

Content-addressable memory

Content-addressable memory (**CAM**) is a special type of computer memory used in certain very-high-speed searching applications. It is also known as **associative memory**, **associative storage**, or **associative array**, although the last term is more often used for a programming data structure.^[1] It compares input search data (tag) against a table of stored data, and returns the address of matching data (or in the case of associative memory, the matching data).^[2] Several custom computers, like the Goodyear STARAN, were built to implement CAM, and were designated **associative computers**

Contents

- Hardware associative array**
- Standards for content-addressable memories**
- Semiconductor implementations**
- Alternative implementations**
- Ternary CAMs**
- Example applications**
- See also**
- References**
- Bibliography**
- External links**

Hardware associative array

Unlike standard computer memory (random access memory or RAM) in which the user supplies a memory address and the RAM returns the data word stored at that address, a CAM is designed such that the user supplies a data word and the CAM searches its entire memory to see if that data word is stored anywhere in it. If the data word is found, the CAM returns a list of one or more storage addresses where the word was found (and in some architectures, it also returns the contents of that storage address, or other associated pieces of data). Thus, a CAM is the hardware embodiment of what in software terms would be called an associative array. The data word recognition unit was proposed by Dudley Allen Buckin 1955.^[3]

Standards for content-addressable memories

A major interface definition for CAMs and other network search engines (NSEs) was specified in an interoperability agreement called the Look-Aside Interface (LA-1 and LA-1B) developed by the Network Processing Forum, which later merged with the Optical Internetworking Forum (OIF). Numerous devices have been produced by Integrated Device Technology, Cypress Semiconductor, IBM, Broadcom and others to the LA interface agreement. On December 11, 2007, the OIF published the serial lookaside (SLA) interface agreement.

Semiconductor implementations

Because a CAM is designed to search its entire memory in a single operation, it is much faster than RAM in virtually all search applications. There are cost disadvantages to CAM however. Unlike a RAM chip, which has simple storage cells, each individual memory bit in a fully parallel CAM must have its own associated comparison circuit to detect a match between the stored bit and the input bit. Additionally, match outputs from each cell in the data word must be combined to yield a complete data word match signal.

The additional circuitry increases the physical size of the CAM chip which increases manufacturing cost. The extra circuitry also increases power dissipation since every comparison circuit is active on every clock cycle. Consequently, CAM is only used in specialized applications where searching speed cannot be accomplished using a less costly method. One successful early implementation was a General Purpose Associative Processor IC and System^[4]

Alternative implementations

To achieve a different balance between speed, memory size and cost, some implementations emulate the function of CAM by using standard tree search or hashing designs in hardware, using hardware tricks like replication or pipelining to speed up effective performance. These designs are often used in routers.

An alternative approach to implementation is based on Superimposed Code Words or Field Encoded Words which are used for more efficient database operations, information retrieval and logic programming, with hardware implementations based on both RAM and head-monitoring disk technology^{[5][6]}

Ternary CAMs

Binary CAM is the simplest type of CAM which uses data search words consisting entirely of 1s and 0s. **Ternary CAM** (TCAM)^[7] allows a third matching state of "X" or "don't care" for one or more bits in the stored dataword, thus adding flexibility to the search. For example, a ternary CAM might have a stored word of "10XX0" which will match any of the four search words "10000", "10010", "10100", or "10110". The added search flexibility comes at an additional cost over binary CAM as the internal memory cell must now encode three possible states instead of the two of binary CAM. This additional state is typically implemented by adding a mask bit ("care" or "don't care" bit) to every memory cell.

Example applications

Content-addressable memory is often used in computer networking devices. For example, when a network switch receives a data frame from one of its ports, it updates an internal table with the frame's source MAC address and the port it was received on. It then looks up the destination MAC address in the table to determine what port the frame needs to be forwarded to, and sends it out on that port. The MAC address table is usually implemented with a binary CAM so the destination port can be found very quickly, reducing the switch's latency

