_2116.html>
  9. <http://www.computerhistory.org/revolution/minicomputers/11/337/1935?position=1>
  10. [Fortran 77 manual](http://www.hp.com/products1/rte/tech_support/documentation/documentation3/92836-90001.pdf)
  11. <http://www.hpmuseum.net/display_item.php?hw=411>

<http://www.cs.ubc.ca/~hilpert/e/HP21xx/index.html>

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **The HP 21xx Processor Series** |  | | |

[[](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/front.jpg)](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/front.jpg)These pages attempt to present a technical overview and some reference material for the HP21xx series of processors. The focus here is on the technical aspects of the series, and more so for the early machines of the series. More comprehensive writeups of the development history are presented at [sites listed below](http://www.cs.ubc.ca/~hilpert/e/HP21xx/index.html#references).

Disclaimer & Limitations: Before receiving an [HP 2116C system](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116CSys/index.html) in 2003, my experience with HP 21xx series computers was limited to passing by an under-utilised HP 21MX system in a university machine room in the early 1980s. (Under-utilised meaning it was from some defunct project and was rarely, if ever, powered up for the years I saw it.) Following is what information I have gleaned about the series from various sources, and from my experience with rejuvenating the 2116C. There is some deduction, some presumption (and the occasional opinion) involved.

### Series Overview

The following table presents the range of models produced over the years, followed by a brief overview of the major changes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **HP 21xx Processors** (ordered by increasing Year of Introduction) | | | | | | | | | | | | **Series** | **Year of Introduction** | **Model** | **Logic Technology** | **Memory Technology** | **Cycle Time (µS)** | **Max. Memory (KWords)** | **Mem with Ext.** | **Module Size (KWords)** | **I/O Slots** | **I/O with Ext.** | | Early | 1966 | [2116A](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116.html) | CTµL, some TTL | core | 1.6 | 8 | 16 | 4 | 16 | 48 | | 1967 | 2115A | 2 | 8 | - | 4 | 8 | 40 | | 1968 | 2114A | 2 | 8 | - | 4 | 8 | ? | | ? | 2114B | ? | ? | - | ? | ? | ? | | 1968 | [2116B](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116.html) | 1.6 | 16 | 32 | 8 | 16 | 48 | | 1970 | 2114C | 2 | 16 | - | 4 | 8? | ? | | 1970 | [2116C](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116.html) | 1.6 | 32 | - | 8 | 16 | 48 | | 2100 | 1971 | 2100A | TTL, CTµL | core | 0.980 | 32 | - | 4,8 | 14 | 45 | | ? | 2100S | ? | ? | - | ? | ? | ? | | MX-M | 1974 | 2105A | TTL ? | LSI 4K chips | ? | 32 | - | ? | 4 | 20/36 | | 2108A | ? | 64 | 192 | ? | 9 | 25/41 | | 2112A | ? | 128 | 256 | ? | 14 | 30/46 | | MX-E | 1975 | 2109A | TTL ? | LSI 4K chips | ? | 64 | 192 | ? | 9 | 25/41 | | 2113A | ? | 128 | 250 | ? | 14 | 30/46 | | MX-M | 1976 | 2108B | TTL ? | LSI | ? | 640 KB | +1152 KB | ? | 9 | 41 | | 2112B | ? | 1280 KB | +768 KB | ? | 14 | 46 | | MX-E | 1976 | 2109B | TTL ? | LSI | ? | 640KB | +1152 KB | ? | 9 | 41 | | 2113B | ? | 1280 KB | +768 KB | ? | 14 | 46 | | 1000-M | 1979 | 2108M | see 2108B | | | | | | | | | 2112M | see 2112B | | | | | | | | | 1000-E | 1979 | 2109E | see 2109B | | | | | | | | | 2113E | see 2113B | | | | | | | | | 1000-F | 1979 | 2111F | ? | LSI | ? | 640 KB | +1800 KB | ? | 9 | 46? | | 2117F | ? | 1280 KB | +1800 KB | ? | 14 | 46 | |

**Early Models**: The first processor of the series - the [2116A](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116.html) - was introduced in 1966. Over the next couple of years the 2114A and 2115A were introduced, sharing the same architecture and instruction set but with reductions in speed, expansion capabilities and size/weight.

The 2116/5/4 models differ mainly in the following:

* memory capacity
* number of I/O interface slots
* speed
* power supply capacity
* size & weight

A few revisions of these models were released, culminating with the 2116C in 1970. The [2116A/B/C](http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116.html) models differ primarily in the maximum amount of memory which can be installed in the main cabinet.

Various hardware options could be obtained to provide extended arithmetic instructions, DMA, memory parity checking, and memory protection. Numerous types of [I/O interfaces](http://www.cs.ubc.ca/~hilpert/e/HP21xx/iointerfaces.html) were available, as well as several pieces of expansion equipment:

* 2150A Memory and I/O Extender: for the 2116A, a separate cabinet of similar size to the 2116.
* 2150B Memory and I/O Extender: for the 2115A and 2116B,C.
* 2151A I/O Extender: 8 additional slots for the 2114 and 2115 [ref: 12930 (pg. 2-1) interface manual, 2115 Vol. 1 ]. Haven't found any indication of use for 2116.
* 2152A Floating Point Processor: Adds floating point instructions to the instruction set. Several boards are installed in the main processor and connected to an external chassis.
* 2160 External Power Supply: for configurations which require more power.

The early models were contempories of the DEC PDP-8 in the mid-late-60s and competed with them in the marketplace. In contrast to DEC, which moved on to a new architecture in the early 1970s with the PDP-11, HP perpetuated the 2100 architecture through the 1970s and later 21xx models competed with the PDP-11s.

**2100 Models**: Reductions in the size of the memory modules over the course of the 2116A to the 2116C left a lot of empty space in the cabinet of the 2116C. This, together with replacement of the linear power supplies of the early models with a switching-mode supply, made it possible to produce a machine of similar or better capability than the 2116C in a cabinet nearer the size of the 2114. The early models were thus all superseded by the HP 2100A/S models in 1971.

The 2100 models also introduced proper micro-programming to the architecture, which contributed to extending the viability of the architecture into the future.

The 2155A I/O Extender was available to provide additional I/O slots. [ref: 12930 (pg. 2-1) interface manual.]

Around this time the architecture of the 21xx Series would also be adopted to form the basis of HP's early microprocessor-based desktop computers such as the 9830, 9825, etc.

**MX-M & -E Series**: In 1974 the 2100s were superseded by the MX series, the primary change being the replacement of core memory with LSI, based initially around 4 KBit chips. In 1975 another group of models providing improved performance were added to the product line. These standard and improved groups would be referred to as the M and E (Enhanced?) series.

**1000-M, -E & -F Series**: As near as I can figure the HP1000 name replaced the MX name, models were relabelled such that the suffix letter previously indicating the version now indicated the series (e.g. 2108B --> 2108M), and the F (Fast?) series models introduced.

### Logic Technology

The early models are based primarily on [Fairchild CTµL](http://www.cs.ubc.ca/~hilpert/e/HP21xx/CTL.html) small-scale integrated circuits. CTµL (Complementary Transistor MicroLogic) was one of the first IC logic families, dating from the mid-1960s. Supply voltages for the logic are +4.5V and -2V.

TTL ICs, and perhaps some DTL, also show up on many of the I/O interface boards and some other boards for these early models, running (marginally) off the 4.5V supply. Such boards may have a combination of CTµL and TTL ICs on them.

The 2100 models use a mixture of TTL and CTµL.

I don't have data but I suspect the 21MX and later models went to all or mostly TTL, as CTµL was obsolete by the mid-70s and TTL had improved in popularity, performance (Schottky and Low-power Schottky devices) and functionality (more MSI devices). The exception to this was around the I/O bus, which retained the logic levels from the CTµL era for the sake of compatibility with existing I/O interfaces. This would be something of an annoying legacy for the remainder of the series history, necessitating a -2V supply and somewhat-special driver ICs just for the sake of the I/O bus.

### Architecture

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/HP21xx/architecture.gif |

There are two 16-bit general-purpose registers (A and B), a 15-bit program counter, a 1-bit overflow flag and a 1-bit extend flag. Words in memory are 16-bits, addressing range is 32,768 words (the 16th bit in an address is used to indicate indirection).

The basic data format is 16-bit two's-complement integer. The overflow bit indicates overflow of the two's-complement (signed) interpretation of an accumulator. The extend bit indicates carry from 2^15 and so can be used for implementing multiple precision arithmetic, and is used in extended bit rotations.

When addressing memory from instructions, the address range is viewed in terms of pages of 1024 words. An instruction which references memory can directly reference locations in the base page (starting at 0) or the current page (the page in which the instruction is located). To get around this limitation, indirect addressing can be used.

