

ATLAS

Christine Sirkhan
Matt O'Connor

History of ATLAS

The Atlas computer made its start in Manchester, during the early 1960s. It was the result of a project, which began in the Department of Electrical Engineering of the University of Manchester in 1956. The group was a combination of electronic engineers investigating the new Transistor switching circuits and Magnetic Core Storage devices, with the programmers who were familiar with the problems and aspirations of using earlier Manchester computing machines for scientific calculations. By 1959 Ferranti Ltd. (a major UK electrical engineering and equipment firm) had joined the project and their collaboration with the University led to the development of the ATLAS computer. The Manchester ATLAS began to provide a computer service in 1962. It went on to provide a reliable service for both scientific and commercial users until 1971.

Pioneer in OS concepts

- Timesharing of several concurrent computing and peripheral operations,
- Multiprogramming
- Interleaved stores
- Paging
- V-Store (Image Store)
- Fixed store (ROM)
- Autonomous transfer units.

Memory Organization

Word

1 word is 48 bits.

½ words (24 bits) stored with a parity bit (50-bit transfers from core).

Core Store

4 stacks of 4096 words (16384 total words)

Divided into 512-word blocks

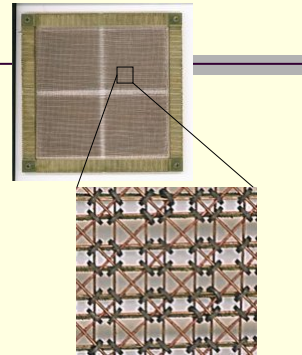
→ 32 pages of 512-words each

Each 512-word block/page is divided between two core

stacks, such that sequential addresses are separated.

Example:

- Address n, n+1
- n --> stack 0
- n+1 --> stack 1
- n+2 --> stack 0
- n+3 --> stack 1
- ...



Consists of a large number of small ferrite (ferromagnetic ceramic) rings, *cores*, held together in a grid structure with wires woven through the holes in the cores' middle. In early systems there were four wires, X, Y, *Sense* and *Inhibit*, but later cores combined the latter two wires into one *Sense/Inhibit* line. Each ring stores one bit (a 0 or 1), so many cores are needed to provide a reasonable amount of memory. Each *plane* stores one bit of an array of machine words, the full word was provided by a *stack of planes*.

Memory Organization

Drum Stores

Each drum stored 24,576 words (x4 drums = 98,304 words)
 First track of each drum contained absolute locations of stored blocks.



*A drum is a large metal cylinder that is coated on the outside surface with a ferromagnetic recording material. It is, simply put, a hard disk platter in the form of a drum rather than a flat disk.

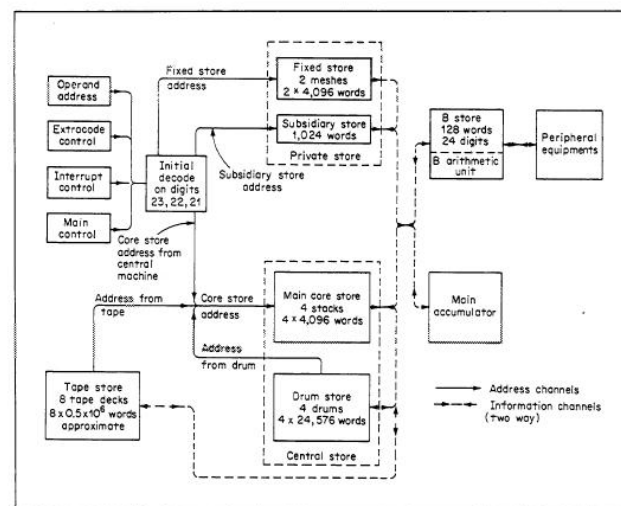
Subsidiary store

Used to retain the locations of all blocks that are placed on the drums. In writing a page onto a drum, the block location is stored with the block itself. When loading the page back into the core, the block location is also reloaded. This provides redundancy for the block locations being loaded, and prevents an incorrect block from being loaded into the core.

System Layout

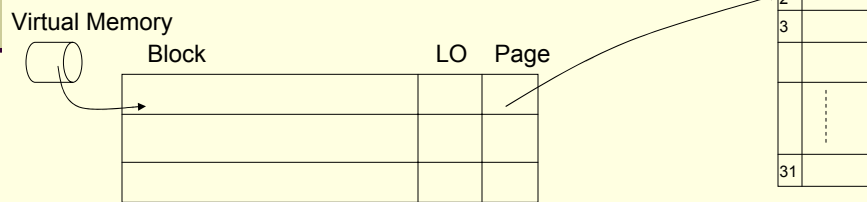
Addresses are formed as 11-bit block addresses, plus 9 bit offsets (termed "position address").

The 12th block-bit (bit 21 of full address) selects between "main memory" (core + drum), and "executive store" (read-only memory containing Supervisor + other prog)



Page Address Register

Contains the current page of all currently loaded blocks, and the block addresses. Block requests are mapped Block --> Page. If the page is not in core, it is loaded and a victim is selected for removal if the core has been made full. PAR checks are performed in parallel. A lock-out bit is used to prevent access to a page that is being loaded into memory. It can also be used to prevent access to pages not in a program's operating space.