Ternary CAMs are often used in network routers, where each address has two parts: the network prefix, which can vary in size depending on the subnet configuration, and the host address, which occupies the remaining bits. Each subnet has a network mask that specifies which bits of the address are the network prefix and which bits are the host address. Routing is done by consulting a routing table maintained by the router which contains each known destination network prefix, the associated network mask, and the information needed to route packets to that destination. Without CAM, the router compares the destination address of the packet to be routed with each entry in the routing table, performing a logical AND with the network mask and comparing it with the network prefix. If they are equal, the corresponding routing information is used to forward the packet. Using a ternary CAM for the routing table makes the lookup process very efficient. The addresses are stored using "don't care" for the host part of the address, so looking up the destination address in the CAM immediately retrieves the correct routing entry; both the masking and comparison are done by the CAM hardware. This works if (a) the entries are stored in order of decreasing network mask length, and (b) the hardware returns only the first matching entry; thus, the match with the longest network mask (longest prefix match) is used.^[8]

Other CAM applications include:

- CPU fully associative cache controllers and translation lookaside buffers
- Database engines
- Data compression hardware
- Artificial neural networks
- Intrusion Prevention System

See also

- [Associative array](#)
- [Content addressable network](#)
- [Content Addressable Parallel Processor](#)
- [Content addressable storage](#) or file system
- [Sparse Distributed Memory](#)
- [Tuple space](#)

References

1. Hannum et al. (2004)
2. Pagiamtzis and Sheikholeslami (2006), pp. 712–713
3. TRW Computer Division(<http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD408276&Location=U2&doc=GetTRDoc.pdf>) Archived (<https://web.archive.org/web/20110805004452/http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD408276&Location=U2&doc=GetTRDoc.pdf>)August 5, 2011, at the [Wayback Machine](#), 1963, p. 17.
4. Storman et al. (1992)
5. Wise, Michael J; Powers, David M W (1984). "Indexing Prolog Clauses via Superimposed Code Words and Field Encoded Words.". *International Symposium on Logic Programming*203–210.
6. Colomb, Robert M (1991)."Enhancing unification in PROLOG through clause indexing"<http://www.sciencedirect.com/science/article/pii/0743106691900049>) *Journal of Logic Programming* **1** (10): 23–44. doi:[10.1016/0743-1066\(91\)90004-9](https://doi.org/10.1016/0743-1066(91)90004-9) (<https://doi.org/10.1016%2F0743-1066%2891%2990004-9>)
7. https://books.google.com/books?id=-rnt_ik0mSYC&pg=PR71&dq=TCAM&
8. Varghese, George *Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices* Morgan Kaufmann, 2005

Bibliography

- Anargyros Krikelis, Charles C. Weems (editors) (1997). *Associative Processing and Processors* IEEE Computer Science Press. ISBN 0-8186-7661-2
- Hannum et al.. (2004). *System and method for resetting and initializing a fully associative array to a known state at power on or through machine specific state* U.S. Patent 6,823,434. Available by searching at <http://patft.uspto.gov/netahtml/PTO/search-bool.html>
- Pagiamtis, K.; Sheikholeslami, A. (2006)."Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey"(PDF). *IEEE Journal of Solid-State Circuits* **41** (3): 712–727.
- Stormon, C.D.; Troullos, N.B.; Saleh, E.M.; Chavan, A.V.; Brule, M.R.; Oldfield, J.V; A general-purpose CMOS associative processor IC and system, Coherent Research Inc., East Syracuse, NY USA, IEEE Micro, Dec. 1992, Volume: 12 Issue:6.
- TRW Computer Division. (1963). *First interim report on optimum utilization of computers and computing techniques in shipboard weapons control systems* (BuWeps-Project RM1004 M88-3U1). Alexandria, Virginia: Defence Documentation Center for Scientific and Technical Information.

External links

- [CAM Primer](#)
- [Content-addressable memory \(CAM\) circuits and architectures: A tutorial and survey](#)
- [Aspex - Computer architecture built around associative memory](#)
- [Initiation of Serial Lookaside specification effort](#) (near bottom of page 1)
- [OIF Serial Lookaside interface agreement](#)
- [Arithmetic Processing using Associative memory](#)
- [SimpleRBM - a very small Restricted Boltzmann machine](#) including training algorithm, which is a kind of Content Addressable Memory of bit vectors.

This page was last edited on 14 October 2017, at 22:13.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.