Certain groups of what are referred to as micro-instructions can be combined into one word. While presumably a direct result of the hardware logic structure poking through to the instruction encoding this is not really micro-programming in the more proper sense of the term. See the [Programming Reference](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html) for more about the instruction set.

I/O is accomplished by controlling **channels**, identified by **select codes**. A channel is physically implemented by an **interface PCA** (printed-circuit assembly) plugged into a slot on the processor I/O bus. The interface PCA is connected by a cable to the external **device**. Channels are controlled via I/O instructions, they are not memory-mapped. Each channel supports a control bit, a flag bit, and one or more registers. In general, the control bit is manipulated by the program to indicate to the device the start of an operation, the flag bit is used by the device to indicate to the program the completion of an operation, and the register(s) carry device-specific commands, status and data.

A few channels and memory locations are reserved for special purposes. See the [Programming Reference](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#ioreserved).

Vectored interrupt handling is a standard provision of the processors. See the [Programming Reference](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#interrupts).

The M-register (memory address) and T-register (memory data transfer), while present in the hardware and on the front panel of some models, are not directly referenced by software.

The programming architecture and instruction set are fairly typical for smaller processors at the time of the original design (circa 1965), but just a few years later (by say 1970) the writing was on the wall for this architecture. By today's standards one can see various anomalies:

* Addressing is word-based. Byte manipulation is awkward.
* There is no hardware stack support, the return address for subroutines is stored in the first word of the subroutine. Subroutines are thus not re-entrant (by default at least).
* There are no index registers.
* All conditional instructions are of the form 'skip next instruction if condition'.

Attempts were made to address some of the anomalies in the later models.

The architecture of the 21xx series also can be likened to a 16-bit version of the PDP-8 with two accumulators. Anyone familiar with the PDP-8 processor will see a good deal of similarity between the two instruction sets.

### Packaged Systems

HP used the 21xx series processors in various systems:

* HP 2000 Timeshared BASIC: Started out on a single 2116 processor, went to a dual-processor 2116 (one CPU for I/O, one for general processing), then to a dual-processor 2100 configuration and so on.
* HP 1000: Complete general-purpose minicomputer systems (processor, peripherals, the RTE OS, compilers, etc.)
* Numerous instrumentation systems such as the HP 9600 Automatic Test/Measurement Systems.

(Note: the HP 3000 Management Systems use a different processor/architecture.)

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **The HP 2116 Processors** |  | | |

[](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/front.jpg)This page covers the hardware configurations of the three versions (A, B and C) of the 2116

processor. An attempt is made to elaborate the similarities and the differences between the three versions. It is not complete however, particularly with regards to the 2116A, for which I have little information.

The three rows of boards in the 2116 cabinet are roughly delineated as:

* Top row: Memory
* Middle Row: CPU
* Bottom row: Input/Output

The three versions differ primarily in the maximum amount of memory which can be installed in the main cabinet, reflected most noticeably in the re-organisation between the versions of the top row of boards. The A model contains up to two 4 KWord core modules while the B model contains up to two 8 KWord modules. The C model contains up to four 8 KWord modules, the modules being different than those of the B.

Note that aside from the I/O interface slots, the backplane is not a bus architecture: a particular slot is intended to receive a particular board. The I/O interface slots (203 thru 218) are a bus architecture, the only difference between slots being the select code assigned to the slot and consequent interrupt priority (lower slot number, higher priority).

The power supplies for the 2116s are relatively complex for 'simple' linear supplies. There are various inter-dependancies between the regulators for proper power-up/down sequencing. The construction of the main logic supplies of +4.5V and -2V is such that the +4.5V line is actually the regulator common and the regulators supply -4.5V and -6.5V.

**Gallery**

All photos are of a 2116C.

|  |  |  |
| --- | --- | --- |
| [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/open.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/open.jpg) Opening the front door gives access to the card cage and allows for front panel lamp replacement. There are also some test mode switches on the rear of the front panel. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/cage.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/cage.jpg) The card cage: DMA & memory on the top row, CPU on the left of the middle row, and I/O on the bottom. The 4 boards to the right on the middle row are just being stored there. On the far right of the middle row are power supply test points. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/hinged.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/hinged.jpg) The card cage slides out on 4 internal rails, the two left rails split and the cage then swings on hinges built into the two right rails, giving access to the backplane wiring and the power supply. |
| [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/splitrail.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/splitrail.jpg) The split rails on which the card cage mounts. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/pcabitslice.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/pcabitslice.jpg) One of the CPU boards: a 4-bit slice of registers and ALU. 4 of these boards are present in the CPU for the 16-bit word-width. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/crimps.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/crimps.jpg) The backplane is wired with a type of crimp connector. |
| [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/pcassa.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/pcassa.jpg) SSA: The planar-array of cores and the sense amplifiers of the 8KW memory module. Jumper block to connect with the XY DRIVER board included on the left. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/pcaxydriver.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/pcaxydriver.jpg) XY DRIVER: Address matrix drivers of the 8KW memory module. Note the TO-19 transistor providing a fix for one blown transistor in a quad-transistor DIP package. | [http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/pcainhibitdriver.jpg](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pics/large/pcainhibitdriver.jpg) INHIBIT DRIVER: The third board of the 8KW memory module. |

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116Aboards.gif |

**2116A Configuration**

The A version permits two 4 KWord modules to be installed in the main cabinet.

Each 4 KWord module appears to be composed of:

* 2 **Sense Amplifier** boards,
* 4 **Address Driver** boards,
* 2 **Inhibit Driver** boards,
* **Core Stack**: 4K core stack located behind the backplane.

Power supply voltages and primary usage are as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Voltage** | **Use** | | +4.5 | logic | | -2 | | ? | core X/Y drivers | | ? | | ? | core inhibit drivers | | +12 | I/O line drivers | | -12 | |

Still much to find out about this model version.

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116Bboards.gif |

**2116B Configuration**

The B version replaced the two 4 KWord modules of the A with two 8 KWord modules, made possible in large measure by the use of [anti-coincident addressing](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html).

Each 8 KWord module is composed of:

* 2 **Sense Amplifier** boards: each board has 17 sense amplifiers for 4K of core,
* 2 **Driver Switch** boards: 1 board drives the X lines for 8K, the other the Y lines,
* 2 **Inhibit Driver** boards: each board has 17 inhibit drivers for 4K,
* **Core Stack**: 8K core stack located behind the backplane. (The stack is actually two groups of 4K with one dimension of address lines flipped between the two groups for anti-coincident addressing.)

Power supply voltages and primary usage are as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Voltage** | **Use** | | +4.5 | logic | | -2 | | +22 | core X/Y drivers | | -22 | | +32 | core inhibit drivers | | +12 | I/O line drivers, core sense amps | | -12 | |

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/HP21xx/HP2116Cboards.png |

**2116C Configuration**

The C version permits four 8 KWord modules to be installed in the main cabinet, however these modules are different than the 8 KWord modules of the B. The core stacks at the back are gone, the core is now mounted on the regular plug-in boards.

Each 8 KWord module is comprised of 3 boards:

* **SSA** board (core stack and sense amps),
* **X Y DRIVER** board,
* **INHIBIT DRIVER** board,
* interconnect plug between the SSA and X Y DRIVER boards. Note there are two types of interconnect, for the different relative orders of the 2 boards. The 0-7K and 16-23K positions use interconnect part no. 02116-63244, while 8-15K and 24-31K use 02116-63243.

The lowest 8 KWord address range is covered in slots 20, 21 and 22. I don't know the addressing order for the other slots.

The new memory modules are similar to those used in the 2100A, enough so that large portions of the schematic for the 2100A modules applies to the 2116C modules.

Note that because the maximum addressable memory (32 KWords) can now be installed in the main cabinet, the memory extender slots (221,222) are no longer needed.

Slots 15 and 16 are presumably for the memory parity and protection options (order not known).

The power supply was altered from that of the B for different voltages required by the new memory modules:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Voltage** | **Use** | | +4.5 | logic | | -2 | | +20 | core X/Y drivers, inhibit drivers | | -20 | core X/Y drivers | | +12 | I/O line drivers, core sense amps | | -12 | |

The alterations focus around the removal of the +32V supply and increasing the current capacity of the +20V supply via use of the 'old' +32V filter capacitors. There is also a change between the B and C models in the manner in which the line supply is switched on/off (push-button on the B, toggle switch on the C). The majority of the schematic for the 2116B supply (2116B Vol. 2 pg. 5-91) still applies to the 2116C.

