

# Designing Switches & Routers with APEX CAM

## Introduction

Content-addressable memory (CAM) allows a design to quickly search a table for a particular item. The data is supplied by the user or the system, and CAM uses the information provided to determine the location of the requested data.

CAM is especially useful in speeding up search operations in switches and routers. This white paper describes how APEX™ CAM can be used in specific applications such as:

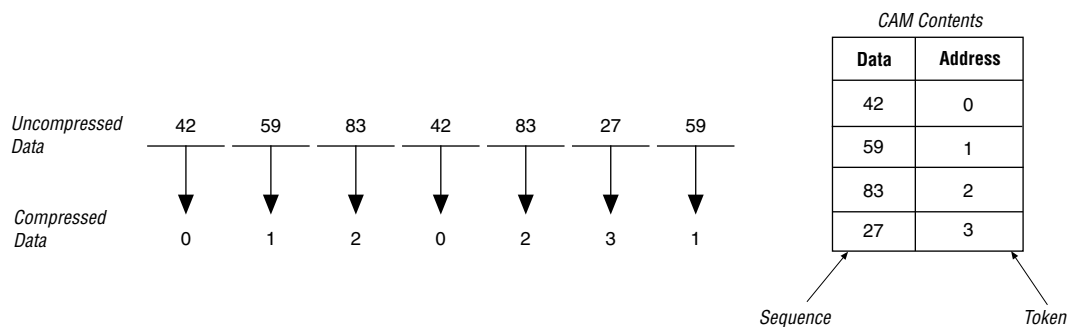
- Data compression
- Network switching
- Internet protocol filtering
- Internet protocol address resolution
- Multi-protocol label switching.

## Data Compression

Data compression increases system bandwidth by removing redundant information and allowing a faster flow of traffic. Most compression algorithm time is spent searching and maintaining data structures to manage the compression; however, replacing the data structure with a hardware search engine can increase the throughput of the algorithm.

In data compression, a shorter bit of data can replace a longer bit of data. For example, an eight-bit number can be represented with a two-bit number. These conversions are translated in CAM. The eight-bit number is compared against the data in CAM, and the addresses in the CAM block are two-bit condensed numbers. In [Figure 1](#), CAM receives the number 42 as data. This number is compared against all the data in the CAM block, then CAM finds the compressed result.

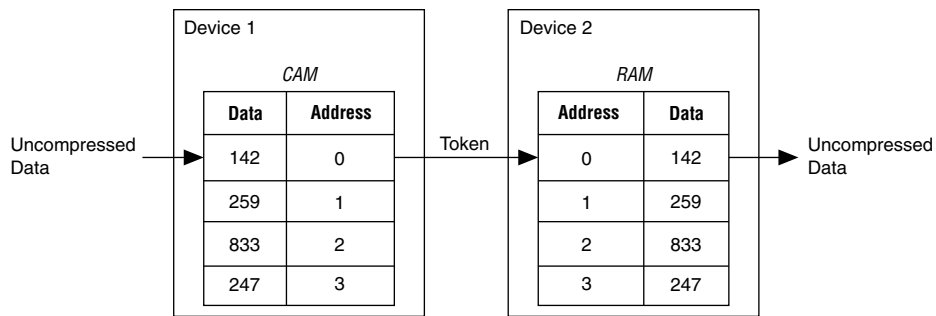
Figure 1. Using CAM for Data Compression



In one compression algorithm, commonly recurring sequences are replaced by tokens. Tokens are shorter than sequences, compressing the data stream. When a word is driven into the CAM block as data, the CAM block returns the address where the word was found. This address is now the token for the sequence. If CAM finds the code corresponding to the word-bit pattern in the input register, then it clears the appropriate symbol or token output and input register. If the code is not found in CAM, then another word is shifted in. The word that was not found can be stored with an escape code to indicate that it was not compressed. CAM generates a result in a single transaction, regardless of table size or length of a search list. This makes CAM ideal for data compression schemes that use sparsely populated tables as part of their algorithm.

