eSilicon®

High-Performance Network Data-Packet Classification Using Embedded Content-Addressable Memory

Embedding a TCAM block along with the rest of the system in a single device should overcome the disadvantages of discrete TCAMs. This paper provides an overview of advantages of embedded TCAMs and describes a few applications that could particularly benefit from embedded TCAM technology.

Abstract

With port speeds exceeding 100Gbps, route lookups that are fundamental to all routers have relied on ternary content addressable memories (TCAM) to provide a lookup response within a clock cycle. However, TCAMs in discrete form are expensive, consume a lot of power, compete for precious real estate on the printed circuit board (PCB), and in some applications lack required flexibility. Embedding a TCAM block along with the rest of the system in a single device should overcome these disadvantages. This paper provides an overview of advantages of embedded TCAMs and describes a few applications that could particularly benefit from embedded TCAM technology.

Executive Summary

Memory technology plays a significant role in implementing packet-processing functions in networking equipment such as switches and routers. Memories are used for packet storage, control tables, and lookup tables for making forwarding decisions. In the past, all these functions could be implemented with a single type of memory. Depending on the performance and buffering requirements of the system, designers had a choice between DDR4 SDRAM and synchronous SRAM technologies. For example, if the buffering requirements were large and performance requirements were not demanding, low-cost DDR3 was the preferred choice. On the other hand, if the buffering requirements were rather small but the performance requirements were high, then the high-performance synchronous SRAM was the preferred choice.

Fast-forwarding to today, the unprecedented rise in bandwidth demand on the Internet has raised the performance requirements of networking equipment. It is now common to find 1Tbps port speeds with High Bandwidth Memory. A significant portion of this bandwidth is driven by video applications that have no tolerance for latencies. This in turn has driven the need for processing packets at wire speed, wherein all the processing associated with a packet is expected to be performed within a packet time.

At 1Tbps, packet time for a 64-byte packet is 2.5ns! Within this time, all packet-processing functions, like parsing, classification and forwarding have to be completed.

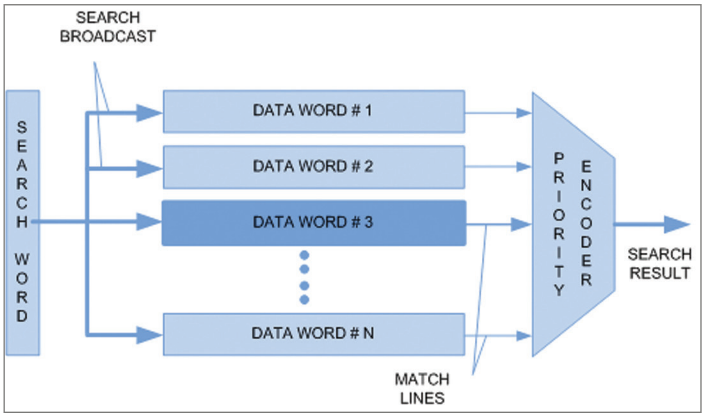
System designers now need to strike a balance between performance, power and cost of the systems. To achieve such balance, system designers have been forced to use different memory technologies to implement different functions.

For implementing high-speed lookup functions, system designers have been relying on specialty memory called ternary content addressable memory (TCAM). TCAMs provide a performance advantage over conventional RAM-based memory search algorithms. They allow simultaneous comparison of desired information against pre-stored entries. This inherent feature of a TCAM can deliver an order of magnitude reduction in search time over alternate methods of lookup table implementations.

Introduction to TCAM and Networking Applications

CAMs compare input search data against a table of stored data, and return the address of the matching data. CAMs have single-clock-cycle throughput making them faster than other hardware- andsoftware-based search systems. CAMs can be used in a wide variety of applications requiring high search speeds. The primary commercial application of CAMs is to classify and forward Internet protocol (IP) packets in network routers.

Figure 1: TCAM Block Diagram



In networking, information is encapsulated in data packets and transmitted through the network. These packets are routed from the source, through the intermediate nodes of the network (called routers) and received at the destination. The function of a router is to choose the appropriate route by comparing the destination address of a packet to all possible routes. A CAM is a good choice for implementing this lookup operation due to its fast search capability.