### Board List

Following is a list of boards for use in the 2116A, B and C. A "?" in the slot column for the model indicates that I do not know whether it is permissible to use that board in that processor model. In most such cases there is another version of the board which is known to be applicable to that processor.

Note: "V Dat Dv" is Version, Date and Division.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **CPU** | | | | | | | | | **Name / Label** | **Board Number** | **Revision V Dat Dv** | **Description** | **2116A Slot** | **2116B Slot** | **2116C Slot** | **Refs** | | Front Panel Coupler | 02116-6184 | ? | Difference from 6208 unknown. | ? | 101 | ? | IB2 | | 02116-6208 | **D** 942 22 |  | ? | 101 | 101 | MB2.70,PBC | | Arithmetic Logic | 02116-6026 | **L** 805 22 | 4-bit slices of the registers and ALU. | ? | 102-5 | 102-5 | MB2.70,PBC | | TIMING GEN | 02116-6281 | 923 |  | ? | 106 | ? | MB2.70 | | **02116-63220** | **C 1043 22** | Difference from 6281 unknown. | ? | ? | 106 | PBC | | Instruction Decoder | 02116-6027 | **K** 830 22 |  | ? | 107 | 107 | MB2.70,PBC | | Shift Logic | 02116-6029 | **K** 729 22 |  | ? | 108 | 108 | MB2.70,PBC | | Extended Arithmetic Unit Timing | 02116-6196 | ? | Provides additional arithmetic instructions. | ? | 109 | ? | IB2 | | Extended Arithmetic Unit Logic | 02116-6202 | ? | See above. | ? | 110 | ? | IB2 | |  | | | | | | | | | **Memory** | | | | | | | | | **Name / Label** | **Board Number** | **V Dat Dv** | **Description** | **2116A** | **2116B** | **2116C** | **Refs** | | Memory Module Decoder | 02116-6300 | 937 | Selects the addressed module. |  | 2 |  | MB2.70 | | 02116-6274 | 811 | Selects the addressed module. (BACKDATED) |  | 2 |  | MB2.70 | | Direct Memory Logic | 02116-6069 | 907 | Buffering, switching between CPU access and DMA access. In spite of the name, this board is required, even if the DMA option boards are not present. |  | 20 |  | MB2.70 | | 02115-6044 | 821 | Logic appears to be identical to 02116-6069, but bias/load resistors are different, this board uses CTL, 6069 uses CTL & TTL. (BACKDATED) |  | 20 |  | MB2.70 | | Sense Amp | 02116-6298 | 902 | 17 sense amps for 4K. |  | 10,11, 12,13 |  | MB2.70 | | 02115-6001 | 744 | 17 sense amps for 4K. (BACKDATED) |  | 10,11, 12,13 |  | MB2.70 | | Driver Switch | 02116-6266 | 943 | Address drivers for 1 dimension of 8K. |  | 8,9, 14,15 |  | MB2.70 | | Inhibit Driver | 02116-6265 | 819 | 17 inhibit drivers for 4K. |  | 4,6, 16,18 |  | MB2.70 | | MDB | **02116-63248** | **A 1018 22** | Memory Data Buffer? |  |  | 13 | PBC | | MAD | **02116-63212** | **A 1016 22** | Memory Address Decoder? |  |  | 14 | PBC | | SSA | 02116-63207 | **A** 1035 22 | 8K core stack and sense amplifiers. (also: **02116-83209**) (on core stack: 02116-63207 / **B** 1035 22 / **5087-1004**) |  |  | 9,10, 19,20 | PBC | | X Y DRIVER | **02116-63211** | **A-1016-22** | Address drivers for 2 dimensions of 8K. |  |  | 8,11, 18,21 | PBC | | INHIBIT DRIVER | **02116-632**60 | **A-**1042**-22** | Inhibit drivers for 8K. |  |  | 7,12, 17,22 | PBC | | Memory Parity Check | 12591-6001 | ? | Optional. | ? | 3 | ? | M13206 | | Memory Protect | ? | ? | Optional. | ? | 21 | ? |  | | Memory Extender | 02116-6181 | ? | For connection to optional 2150A extender cabinet to provide additional memory. | 221,222 | 221,222 |  | BI2 | |  | | | | | | | | | **DMA** | | | | | | | | | **Name / Label** | **Board Number** | **V Dat Dv** | **Description** | **2116A** | **2116B** | **2116C** | **Refs** | | DMA WORD COUNT | **02116-6206** | **C 908 22** | The DMA boards provide 2 independant channels of Direct Memory Access for high speed I/O devices such as magnetic tape and disks. | ? | 116,117 | 1,2 | PBC | | DMA Address Encoder DMA ADDRESS ENC | 02116-6205 | ? | ? | 118 | ? | IB2,M13206 | | 02116-6315 | **B 936 22** | ? | ? | 3 | PBC | | DMA CONTROL | 02116-6204 | A 751 6 | ? | 119 | 4 | PBC | | DMA Character Packer DMA CHAR PACKER | 02116-6203 | ? | ? | 120 | ? | IB2,M13206 | | 02116-6313 | **A 750** 22 | ? | ? | 5 | PBC | |  | | | | | | | | | **I/O Bus Support** | | | | | | | | | **Name / Label** | **Board Number** | **V Dat Dv** | **Description** | **2116A** | **2116B** | **2116C** | **Refs** | | I/O Control  A201 | 02116-6041 | 640 710 L 839 22 | Part of I/O bus interface with CPU. Global interrupt control. | 201 ? | 201 ? | ? 201 | MA3.67,MB2.70 PBC | | I/O Address | 02116-6042 | 651 703 | Part of I/O bus interface with CPU. Select-code decoding & central interrupt register. | 202 | ? | ? | MA3.67 | | I/O Address CENTRAL INTERR. | 02116-6194 | na C 829 22 | Referred to by both names. Difference from 6042 unknown. | ? ? | 202 ? | ? 202 | MB2.70 PBC | | I/O BUSS LOADER (sic) | **02116-6047** | 701 **B 920 22** | Termination resistors, should come after all I/O interface boards. Not required if all interface slots (203-218) are filled. | 218 | 218 | 218 | MA3.67,PBC | | I/O Extender (part 1) | 02116-6182 | ? | For connection to optional 2150A extender cabinet to provide additional I/O slots. | 219 | 219 | 219 | BI2 | | I/O Extender (part 2) | 02116-6183 | ? | See above. | 220 | 220 | 220 | BI2 | |  | | | | | | | | | **Miscellaneous** | | | | | | | | | **Name / Label** | **Board Number** | **V Dat Dv** | **Description** | **2116A** | **2116B** | **2116C** | **Refs** | | POWER FAIL | 02116-6175 | D 925 22 | Power fail detection and interrupt generation. Also generates POFP (Power On-Off Pulse) and PON (Power On Normal) signals. | ? | 1 | ? | MB2.70 | | POWER FAIL | 02116-6305 | **D** 925 22 | This is a -6175 with a handwritten label on it saying it has been rebuilt as -6305. One zener may have been replaced. | ? | ? | 6 | PBC | | Power Fail with Auto-Restart | 12588-6001 | ? | Appears to be an optional alternative to the Power Fail board. | ? | 1 | ? | M13206 | |  | | | | | | | | | **Power Supply** | | | | | | | | | **Name / Label** | **Board Number** | **V Dat Dv** | **Description** | **2116A** | **2116B** | **2116C** | **Refs** | | Logic Supply Regulators | 02116-6014 | D 638 22 D 1049 22 | Control for +4.5V, -2V, +12V regulators. | ? | 301 | 301 | MB2.70 PBC | | Memory Supply Regulators | 02116-6015 | D 822 6 | Control for +22V, -22V, -12V, +32V regulators. | ? | 302 |  | MB2.70 | | 02116-63214 | E 1049 22 | Control for +20V, -20V, -12V regulators. Almost identical to -6015 but with the components for +32V left out. (Also marked with 02116-8015.) |  |  | 302 | PBC | | Crowbars PCA | 02116-6126 | D 837 6 |  |  | 121,2 |  | MB2.70 | | 02116-63213 | D 1009 22 | Identical to -6126 but with the components for +32V left out. (Also marked with 02116-8126.) |  |  | 121,2 | PBC | |

Key to Refs:

* PBC: Physical board from my 2116C.
* MA3.67: Manual for 2116A, Vol. 3, 1967; from bitsavers.
* MB2.70: Manual for 2116B, Vol. 2, 1970; from bitsavers, esp. Pg. B-02/507.
* IB2: Class (Instructional) manual for 2116B, Vol. 2; from bitsavers.
* M13206: Manual in 13206 PDF from bitsavers, contains list of 2116B configs for 2000A systems.