Supervisor and Extra Functions

These exist in the *extracode* region, which is a special read-only store that has its own subsidiary store of 1024 words as working-space for functions described in the *extracode* region. Access to this region is performed by a jump to an address prefixed by a special bit (21st addressing bit).

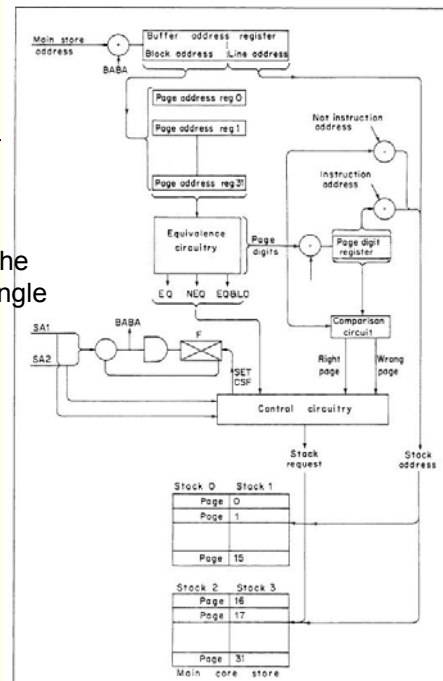
Extracode contains large number of built in subroutines which provide:

1. A number of orders which would be expensive to provide in the machine both in terms of equipment and also time because of the extra loading on certain circuits. An example of this is the order:
2. Shift accumulator contents $\pm n$ places where a is an integer.
3. The more complex mathematical operations, e.g., $\sin x$, $\log x$, etc.,
4. Control orders for peripheral equipments, card readers, parallel printers, etc.,
5. Input-output conversion routines,
6. Special programs concerned with storage allocation to different programs being run simultaneously, monitoring routines for fault finding and costing purposes, and the detailed organization of drum and tape transfers

Main core store control

Control signals

SA1 and **SA2** indicate whether the address presented is that of a single word or a pair of sequentially addressed instructions.



Core and Drum Paging

- As long as there are free pages in the core, no pages need to be swapped out to the drum stores.
- When the store becomes full, a page is swapped to the drum to ensure a
 - free page is always available for the next read-page request.
- A “learning” program is used to determine which block to remove from core and place onto the first available drum location.
- Two ‘time’ values are maintained per page.
- Timing is performed in terms of instruction counts, rather than a clock time, in order to prevent long I/O operations from affecting the “learning curve”.

“Learning” program

The first value T is the length the last period of inactivity for a given block. The other value t is the length of time since the block in that page has been used. Thus, T is updated to t whenever an access to the block occurs, and t is reset to zero. The selection algorithm runs three tests in order:

- 1) Any page for which $t > T + 1$
- 2) That page with $t \neq 0$ and $(T - t)$ max
- 3) That page with T max (all $t = 0$)

These are applied in order.

- The first rule selects a block such that it has not been accessed in a period of time longer than its last period of inactivity.
- The second rule selects a block such that the block is one which will not be required by the program for the longest period of time (thus, this is a prediction to the future).
- The third rule is a catch-all that will simply select the block with the longest duration of prior inactivity (least-used).

“Learning”

T is initialized when a block is loaded off the drum and into the core as:

ζ = time-of-transfer – value-of-t-for-transferredpage

When the block is brought into the core:

$T = \text{time-of-transfer} - \zeta$.

$t = 0$

*It should be noted that the read operation (drum to core) could be initiated before a block was selected for removal from the core.

Summary

- Privileged Instructions: ATLAS was one of the first systems developed that implemented a private store for supervisor only calls.
- Paging: ATLAS uses paging to make a two level storage system appear to the programmer as if it were one.
- Interleaved Storage: The core has 4 stacks, in which sequential words are interleaved between 2 of the stacks making it possible to read two words in parallel.
- Paging Victim Selection: Implemented with a Learning program that keeps track of the last period of inactivity & the length of time since the block in the page has been used to determine/predict the next victim.

Conclusions about ATLAS

- Makes a core-drum store combination appear as a single level store (not apparent to user, and the programmer need not worry about it).
- The automatic system requires additional equipment and introduces some new complexity.
- No matter how intelligent a programmer may be he can never know how many programs or peripheral equipments are in operation when his program is running. Making this design and the combination of the 'learning' program a very optimal choice for its time (1960's).

References

- Fotheringham, J., Dynamic Storage Allocation in the Atlas Computer, Including an Automatic use of a Backing Store
- Kilburn, T., Edwards, D.B.C., Lanigan, M.I., and Sumner, F.H., One-level Storage System IRE Trans., EC-11, vol. 2, April 1962, pp. 223-235.
- History of the Atlas
http://en.wikipedia.org/wiki/Atlas_computer