The figure above shows a simplified block diagram of a CAM. The input to the system is the search word that is broadcast onto the search lines to the table of stored data. Each stored word has a match-line that indicates whether the search word and stored word are identical (the match case) or are different (a mismatch case, or miss). The match-lines are fed to an encoder that generates a binary match location corresponding to the match-line that is in the match state. An encoder is used in systems where only a single match is expected.

In CAM applications where more than one word may match, a priority encoder is used instead of a simple encoder. A priority encoder selects the highest priority matching location to map to the match result, with words in lower address locations receiving higher priority. In addition, there is often a hit signal (not shown in the figure) that indicates the situation where there is no matching location in the CAM. The overall function of a CAM is to take a search word and return the matching memory location.

The conventional CAMs are binary, where each cell can take two logic states 0 and 1. Binary CAM performs exact match searches, whereas a ternary CAM (TCAM) allows pattern matching with the use of “do not care” (X). Each cell in TCAM can take three logic states: 0, 1, and X. X is either 1 or 0; and it behaves as a wildcard during a search. This means that TCAM can store a range of data as an entry. For example, the decimal range 0 to 255 can be represented in a TCAM by an entry 0XXXXXXXX, while a binary CAM would require 256 distinct entries to represent the same range of values.

Because of this feature, TCAMs are well suited for network operations, where the action performed on a packet can be identical for an entire range of destination addresses, such as packet classification and IP address lookup.

Internet protocol defines a mechanism to forward Internet packets. Each packet has an IP destination address. All routers have an IP address lookup table (forwarding table) that associates any IP destination address with an output port number or next-hop address. When a packet comes in, the router extracts the IP destination field and uses the IP destination address to look up the table to get the output port number for this packet.

Longest Prefix Match Lookup

IPv4 addresses are 32 bits long. An address can be represented in dotted-decimal notation: 32 bits are divided into four groups of 8 bits with each group represented as decimal and separated by a dot. For example, 128.114.20.15 could be a computer IP address of a corporation. An IP address is portioned into two parts: A constituent network prefix and a host number on that network. Classless inter-domain routing (CIDR) uses a notation to explicitly mention the bit length for the prefix. Its form is “IP address/prefix length.”

For example, 134.117.87.15/24 means that 134.117.87 is for the network and 15 is for the host. 134.117.87.15/22 means that 134.117.84 is for the network and 3.15 is for the host. For the latter case, some calculations are needed. This address has 22 bits as prefix and 10 bits as host. For our discussion it is important to realize that the longer the prefix, the smaller the subset. The fundamental goal for IP address lookup process is to identify the most specific prefix that matches an IP address. It is also called the longest prefix match, because the longest the prefix, the more specific it is.

The prefix in the routing table is a bit string specifying the initial substring of a network address and trailing bits as wildcards. For example, a prefix 100110XXXX represents the IP address ranges from 1001100000 to 101101111. Thus TCAMs are very well suited for storing prefixes of different lengths. In a TCAM, each bit of the incoming data is compared with the same position bit of the stored data, and the result is the address of the memory location where the match passed. In some cases, a search will result in multiple matches.

To perform the longest prefix matching operation, all prefixes are stored in a TCAM in decreasing order of lengths. The TCAM searches the destination IP address of an incoming packet with all the prefixes in parallel. Several prefixes may match the destination IP address. Priority encoder logic then selects the first matching entry, the entry with the matching prefix at the lowest physical memory address, which is the longest matching prefix. The physical memory address is used to extract the corresponding forwarding information, such as next hop, from the SRAM module. Because a TCAM can directly store prefixes and find the longest matching prefix in a single cycle, it has become an attractive solution.

Challenges Associated with Discrete TCAM

Cost, Power and Real Estate

Discrete CAMs that are available in the market are traditionally associated with high costs, consuming significant amount of power, and occupying a large PCB footprint. It is typical for a TCAM device to cost much more than other system components like network processors, CPUs, FPGAs, or traffic managers.