### Miscellania

* Power supply and backplane are accessed from front by removing 4 flat-head screws on black bezel behind front door and pulling out the card cage. Card cage is mounted on internal sliding rails.
* To access crowbars remove 6 flat-head screws on card cage, not round-head screws on crowbar cover.
* The two power supply regulator / heatsink modules drop down after removing screws behind the rear panel.
* Front panel lamps are CM 345. These lamps are specified as 40mA @ 6V, however they run under-voltage in the processor, which improves their longevity. The lamps in register-bit positions draw about 31ma @ 4.1V, the lamps in the push-buttons draw about 36mA @ 5.4V, as measured in a 2116C.
* Available: schematic for 2116C Power Supply.
* Available: schematic for X Y Driver board of 2116C 8 KWord memory module.

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **Fairchild CTµL Integrated Circuits** |  | | |

Fairchild CTµL (Complementary Transistor MicroLogic) was one of the first IC logic families, dating from the mid-1960s, and is a form of non-saturating, current-mode logic, similar to ECL.

To quote from the Fairchild 1966 catalog:

Fairchild Complementary Transistor Micrologic integrated circuits were designed for very high-speed, low cost commercial systems applications. The logic form is AND-OR-NOT. All circuits have provisions for output OR ties.

ITT second-sourced some of these ICs, based on the observation of some ICs stamped with "ITT" on boards in an HP 2116C.

**Known Devices:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **CTµL Integrated Circuits & Cross Reference** | | | | | | **HP** | **Fairchild** | **F SL** | **Function** | **Refs** | | 1820-0186 | ? | ? | dual 2-in AND, dual resistors | B2,2100 | | 1820-0187 | ? | ? | dual 2-in NOR, dual resistors | B2,2100 | | 1820-0482 | ? | ? | binary to 1-of-8 decoder | 2100 | | 1820-0485 | ? | ? | hex level restorer (2-in AND, quint 1-in) | 2100 | | 1820-0952 | 9952 | SL3455 | dual 2-in NOR | F66,B2,2100 | | 1820-0953 | 9953 | SL3456 | dual 2-in AND, 3-in AND | F66,B2,2100 | | 1820-0954 | 9954 | ? | dual 4-in AND | F66,B2,2100 | | 1820-0955 | 9955 | ? | 8-in AND | F66,B2,2100 | | 1820-0956 | 9956 | SL3459 | dual 2-in AND Buffer | F66,B2,2100 | | 1820-0957 | 9957 | ? | MS FF | F66,B2,2100 | | 1820-0964 | 9964 | ? | dual 3-in AND, 1-in gate, OR'able | F66,B2,2100 | | 1820-0965 | 9965 | ? | quad 1-in gate, OR'able | F66,B2,2100 | | 1820-0966 | 9966 | ? | quad 2-in AND, 2 outputs ORed | F66,B2,2100 | | 1820-0967 | 9967 | ? | JK FF | F66,B2,2100 | | 1820-0968 | 9968 | ? | dual Latch | F66,B2,2100 | | 1820-0971 | 9971 | SL3467 | quad 2-in AND, outputs ORed in 2 pairs | F66,B2,2100 | | 1820-0972 | ? | ? | dual 2-in NOR, dual resistors | 2100 | | 1820-0973 | ? | ? | dual 2-in AND, 3-in AND | 2100 | | 1820-0974 | ? | ? | dual 2-in AND, dual resistors | 2100 | | 1820-0975 | ? | ? | JK FF (similar to -0967) | 2100 | |

**Source of Data, Key to Refs:**

* F66: 1966 Fairchild catalog.
* B2: Manual for 2116B, Vol. 2, Pg. A-08/504, 1970; from bitsavers.
* 2100: Drawings manual for 2100A, 1975; from bitsavers.
* Physical ICs in a 2116C.

**Period:** Mid 1960s to early 1970s. The earliest direct reference is a Fairchild catalog from 1966. The HP 2116A was being designed at least as early as 1965. Additional devices show up in the HP 2100A, which was released in 1971.

**Supply Voltages:** Vcc = +4.5 V ±10%, Vee = -2V ±10%.

**Technology:** The diagram presents the basic schematic of CTµL AND and NOR gates. These are taken from the Fairchild 1966 catalog. Delay time is 3 nS for AND gates of this form, 9 nS for NOR gates.

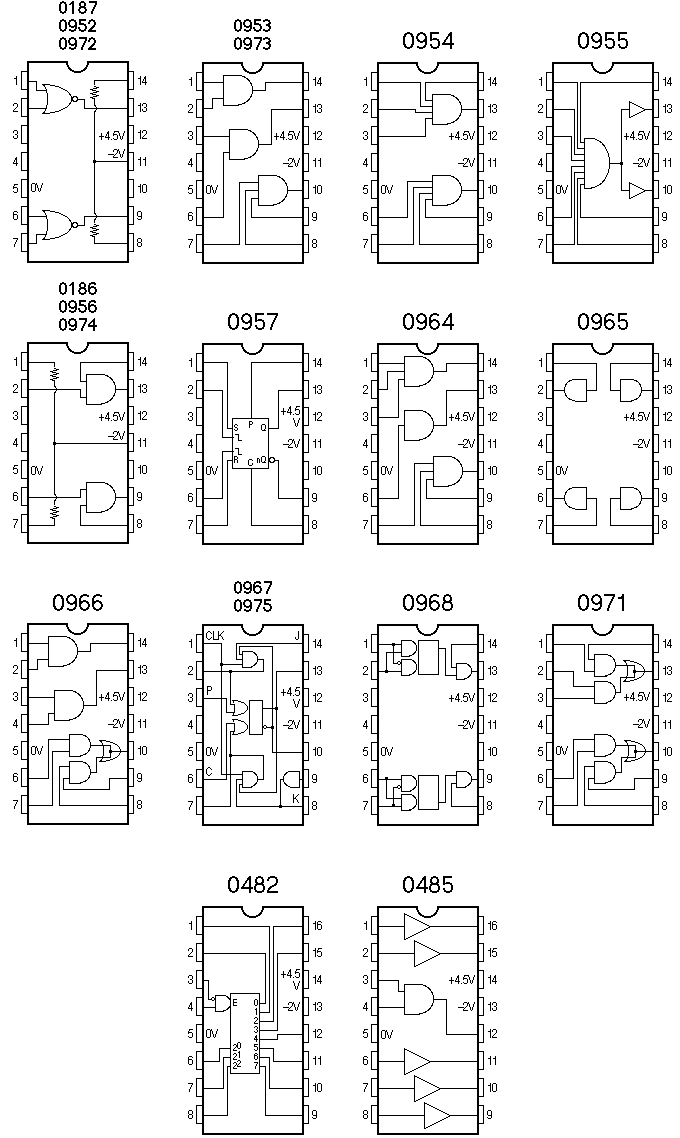
|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/HP21xx/CTLinternal.gif |

**Logic:** Gate symbols are presented in accordance with

* TRUE/1 = +? V
* FALSE/0 = ? V

**Packaging:** Standard 14-pin DIP with some later devices in 16-pin DIP.

**Note:** IC packages as viewed from above.



|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **Programming Reference** |  | | |

### Instruction Set Quick Reference

This reference covers the base instruction set applicable to processors up to and including the 2100A. The extended instruction set (a hardware option) and the additions of later processors are not covered.

The instruction set is broadly divided into 4 groups. The instructions documented here occupy a maximum of 1 word. In the Shift-Rotate Group and Alter-Skip Group, 'micro-instruction encoding' permits multiple instructions in one word.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **Key** | | | r | : A=0 / B=1 register | | i | : Direct=0 / Indirect=1 addressing | | p | : Zero=0 / Current=1 page | | f | : Hold=0 / Clear=1 device flag | | s | : bit of I/O device select code | | m | : bit of memory address | | e | : 1 to enable rotate or shift micro-instruction | | . | : 1 to enable additional micro-instruction | | \_ | : doesn't matter | |