RAM in another device can be used to uncompress the compressed data or token from the first device's CAM block. In RAM, the compressed data or token is saved as the uncompressed data's address. Figure 2 illustrates the compression and decompression of data in two devices through CAM and RAM.

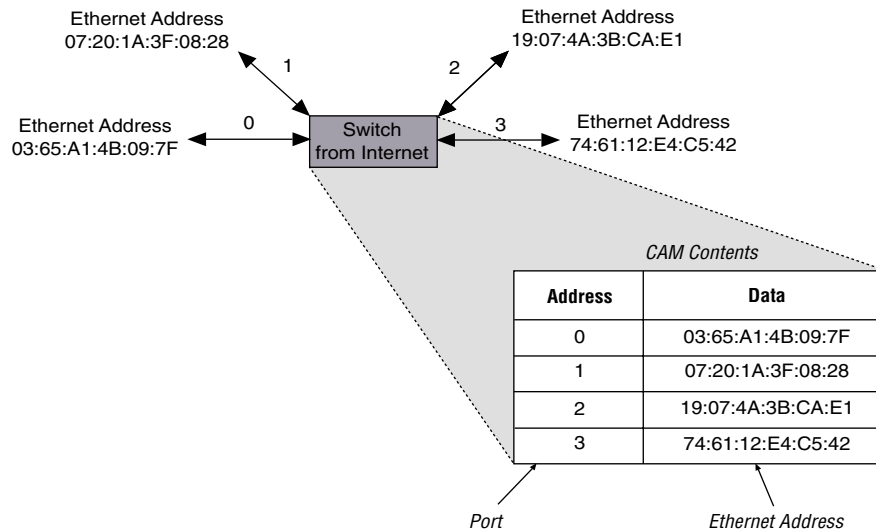
Figure 2. Compression & Decompression in Two Devices through CAM & RAM



### Network Switch

CAM can be used in switch applications to extract and process address information from incoming data packets. CAM compares the stored destination address in order to switch the packet to the correct outgoing port. In Figure 3, a network switch routes incoming packets to one of four destinations based on ethernet addresses and switch port numbers stored in CAM. CAM then compares the address of each received packet against the stored addresses and port numbers. If the comparison yields a match, routing control forwards the packet to the proper destination port.

Figure 3. Using CAM as a Network Switch

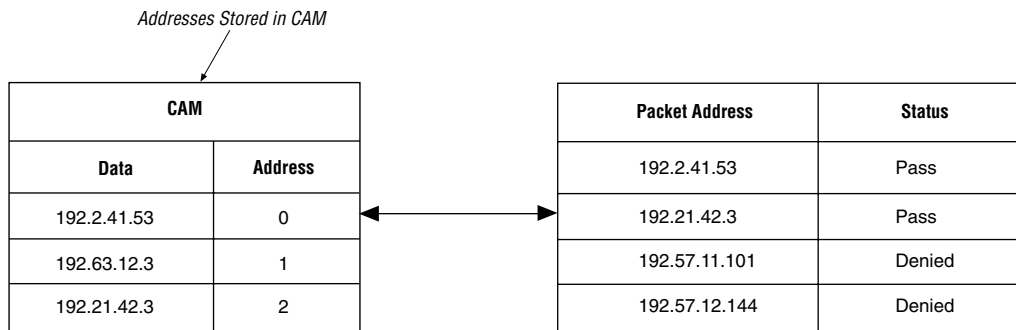


### IP Filter

An IP filter restricts unauthorized local area network (LAN) access or traffic on a wide area network (WAN). Restricting Internet traffic across a LAN allows the administrator to give the LAN workstations permission to specific Internet-based applications, such as e-mail. Filters can be implemented in both software and hardware to improve performance and control security.

CAM can improve the performance of the IP filter by storing the source or destination addresses of packets that are allowed to pass along the network. When CAM receives the address information, it compares the address to the list of stored addresses. If a match is found, the packet is sent through to its destination. If no match is found, the packet is not transmitted. Figure 4 displays an example of this process.