Most of the telecom platforms that are designed for central offices provide no more than 150W-200W of power budget per slot. System designers are often challenged to meet power budgets of line cards when TCAMs are used in the system. A typical line card has a CPU, network processors, traffic managers, switching fabric interfaces, and a significant number of memory components. But a typical TCAM already consumes around 15-20W, which make it difficult for a line card design to meet power budget with all these components.

The power consumption of discrete TCAMs forces the need to use large packages for heat dissipation. The large package occupies a significant amount of PCB real estate. This makes it a challenge for system designers to accommodate them in real estate-constrained line cards.

Maintenance and Table Management

The "longest prefix match" application that was discussed earlier requires table entries to be stored in order for the priority encoder to determine the correct result. This particular requirement would not be an issue if network routes remained static. However, the network infrastructure is a dynamic environment and the routes change constantly. This forces the content of the TCAMs to be sorted and arranged, such that the entries are in order of decreasing prefix length with no empty space in between entries. This requires system designers to allocate significant processor cycles—from the already limited performance/power budget—for table management and maintenance.

Inflexibility with Table Configurations

As we will see later, every application has its own unique needs with respect to configurations. Also it is conceivable for a single application to use CAMs of two to six different configurations. Most discrete CAMs suffer from a lack of flexibility when it comes to the line width because they are built for the most popular configuration. This lack of flexibility results in inefficiencies and the need for additional devices if the configurations offered do not meet the system designer's requirements.