* [Shift-Rotate Group](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#srg)
* [Alter-Skip Group](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#asg)
* [Input/Output Group](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#iog)
* [Memory Reference Group](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#mrg)
* [Some Useful Micro-Instruction Sequences](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#usefulseq)
* [Micro-instruction Notes](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#micro)
* [Miscellaneous Notes](http://www.cs.ubc.ca/~hilpert/e/HP21xx/pgmref.html#notes)

|  |
| --- |
| **HP 21xx Series Instruction Set** |
| Opcode Bit Encoding Description  ------ --------------------- ------------------------------------------------------- |
| **Shift-Rotate Group:** |
| NOP 0 000 000 000 000 000 No operation  CLE 0 000 .0. ... 1.. ... 0 --> E  SLr 0 000 r0. ... ..1 ... Skip if LSB of A or B = 0  rLS 0 000 r0e 000 .e. 000 Arithmetic Shift Left A or B  rRS 0 000 r0e 001 .e. 001 Arithmetic Shift Right A or B  RrL 0 000 r0e 010 .e. 010 Rotate A or B Left  RrR 0 000 r0e 011 .e. 011 Rotate A or B Right  rLR 0 000 r0e 100 .e. 100 Shift A or B Left, 0 --> sign  ERr 0 000 r0e 101 .e. 101 Rotate A or B Right with E  ELr 0 000 r0e 110 .e. 110 Rotate A or B Left with E  rLF 0 000 r0e 111 .e. 111 Rotate A or B Left Four |
| **Alter-Skip Group:** |
| CLr 0 000 r10 1.. ... ... 0 --> A/B  CMr 0 000 r11 0.. ... ... NOT(A/B) --> A/B (one's CoMplement A)  CCr 0 000 r11 1.. ... ... -1 --> A/B (Clear and Complement A)  CLE 0 000 .1. .01 ... ... 0 --> E  CME 0 000 .1. .10 ... ... NOT(E) --> E  CCE 0 000 .1. .11 ... ... 1 --> E  SSr 0 000 r1. ... .1. ... Skip if Sign of A/B = 0  SLr 0 000 r1. ... ..1 ... Skip if LSB of A/B = 0  SZr 0 000 r1. ... ... .1. Skip if A/B = 0  SEZ 0 000 .1. ... 1.. ... Skip if E = 0  INr 0 000 r1. ... ... 1.. A/B + 1 --> A/B, if signed overflow set OVF, if carry 15 set E  RSS 0 000 .1. ... ... .01 Reverse Skip Sense |
| **Input/Output Group:** |
| HLT 1 000 \_1f 000 sss sss Halt  STF 1 000 \_10 001 sss sss 1 --> device flag  CLF 1 000 \_11 001 sss sss 0 --> device flag  SFC 1 000 \_10 010 sss sss Skip if device flag = 0 (device busy)  SFS 1 000 \_10 011 sss sss Skip if device flag = 1 (device ready)  MIr 1 000 r1f 100 sss sss device buffer OR A/B --> A/B (Merge Into A or B)  LIr 1 000 r1f 101 sss sss device buffer --> A/B (Load Into A or B)  OTr 1 000 r1f 110 sss sss A/B --> device buffer (OuTput A or B)  STC 1 000 01f 111 sss sss 1 --> device control bit  CLC 1 000 11f 111 sss sss 0 --> device control bit  STO 1 000 010 001 000 001 1 --> overflow  CLO 1 000 011 001 000 001 0 --> overflow  SOS 1 000 01f 011 000 001 Skip if overflow = 1  SOC 1 000 01f 010 000 001 Skip if overflow = 0 |
| **Memory Reference Group:** |
| LDr i 110 rpm mmm mmm mmm (M) --> A/B (Load A or B)  STr i 111 rpm mmm mmm mmm A/B --> (M) (Store A or B)  ADr i 100 rpm mmm mmm mmm A/B + (M) --> A/B, if signed overflow set OVF, if carry 15 set E  CPr i 101 rpm mmm mmm mmm Skip if A/B <> (M) (ComPare to A or B)  AND i 001 0pm mmm mmm mmm A AND (M) --> A  XOR i 010 0pm mmm mmm mmm A XOR (M) --> A  IOR i 011 0pm mmm mmm mmm A OR (M) --> A  ISZ i 011 1pm mmm mmm mmm (M) + 1 --> (M), skip if (M) = 0  JMP i 010 1pm mmm mmm mmm Jump: M --> P  JSB i 001 1pm mmm mmm mmm Jump to SuBroutine: P+1 --> (M), M+1 --> P |
|  |
| **Some Useful Micro-instruction Sequences:** |
| CLr,INr 1 --> r  CMr,INr -r --> r (take negative of A or B)  CMr,INr  CMr r - 1 --> r (decrement A or B)  SZr skip if r = 0  SZr,RSS skip if r <> 0  SSr skip if r < 0  SSr,RSS skip if r >= 0  SSr,SZr skip if r <= 0  SSr,SZr,RSS skip if r > 0 |

#### Micro-instruction Notes:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **Table 1: Permitted Shift-Rotate combinations** (executed left to right) | | | |  |  |  |  | | --- | --- | --- | --- | | ALS, ARS, RAL, RAR, ALR, ALF, ELA, ERA, | CLE, | SLA, | ALS ARS RAL RAR ALR ALF ELA ERA | | |  |  |  |  | | --- | --- | --- | --- | | BLS, BRS, RBL, RBR, BLR, BLF, ELB, ERB, | CLE, | SLB, | BLS BRS RBL RBR BLR BLF ELB ERB | |  |  | | --- | | **Table 2: Permitted Alter-Skip combinations** (executed left to right) | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | CLA, CMA, CCA, | SEZ, | CLE, CME, CCE, | SSA, | SLA, | INA, | SZA, | RSS | | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | CLB, CMB, CCB, | SEZ, | CLE, CME, CCE, | SSB, | SLB, | INB, | SZB, | RSS | | |

1. Permitted sequences of micro-instructions are shown in tables 1 and 2. Only one micro-instruction can be chosen from a multiple choice column.
2. Shift-Rotate instructions have two possible shift-rotations.   
   A "1" in bit 9 enables shifts or rotates in the first position,   
   a "1" in bit 4 enables shifts or rotates in the second position.
3. In the Alter-Skip group, if two or more skip functions are combined, the skip will occur if either or both conditions are met. One exception exists: refer to RSS instruction.
4. RSS Instruction: Skip occurs for any of the preceding skip instructions, if present, when the non-zero condition is met. RSS without a skip instruction in the word causes an unconditional skip. If a word with RSS also includes both SSA/B and SLA/B bits 15 and 0 must both be one for skip to occur. In all other cases the skip occurs if one or more skip condition is met.
5. Note CLE, SLA and SLB instructions have encodings in both the Shift-Rotate group and Alter-Skip group.
6. (Bit 11 of SLA instruction in Alter-Skip group appears to be incorrect in source documentation!!)

#### Miscellaneous Notes:

1. The A and B registers may be addressed as locations 00000 and 00001 respectively.
2. ISZ referencing locations 0 or 1 (A or B register) does not cause setting of the extend or overflow bits (unlike INA and INB).
3. When applicable, the extend (E) bit is set upon carry from bit 15 (2^15). The overflow bit is set if an applicable operation results in overflow of the two's complement (signed) interpretation of the result, i.e.: if the sign bit is altered such that it no longer correctly represents the result.
4. The extend bit is not affected unless specifically stated. However, if a rotate-with-E instruction (ERA/B, ELA/B) is coded but disabled by a "0" in bit 9 or 4, the E register will be updated even though the A or B register is not affected; code a NOP (three zeros) to avoid this situation.
5. I/O Instructions: Bit 11, where relevant, specifies the A- or B-register or distinguishes between set control and clear control; otherwise it may be "1" or "0" without affecting the instruction (although the assembler will assign zeros, as shown).

(Compiled from HP-2100A Reference Manual from <http://oscar.taurus.com/~jeff/2100/index.html>)

### Interrupts

Prioritised vectored interrupt servicing is a standard provision of the processors. The interrupt vector address for a channel is the same as the channel select code. To handle an interrupt the appropriate interrupt vector memory location must be loaded with an instruction (not the address of the interrupt service routine). For example, to handle an interrupt for channel 11, memory location 11 must be loaded with an instruction. When an interrupt occurs the instruction will be executed without first altering or loading the PC. Consequently, if the instruction is a JSB to an interrupt service subroutine then the return address placed in the first word of the subroutine will be the PC value when the interrupt occurred (in other words: the desired restart address).

Once the processor has trapped thru an interrupt vector it is deemed to be executing at the associated priority level and remains so until the flag bit (or control bit?) for the interrupting channel is cleared. Thus, the last thing a typical interrupt service subroutine does is to clear the channel flag and return from the subroutine.

Lower priority interrupts (channels with higher select code) cannot interrupt the processor when it is executing at a higher priority level (interrupted by a channel with lower select code).

### Reserved Channels and Memory Locations

A few channels and memory locations have dedicated applications, independant of the I/O bus interface slots. These are documented in the table below.