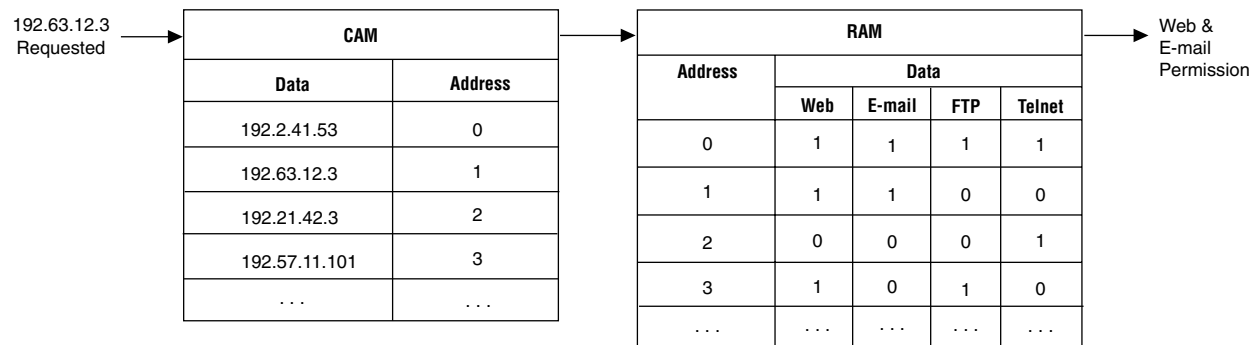
Figure 4. CAM as an IP Filter



If more than one application needs to access the LAN, a combination of CAM and RAM can perform a multiple application filter. CAM stores the source or destination address of a packet, and RAM stores the permissions. Each bit of RAM stores one specific permission.

Figure 5 shows an example with requested packets having explicit permission to access the Internet and e-mail.

Figure 5. Multiple-Permission IP Filter



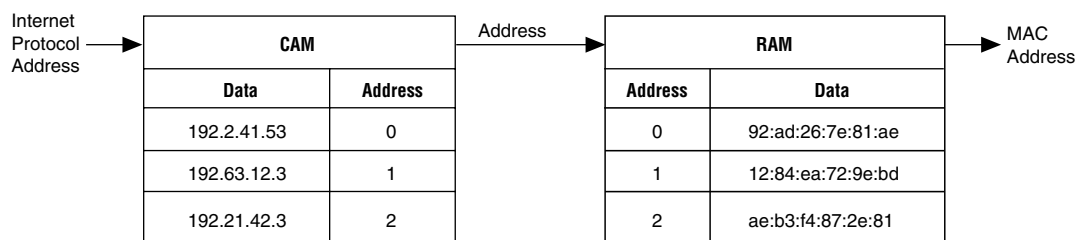
## Internet Protocol Address Resolution

Internet protocol address resolution is used by layer 3 switches to convert Internet protocol addresses to Ethernet or media access control (MAC) addresses and vice versa. An Internet protocol address is a 32-bit value that identifies each individual device on a network that is able to send or receive data. When a source attempts to send data over the Internet, the Internet protocol portion of the transmission control protocol/Internet protocol (TCP/IP) includes the source Internet protocol address and the destination Internet protocol address in the packet. Once all the required information for delivery is included in the packet, the TCP/IP stack sends the entire packet across the Internet to the destination. The destination receives the data, and a response can be sent to the source by using the existing Internet protocol address.

A MAC address, or Ethernet address, is the unique physical address designated to a hardware device (such as a personal computer) when it is manufactured. When a computer is connected to a LAN, a correspondence table relates the Internet protocol address of the device to its MAC address. A combination of CAM and RAM can implement a table to maintain a correlation between each MAC address and its corresponding IP address and perform address conversion in both directions. CAM contains the Internet protocol addresses, that point to corresponding locations in RAM, which contains the MAC or Ethernet address.