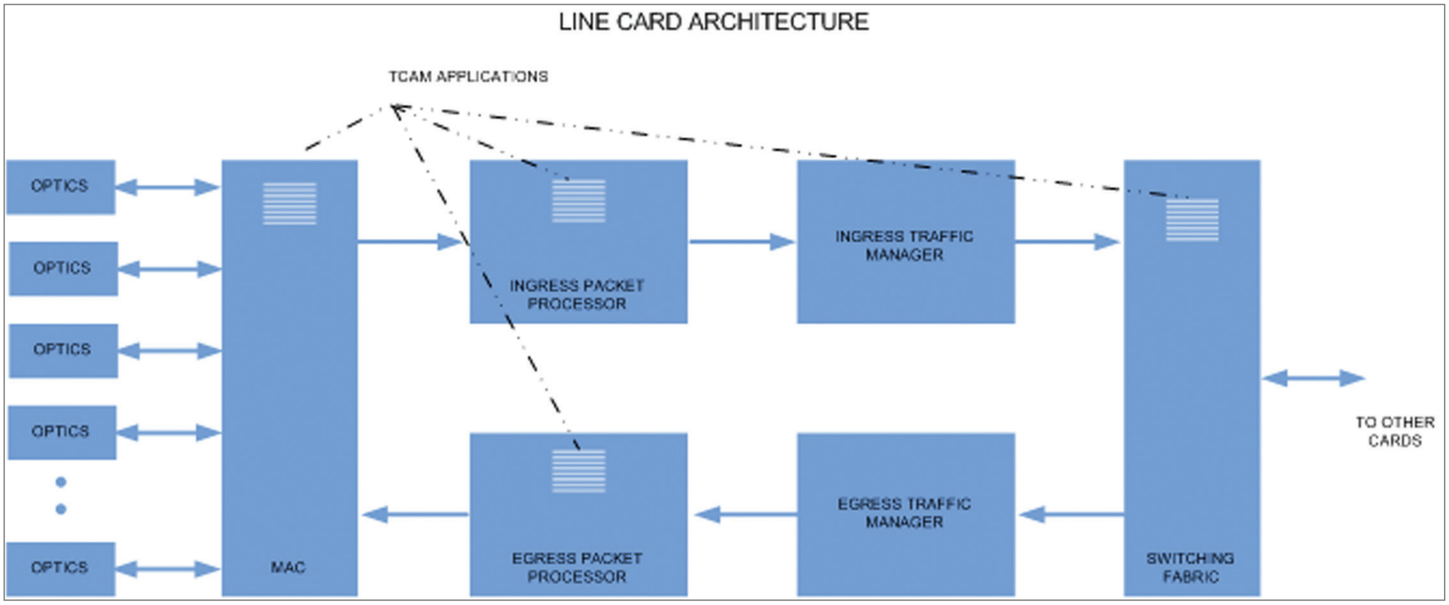
Complex I/O Requirements

Finally one of the key challenges with external CAMs has been the large I/O requirements. Given limited timing budgets for processing packets, system designers are looking to solutions that get them a lookup result in a single clock cycle. This means they would like to forward the entire line to the CAM instead of splitting them across multiple cycles. In addition to this, it is always conceivable for the CAM interface to either equal or exceed port speed. This has forced some of the newer generation TCAMs to use high-speed SerDes as the default choice for I/O. This approach, although very attractive for solving I/O speed and width problem, compounds the cost, power, latencies and interoperability issues.

Embedded TCAM Advantages and Applications

The limitations of discrete TCAMs identified in the previous section provide the case for embedding the TCAM in system-on-a-chip (SoC) designs. This section discusses various networking applications that can take advantage of embedded TCAMs. We

Figure 2. Line Card Architecture



will start with a brief description of the architecture of a typical networking line card, followed by a detailed explanation of applying embedded TCAMs in networking line card design.

The Figure 2 on the previous page provides an architectural view of a typical networking line card and highlights the areas where embedded TCAMs could be used. In the ingress direction, the MAC aggregator receives the packet and forwards it to the ingress packet processor. The ingress packet processor then parses the packet, strips the packet headers and performs classification to determine the destination of the packet. The ingress packet processor also adds quality of service (QoS) parameters to the forwarded packet. The ingress traffic manager queues the packets according to classification and QoS parameters, and schedules the transfer of packets to the switching fabric.

Similarly in the egress direction, the packets along with the switching header arrive at the egress traffic manager. The egress traffic manager strips the packet of the switching header and forwards the rest to the egress packet processor according to QoS parameters. The egress packet processor then performs some basic packet processing functions, including classification, and forwards the modified packets to the MAC aggregator. The MAC aggregator performs additional classification on the packet to determine the output port, modifies the packet and forwards it to the appropriate port.

As one can see from the above applications, TCAMs are widely used across multiple functional devices within a line card. In addition to performance requirements, each of these functions will require very different configurations depending on the nature of lookup, content of lookup, and the position of the platform within the network. Platforms positioned in the access locations inherently require a lower number of entries, while platforms positioned in the edge and core of the networks will require large number of entries. Also, in certain instances, tables will be consolidated in one location, and eliminate the requirements for lookups in other locations in a line card.

The above applications require high degree of flexibility in configuring discrete TCAMs. However, the discrete TCAM market is not large enough for suppliers to offer a huge portfolio of devices with different configurations. The following table gives an idea of various configurations demanded by some of the popular applications.

As discussed above, TCAM configurations will be changed depending on the architecture and positioning of the platform in the network. If the network element is expected to support fine-grained QoS and a large number of applications and users, then deep tables will be needed. In this case, large TCAMs will be required. On the other hand, if the network element is less sophisticated and is expected to support best-effort services, very small tables resulting in shallow TCAMs will be used. The data width of a table varies with applications. It depends on the performance requirements and timing budgets of the designs.