For DMA, memory protection and parity the applicable optional board(s) must be installed.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Reserved Channels and Memory Locations** | | | | | | | | **Input/Output Use** | **Output Register** | **Input Register** | **Control Bit** | **Flag Bit** | **Channel / Address** | **Memory Location** | | global device & interrupt control |  |  | clear all controls [1] | interrupts enable | **00** | A register | | front panel & overflow | S-register (front panel lamps) [6] | front panel switches / S-register [6] |  | overflow | **01** | B register | | DMA Channel 1 control [3] | address / length | length | select address or length | DMA complete | **02** | A/B exit, mem protect lower bound | | DMA Channel 2 control [3] | address / length | length | select address or length | DMA complete | **03** | A/B exit | | last interrupt |  | address [2] |  |  | **04** | power fail interrupt | | memory protection & errors [5] | protect upper bound | error type/addr | protection interrupt enable | parity interrupt enable | **05** | memory error interrupt | | DMA Channel 1 [4] | device |  | start | abort | **06** | DMA 1 interrupt | | DMA Channel 2 [4] | device |  | start | abort | **07** | DMA 2 interrupt | | I/O interface (slot 203 on 2116) | as per interface | as per interface | as per interface | as per interface | **10** | corresponding I/O interrupt | | **. . .** | " | " | " | " | **. . .** | " | | I/O interface (slot 218 on 2116) | " | " | " | " | **27** | corresponding I/O interrupt | |

Notes:

1. Clearing control bit on channel 0 clears the control bit of all devices from 6 up.
2. Address/channel select code of most recent interrupt. [ref: 2116B Class Vol. 2 / pg. 6-38]
3. DMA Control.   
   Control bit selects the Address register (0) or Length register (1).   
   **Address register**: input operation (bit 15=0) or output operation (bit 15=1) and starting address of transfer (bits 14:0).   
   **Length register**: negative of length (word count) of transfer (bits 13:0, bits 15 and 14 read as 0). The program loads the length register with the negative of the write count or negative of the maximum read count before the operation. The DMA hardware increments the register towards 0 during the operation. Upon completion of the operation the program can use the value read from the length register to calculate the number of words transferred.
4. DMA Control.   
   **Device & options register**: STC per word (bit 15), CLC at end (bit 13) and device select-code (bits 5:0). With bit 15 set, an STC operation is performed on the device interface for every word transfer. With bit 13 set, a CLC operation is performed on the device interface at the completion of the operation.

[DMA refs: 2100 documention, example program in Type 13181 Magnetic Tape I/O Interface manual, experimentation with the hardware]

1. Memory Protect and Memory Parity information is from 2100 documention, presumed applicable to 2116.
2. On the 2115/6 it is only possible to read the front panel switches. On the 2100s, an actual register (the S-register) exists which can be altered by the front panel switches or an output instruction, can be read by an input instruction, and is displayed on the front panel lamps. Not sure about the 2114.
3. Two sequential select-codes apply to one physical slot for use by some devices. [ref: 2116B Class Vol. 2 / pg. 6-1]

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **Software for HP21xx Series** |  | | |

### Basic Binary Loaders

In HP parlance, a Basic Binary Loader, also referred to as an Absolute loader, is a small program for loading Absolute Load Format (ABS) files. BBLs are intended for bootstrapping a processor, there will normally be a BBL residing in the top 64 words of memory.

Several disassembled and newly-commented versions of the old BBLs are included here. The machine code used for the disassembly was obtained from the "STANDARD SOFTWARE SYSTEMS Operating Manual" for the 2116A/2115A models (1967). These period BBLs are:

* Paper Tape Reader BBL
* Teleprinter BBL (using 12531A bit-banged serial I/O interface)
* 2020A/B Magnetic Tape BBL (for 7 track drives)

The following is 'new', but is just a slightly modified version of one of the originals:

* 12531B/C/D Async Serial I/O Interface BBL
* [Basic Binary Loaders Directory Archive](http://www.cs.ubc.ca/~hilpert/e/HP21xx/software/bbl.tgz) (.tgz)

#### ABS Format:

Absolute Load Format is an encoding format used with the HP2100 series for loading binary images into a machine. 'Absolute' refers to the encoding containing absolute addresses at which to load data, in contrast to the data being relocatable.

An Absolute Load file is a sequence of 0 or more blocks, followed by a trailer of 10 or more null bytes:

AbsFile = Block ... Block 0 0 0 0 0 0 0 0 0 0

where each block is a sequence of bytes:

Block = lenByt ignoreByt addrHiByt addrLoByt data1HiByt data1LoByt

... data{Len}HiByt data{LEN}LoByt checksumHiByt checksumLoByt

len = the number of data words in the block. The checksum is the sum of the address word and the data words.

The trailer is necessary as an EOF indication for the BBLs. The null bytes of the trailer correspond to unpunched paper tape.

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | **Magnetic Core Memory Systems** |  | | |

|  |
| --- |
| <http://www.cs.ubc.ca/~hilpert/e/coremem/bwa2.jpg>  Core memory as wall art. This is a [Burroughs 3-wire planar array](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#modules) comprising 32,768 words of 20 bits/word. |

For roughly a quarter of a century, from the mid-1950s to the late-1970s, the word "core" used in relation to computers was a synonym for main memory or RAM.

If you have looked at a core memory plane, or a picture of one (or need to repair one in the absence of service info), and wondered just why the cores and wires were arranged in such patterns, this article may provide some answers. For those old enough to have direct professional experience with core-based systems the following may be 'old hat', for those too young it may fill in some gaps in the common literature.

This article is based primarily on the examination of several core memory [modules](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#modules), along with some written [references](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#references). It covers the basic principles, functional electrical arrangement and physical topologies of the predominant forms of core memory systems: the 3-dimensional 4-wire and 3-D 3-wire systems. It does not claim to be a comprehensive review of the technology as there were other techniques and organisations used in addition to those described here.

|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/key.gif | |

### 1. Brief History

Core memory was developed in the late-1940s/early-1950s as part of the Whirlwind computer project at MIT. Whirlwind had started out during WWII as a 'configurable' hydraulic-servo-based aircraft simulator and migrated over the years into a digital electronic real-time processing system. It eventually found it's raison-d'etre as a proof-of-concept for what would become the SAGE continent-wide nuclear-attack bomber-detection system.

Whirlwind became very expensive as it progressed and it's justification for existence in the face of all the other U.S. federally-funded computer projects - it's distinguishing characteristic - was it's real-time processing capabilities. It had a relatively small word size for the day (16 bits) and a hardware multiplier to help meet the real-time objectives, and it likewise needed a fast read/write memory. Unfortunately, the holding-beam form of storage-tube memory the developers of Whirlwind had produced for the machine was proving to be unreliable and required frequent maintenance.

In 1949 Jay Forrester, the director of the Whirlwind project, set about to find a better form of memory. The long average access times of the other popular memory systems of the day - acoustic delay lines and magnetic drums - made them inadequate for Whirlwind. Recent developments in ferro-magnetic materials caught his attention and he quietly began to explore their potential as a two-state memory. A small core-memory test system was constructed and was successful in showing core's viability. In 1953 the storage-tube memory was discarded and Whirlwind was converted to using core memory.

Core quickly became the predominant form of main memory for computers. Other machines, such as the IBM 701 and the NORC (Naval Ordnance Research Calculator), also had their storage-tube memory replaced with core. Core memory rode through the transitions of logic technology from vacuum tubes to transistors to SSI ICs and weathered competition from thin-film and plated-wire memory technologies in the 1960s. It was finally replaced by semiconductor LSI technology during the mid-to-late-1970s.

Core continued to see use for many years in some special-requirements systems, where use over time helped prove design reliability and where the non-volatility of core might be used to advantage, such as for speedy recovery of state in real-time systems. I have seen it in service as recently as 2004 in a telephony control application, the design and construction of which dates from the late-1970s/early-1980s, and it still functions in a few old systems maintained by computer museums and collectors. It's legacy lives on in the so-named "core" files produced on UNIX systems when a program crashes.

### 2. Basic Principles

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/magnetising.gif |

A doughnut shaped core of ferrite material can be placed into either of two saturated magnetic states. Current pulses in a wire going thru the core can be used to set the magnetic state of the core, based on the magnitude and direction of the current (figure 2a).

The hysteresis-loop graph in figure 2a describes the response of the magnetic field of the core to the current in the wire through the core. As a memory device the benefit of hysteresis is the bistable characteristic, significantly reducing noise sensitivity of the device and eliminating state decay and the consequent need for refreshing. The points +/-Rem on the hysteresis graph represent the remanent magnetic field of the core when no current is passing through it, i.e. the two storage states. +/-Is are the current levels required to bring the magnetic state of the core to saturation.