Address resolution protocol (ARP) is used to map an Internet protocol address to a recognized MAC address in the local network. When a host machine receives a packet, ARP runs a search to verify that the packet’s destination address is on its particular network. The CAM and RAM combination, implemented in the ARP, compares the Internet protocol address against the content of the CAM table to find the appropriate MAC address in RAM. If CAM contains the Internet protocol address, the packet can be converted to the appropriate length and format, then sent to the destination device. If no match is found, ARP broadcasts a request packet to all the devices on the LAN in search of the corresponding address. If a device recognizes the Internet protocol addresses, it sends a message to the ARP claiming the Internet protocol addresses, it sends a message to the ARP claiming the Internet protocol addresses. ARP updates the contents of CAM and RAM for future reference and sends the packet to the destination machine. Figure 6 shows this Internet protocol address resolution process. This process can be implemented in software; however, implementing the search in hardware improves system performance.

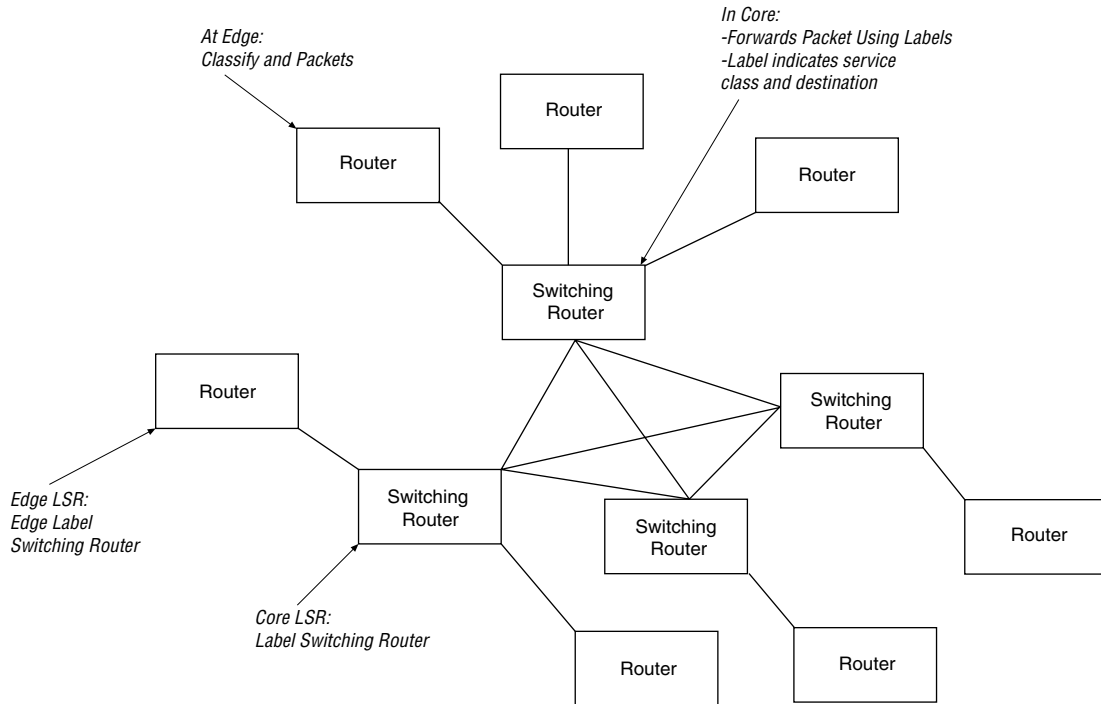
Figure 6. Internet Protocol Address Resolution



## Multi-Protocol Label Switching

Multi-protocol label switching provides a solution that combines the control of IP routing with the simplicity of layer 1 switching. Figure 3 shows an example of layer 1 switching. Multi-protocol label switching is a new routing/forwarding technique for the core of the Internet and supports the use of advanced routing services. This solves problematic issues such as the quality of services and delivery of new routing capabilities. Figures 7 and 8 show multi-protocol label switching by indicating that routers on the edge or in the core of the network have separate functions.

Figure 7. Multi-Protocol Label Switching Network



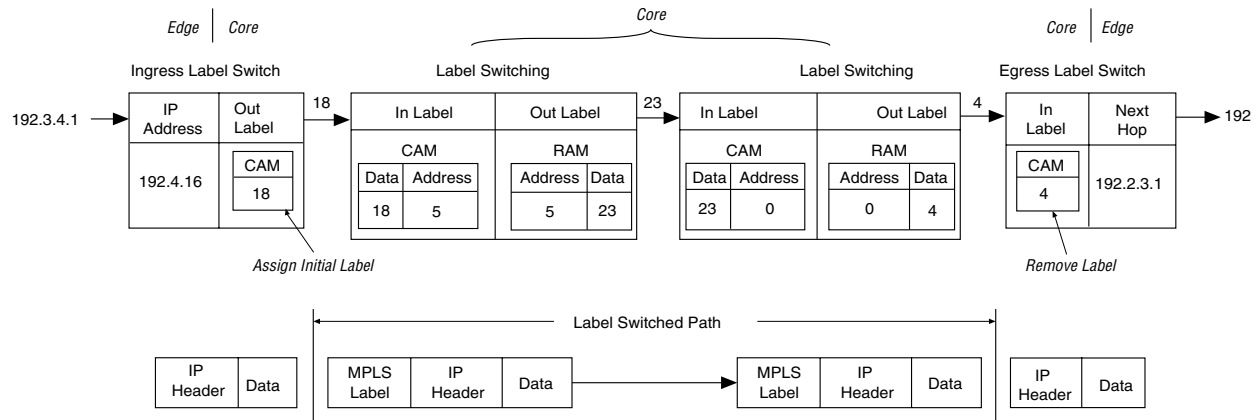
A multi-protocol label switching system is composed of two functional components: control and forwarding. The control component uses a standard routing protocol to exchange information and maintain a forwarding table with other routers along the network. When a router receives a packet, the forwarding component searches the forwarding table, which is maintained by the control component, to make routing decisions for each packet. The forwarding component is based on a label-swapping forwarding algorithm. Each control component is responsible for assigning and distributing a set of labels, as well as maintaining other relevant control information. Because multi-protocol label switching allows different modules to assign labels to packets, it decouples the forwarding of a packet from the contents of the packet's IP header.

At the ingress edges (entry) of the network, each incoming packet is classified and an initial label is assigned. The label switch performs a “longest-match routing” table lookup, assigns a label to the packet, then forwards the packet to the next hop on the label-switched path. In the core of the network, when a labeled packet reaches a switch, the forwarding component uses the input port number and label to perform an exact match search from its forwarding table. If a match is found, the forwarding component retrieves the outgoing label, the outgoing interface, and the next hop address from the forwarding table. The forwarding component then replaces the incoming label with the outgoing label, then directs the packet to the outbound interface for transmission to the next hop in the label-switched path. When the packet reaches the egress edge, the forwarding component searches its forwarding table. If the next hop is not a label switch, the egress switch discards the label and forwards the packet using conventional longest-match IP forwarding. [Figure 8](#) shows the basic multi-protocol label switching operation.

CAM can implement the required table and perform a fast search operation in the multi-protocol label switching operation. A CAM block finds the appropriate label for the incoming packet by searching the lookup table in the ingress label switch. The label is used as an index of a table, which specifies the next hop and the new label. The packet is forwarded to its next hop with the label attached. At the last edge of the network or egress label switch section, a CAM block can efficiently implement the table to remove the label from the incoming packet and forward the packet using IP forwarding.

A combination of CAM and RAM can be used to implement the table, which routes the incoming labeled packet to its next hop and finds the new label. Typical multi-protocol label switches store up to 1,024 labels at a time, requiring a  $1,024 \times 32$  CAM block. This CAM block requires only 32 embedded system blocks (ESBs) and can be efficiently implemented within APEX devices. Because the out-label is another label, RAM consisting of  $1,024 \times 32$  bits consuming 16 ESBs can assign the appropriate label to the incoming label. A total of 48 ESBs can perform the multi-protocol label switching operation (see Figure 8).

Figure 8. Multi-Protocol Label Switching Basic Operation



## Conclusion

Many networking applications require a fast search operation. APEX CAM can provide a solution for search operations in critical applications such as data compression, network switching, Internet protocol filtering, Internet protocol address resolution, and multi-protocol label switching. Using CAM simplifies and accelerates these functions and applications by increasing the performance of table look-up and translation implementation.