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/coincidence.gif |

The materials used in memory cores are selected to produce as near a rectangular loop as possible. With this characteristic sending only 1/2 the saturation current through a core will leave it's magnetic state unaffected. With a number of cores threaded into a 2-dimensional array (figure 2b) only the core at the coincidence of two wires with 1/2-currents flowing in them will be affected, the other cores on the two wires receiving insufficient current. This structure reduces the number of address decoder outputs and drivers required to address a given number of bits to a square-root relation. Figure 2c shows how a core can be set to either of it's (saturated) magnetic states with appropriate currents. The orientation of the core relative to the current directions is significant: the currents must sum to produce the saturation level. The assignment of logic values vs. current direction is arbitrary.

To read the 'memory', the core will be forced into a given state (figure 2d). If the core was in the opposite state, the reversal of the core's magnetic field constitutes a changing magnetic field which will induce a detectable current pulse in a third wire threaded through the core (referred to as the sense wire). If the core was already in the given state, while there is some perturbation of it's magnetic field by the addressing currents, this perturbation and the current induced by it are smaller and distinguishable from those of a field reversal. Sensing issues are dealt with further in the [Sensing section](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#sensing).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/set.gif | |  | |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/sense.gif | |

### 3. Addressing a Bit - The Core Plane

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/bitarray.gif |

Figure 3a shows a simple 8 by 4 array of cores (referred to as a bit-array or bit-plane) and the associated hardware modules to address these cores. The drivers have three inputs: an enable (E) input and positive and negative inputs to select the direction of current sent through the address wire. An arbitrary core can be set to either of it's two saturated magnetic states by setting up an address and pulsing the desired SET0 input or SET1 input.

This construct can be found in use for arrays of small dimensions but for larger arrays it is common to make the address decoders simpler and reduce the number of drivers by taking advantage of the ability to control both ends of an address wire. When considering one axis of n address wires, rather than using a 1-of-n decoder and n drivers, the axis of address wires can be viewed as an n=j\*k array, and a 1-of-j decoder, a 1-of-k decoder and j+k drivers used.

For example, figure 3b shows one 64-wire axis reduced to two 1-of-8 decoders (8\*8=64) and sixteen drivers, rather than one 1-of-64 decoder and sixty-four drivers. In this scenario there are two types of drivers, eight (upper) which will select the current direction and eight (lower) which will permit current flow in either direction.

In practice, a wide variety of decoder/driver designs were produced and aspects of real implementations may differ from those presented here in terms of control signal organisation, power supply configuration, etc. Of particular note, pulse transformers were widely used in the decoder/drivers.

|  |
| --- |
| http://www.cs.ubc.ca/~hilpert/e/coremem/reduce.gif |

Note the preceding diagrams show the functional electrical organisation of addressing a bit-array, the actual physical organisation adds some twists (literally) and is discussed in the following section.

### 4. Addressing a Word - The Core Stack

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/stack.gif |

So far, addressing just a single bit in an address space has been discussed. To produce a word- or byte-structured memory the above constructs could simply be repeated, once for each bit in a word. The downside to this is the large number of address-wire drivers required. Instead, the address wires are typically run through each of the bit-arrays of a word as in figure 4a.

It is not difficult to see this produces a problem: setting the state of one bit sets all the bits of that word to the same state. The usual remedy for this is to run another wire through the cores, referred to as the inhibit wire. Each bit-array will have it's own inhibit wire and driver (figure 4b).

When writing a word, if a bit of that word is not to be altered the according inhibit wire will carry a current opposed to that of the parallel address wire, thus inhibiting the address wire's influence by reducing the sum current through the addressed core to below the saturation level.

Alternating address wires are fed from opposite directions, which matches the back-and-forth weave of the inhibit wire. The orientation of half the cores is rotated 90 degrees so they still catch the current-direction combinations which sum to produce the saturation level.

The sequences involved in writing are discussed further in the [Read/Write Cycles and Interfacing section](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#interfacing).

Figure 4a also gives a graphic indication of the source of the term 'core stack'. In practice, while there were early implementations that followed this structure physically, later implementations generally reduced the stack height by placing multiple bit-arrays in the same physical plane (figure 4c), referred to as a planar array.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/inhibit.gif | |  | |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/planar.gif | |

### 5. Addressing with Anti-coincidence

While searching for a problem in an 8 KWord core memory module for an HP2116C computer I was perplexed when I could find only enough address drivers to address 4 KWords. How was it that another 4K was being addressed?

Notice that with 2 address wires going through each core and 2 possible directions of current through each wire there are 4 possible combinations of current through each core. However, only 2 combinations are used: the 2 which go through the core from the same side (are coincident) and so sum to produce the saturation level. In the other 2 combinations (anti-coincident) the address wire currents cancel each other and have no effect on the core.

If another set of cores is added to a bit-array in such a manner that one axis of address wires goes through the new cores from the other direction (the X-axis in figure 5), then this new set will respond to the other 2 current combinations while ignoring the original 2 combinations, and vice-versa for the original set. The combinations are shown in table 5, where "+" and "-" indicate the direction of address-wire currents from the drivers.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/anticoincidence.gif | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | **Table 5:** Anti-Coincident Current Combinations | | | | | **X** | **Y** | **Original Core** | **New Core** | | - | - | 0 | n.a. | | + | + | 1 | n.a. | | + | - | n.a. | 0 | | - | + | n.a. | 1 | | n.a. = not affected | | | | | |

To address the new cores the current direction on the redirected axis of address wires must be reversed. The drivers are already capable of sending current in either direction, so all that is required is to add a few logic gates to swap the influence of the SET1 and SET0 signals on the one axis of address-wire drivers. If the "new ADDRESS BIT" signal is 0 the original cores will be addressed; if 1, the new cores are addressed.

### 6. Sensing

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/sense2by2.gif |

Recall from the [Basic Principles section](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#principles) that to read the state of a core it will be forced into a given state and current detected in a sense wire will indicate whether a reversal of the core's magnetic field occurred. However the currents in the address wires forcing the (maybe) core field reversal will also induce currents in the sense wire and these must be distinguished from the current induced by a field reversal.

To this end the sense wire is woven at 45 degrees to the address wires to minimise induction and in a back-and-forth manner such that the undesired currents cancel, leaving the current induced by the field reversal. Looked at another way, the number of crossings of a given address wire by the sense wire is even, with half the crossings being in one direction and half in the opposite direction. Figure 6a shows this for a minimal 2 by 2 array. The sense wire is shown in two colours for clarity although it is electrically a single continuous wire. As with the inhibit wire, each bit-array has it's own sense wire.

None-the-less, induced currents from irregularities in the weaving and from perturbations of the magnetic fields of 'half-selected' cores, along with capacitive coupling, will produce noise on the sense wire and presumably this placed an upper limit on the size of a bit-array.

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/sense14by8.gif |

Figure 6b shows the sense wire weave of the 14 by 8 array from an early calculator. Weaving patterns can get considerably more complex for larger arrays as shown by that of a [96 by 100 array](http://www.cs.ubc.ca/~hilpert/e/coremem/sense96by100.gif). An entirely different sense wire weave is discussed in the [3-wire section](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#3wire).

A consequence of this weaving pattern and the core orientations is that a pulse induced in the sense wire may be either positive or negative and the sense amplifier must be able to respond equivalently (produce a logically identical output pulse) for both possibilities. Sense amplifiers were constructed from either discrete components, standard voltage comparator integrated circuits or specialised ICs and generally include a STROBE input so that field reversals occurring when not sensing can be ignored. Figure 6c shows some real-world examples.

|  |
| --- |
| http://www.cs.ubc.ca/~hilpert/e/coremem/senseamps.gif |

### 7. 3-Wire Systems

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/burroughs.gif |

The system described so far is referred to as a 4-wire system, an obvious result of the four wires going through each core. 3-wire systems were also developed, to simplify the construction process (which was largely done by hand) and to permit smaller cores to be used.

Observe that when reading a word the inhibit wire is not used as reading entails setting all the bits of a word to the same state, and when writing the sense wire is not used. The functions of the sense and inhibit wires can be combined into a single wire. Figure 7a shows the organisation of the X=128 by Y=256 bit-arrays used in a Burroughs 3-wire module.

Note there are two inhibit drivers, sense amplifiers and sense/inhibit wires used for each bit-array. This is not really related to the 3-wire design, presumably it is a consequence of the large size of the bit-array producing noise issues beyond the capability of a single sense amplifier to tolerate. There are 32,768 cores in each bit-array of this module, so 16,384 cores per sense amplifier.

When sensing, the sense/inhibit (S/I) wire looks like the usual loop (see also figure 7c). It runs closely parallel to an address wire for blocks of 64 cores, but the transpositions of the S/I wire results in it parallelling the same address wire from the opposite direction for another 64 cores. For example, in figure 7a, an S/I wire (red segment) runs closely parallel to address wire X0 over the distance Y192 to Y255. The same S/I wire (blue/green segment) runs closely parallel to the same address wire over the distance Y128 to Y191, but is now coming from the opposite direction. Currents induced in the S/I wire from wire X0 thus cancel.

In addition, when sensing, the timing of the address current pulses appears to have been set such that the current on X-axis wires (parallel to the S/I wire) is turned on slightly before that on the Y-axis (perpendicular to the S/I wire). A core field reversal will be triggered by the later leading edge of the pulse on the Y-axis. Referring to figure 7b, current transitions on the address wires induce pulses in the S/I wire (X and Y pulses), the Y pulses being smaller because they result from the perpendicular Y-axis wire. As the pulses are separated in time they do not sum and their level stays below the threshold for the sense amplifier. A core field reversal produces a larger pulse (R), enough to trigger the sense amplifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/burroughstime.gif | |  | |  | | --- | | http://www.cs.ubc.ca/~hilpert/e/coremem/burroughssense.gif | |

When inhibiting, the S/I wire looks like two parallel elements. The sense amplifier ends are terminated with a low enough impedance to ground to permit the inhibit currents through (figure 7c). The paired alternation of the X-axis drivers matches the two 'inhibit elements' of the S/I wire. When an inhibit driver is activated, two cores on a given Y address wire will see inhibit currents. For example, suppose the core (X0,Y0) is to be inhibited during a write operation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Core (X0,Y0) will see currents: | (X=+1/2) | + (Y=+1/2) | + (I=-1/2) | = +1/2 |
| Core (X1,Y0) will see currents: | (X=0) | + (Y=+1/2) | + (I=-1/2) | = 0 |

Both cores are left in their current state, as is desired.

Another approach to a 3-wire system, referred to as 2-1/2 D system, is to weave only one of the address axis wires through the stack, and have separate drivers for the other axis of each bit-array (still at the cost of a lot more drivers).

### 8. Read/Write Cycles and Interfacing

|  |  |
| --- | --- |
|  | http://www.cs.ubc.ca/~hilpert/e/coremem/interface.gif |

The techniques described so far result in a word based, randomly accessible, arbitrary-sized (within limits) memory, which, for the technology, makes quite efficient use of components. Some quirks of core memory still have to be dealt with to make it practical for use in a processing system.

The read signals coming out of the sense amplifiers are very brief pulses corresponding to the field reversal of a core so they need to be latched into a register where they can be held for use (the one-word register in figure 8).

Reading a word is destructive - leaving all the cores (bits) in that word in the same state - so the data must be rewritten. When writing a word the first step will be to place the cores of that word into the same state so that some cores may then be changed to the other state while the remainder are inhibited from changing. Consequently a read cycle and a write cycle become almost identical operations:

1. If a read cycle, clear the word register.
2. Pulse the desired X and Y address wires, with the current directions set appropriately for sensing. The cores are set to 0. If a read cycle, the STROBE signal is enabled so that sense pulses generated by the field reversal of cores which had 1 stored in them set the according bits in the word register to 1.
3. Enable current in the inhibit wire of cores which are to stay 0.
4. Pulse the desired X and Y address wires with the current directions set appropriately for setting, setting the uninhibited cores to 1.

As many computer instructions modify an operand in memory, and it would be wasteful of time to execute a full read cycle followed by a full write cycle, some systems incorporated a third type of cycle: the read/modify/write cycle, in which the data is read from core and modified by the CPU before being re-written.

Systems typically include a small state machine to generate control signals in the appropriate sequences for the various memory cycles.

Over the years the cores got smaller, switching times became faster, and array sizes increased. All this required tighter tolerances for address pulse timing and current levels. More sophisticated systems include circuitry for regulating and controlling these parameters. Temperature compensation of these parameters may also be included as characteristics of the ferrite core material change with temperature.

Core memory is non-volatile (does not lose it's state when powered off) and many systems took advantage of this feature for system recovery or to simplify the boot procedure. However, the destructive read produces a problem: data may be lost if power fails after the sense operation but before the data is re-written. To deal with this, systems may include special circuitry to ensure the completion of a memory cycle when power-fail is detected.

### 9. Specialised Components

The requirements of core memory resulted in the production and use of some specialised components (in addition to the cores themselves).

* The diode arrays used for steering address driver currents may be found in DIP IC packages (generally identifiable by the absence of direct power supply connections) or potted modules.
* Multiple driver transistors may be found in DIP packages, typically quad NPN arrays.
* Pulse transformers were used liberally in many systems and may be found in packages that look like potted DIPs or other small potted modules.
* Specialised ICs were developed for sense amplifiers.

### 10. Some Physical and Electrical Characteristics

A book from the 1950s [[ref:6]](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#references) provides the following examples of physical characteristics for cores:

Core sizes vary, some smaller ones having an outside diameter of only 0.08 inch and an inside diameter of 0.05 inch. Larger ones range up to 0.4 inch outside diameter, . . .   
. . .  
The [sense pulse ranges] in amplitude between 50 and 75 millivolts. . .   
. . .  
Switching time ranges from 1 microsecond to 5 microseconds, depending on the type and quality of the ferrite core. Driving current ranges from 400 to 800 milliamperes.

Cores were reduced in size by another factor of 5 or so over the years, a Burroughs module from the late 1970s has cores of approximately 0.015 or 1/64 inches outside diameter.

### 11. Some Real-World Modules

Table 11 presents some physical details of actual core memory modules. These represent a wide range of practical core implementations. They don't get smaller than that required by a simple calculator, but they did get larger than the 655,360 cores of the Burroughs module, on the order of 2,000,000 cores in a planar array according to [[ref:5]](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#references).

Module here means a group of cores woven together with the same set of address wires, together with their support circuitry. Memory systems of larger storage capacity were of course constructed from multiple such modules.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Table 11:** Some Examples of Core Memory Modules | | | | | | | | |  | **Machine** | **Year** | **Form** | **Intention** | **Bit Arrays** | **Bit Array Dimensions** | **Number of Cores** | | Small | Casio AL-1000 Calculator | 1967 | 4-wire | 7 registers of 16 BCD digits | 4 | 14 \* 8 = 112 | 448 | | Medium | Hewlett-Packard 5480A Digital Signal Analyser | 1971 | 4-wire | 1024 words of 24 bits | 24 | 32 \* 32 = 1024 | 24,576 | | Medium | (unknown, German origin) | 1974 | 4-wire | 9600 words of 4 bits | 4 | 96 \* 100 = 9600 | 38,400 | | Large | Hewlett-Packard 2116C Computer 8 KWord memory module | 1970 | 3-wire anti- coincident | 8192 words of 16 bits with parity | 17 | 2 \* 64 \* 64 = 8192 | 139,264 | | Very Large | Burroughs module (machine unknown) | 1978 | 3-wire | 32,768 words of 20 bits | 20 | 128 \* 256 = 32,768 | 655,360 | |

### 12. References

1. Examination and reverse engineering of the [above-mentioned modules](http://www.cs.ubc.ca/~hilpert/e/coremem/index.html#modules).
2. Computer Maintenance Course Students Manual HP 2116B / HP 2115A CPUs, Hewlett-Packard Co., 1969. (online at [bitsavers.org](http://www.bitsavers.org/).)
3. HP 2100A Computer Diagrams Manual, Hewlett-Packard Co., 1971. (online at [bitsavers.org](http://www.bitsavers.org/).)
4. Project Whirlwind: The History of a Pioneer Computer, Kent C. Redmond and Thomas M. Smith, Digital Press (DEC), 1980, TK 7889 W47 R43
5. Ferrite-Core Memories, by Ronald A. Hill of Ferroxcube Corp., article in Electronics World, October 1970, Ziff-Davis Publishing Co.
6. Fundamentals of Digital Computers, by Matthew Mandl, Prentice-Hall Inc., appears to be from mid-late-1950s.
7. Teach Yourself Electronic Computers, by F. L. Westwater O.B.E., English University Press Ltd., London, 1962.
8. General Information Manual, Introduction to IBM Data Processing Systems, IBM Corp., 1964.
9. Physics, Paul A. Tipler, Worth Publishers Inc., 1976.
10. Coincident Current Ferrite Core Memories, by James R. Jones, article in Byte Magazine, July 1976, Byte Publications Inc.
11. [Columbia U. NORC pages](http://www.columbia.edu/acis/history/norc.html).
12. [IBM museum 701 pages](http://www-03.ibm.com/ibm/history/exhibits/701/701_1415bx37.html).
13. [USN Tutorial](http://ed-thelen.org/comp-hist/navy-core-memory-desc.html).