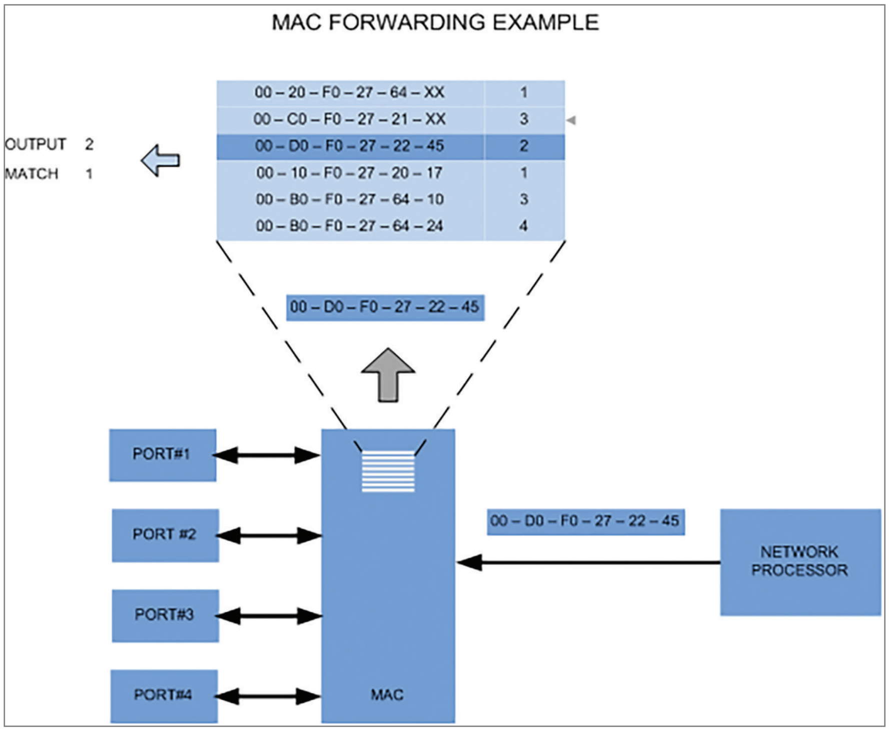
Table 1. CAM Configurations

	MAC	Switching	Packet Processing
Port Bandwidth	100Gbps	20x20 10Gbps	100Gbps
CAM Type	BCAM	BCAM	TCAM
Application	L2 Packet Classification	Port Identification	L2- L4 Packet Classification
Lookups/Packet	1	1	>4
Data Width	72 - 128	32 -96	128 - 256
Depth	512 - 1024	128 - 512	512 - 4096
Access/Sec	250Msps	512Msps	>512Msps

Embedded TCAM in the Media Access Control (MAC) Aggregator

The media access control (MAC) device performs layer 2 functionality in networking equipment. Layer 2 processing typically involves two important functions: learning and forwarding.

Figure 3. MAC Forwarding


MAC FORWARDING EXAMPLE
00 – 20 – F0 – 27 – 64 – XX 1
00 – C0 – F0 – 27 – 21 – XX 3
00 – D0 – F0 – 27 – 22 – 45 2
00 – 10 – F0 – 27 – 20 – 17 1
00 – B0 – F0 – 27 – 64 – 10 3
00 – B0 – F0 – 27 – 64 – 24 4
OUTPUT 2
MATCH 1
00 – D0 – F0 – 27 – 22 – 45
PORT#1
PORT #2
PORT#3
PORT#4
MAC
00 – D0 – F0 – 27 – 22 – 45
NETWORK PROCESSOR

Learning is the process of obtaining the MAC address of connected devices. When a frame reaches the port of a switch, the switch extracts the MAC address of the source device from the frame and compares it to the MAC address table present in the TCAM. If the MAC device cannot find a corresponding entry in the MAC address table, it will add the address to the table with the port number that it extracts from the received frame.

If the MAC address is already available in the MAC address table, the MAC device compares the incoming port with the port already available in the MAC table. If the port numbers are different, the switch updates the MAC address table with the new port number. This will normally happen when network administrators remove the cable from one port and attach it to another port.

Forwarding is the process of passing network traffic from the input to one of the output ports of the aggregator. In the egress direction in our example, all packets arrive from the network processor through a single interface, and then the packets are forwarded to a specific port. The MAC device receives the frame, extracts the source MAC address and destination MAC address, as part of the learning function and forwarding function, respectively. If the destination MAC address is found in the MAC address table, the device forwards the frame via the corresponding port of the MAC address.

If the destination MAC address is not found in the MAC address table, the switch forwards the frame through all its ports except the source port. This is known as flooding. Normally flooding happens when the network switch is starting up. Flooding prevents loss of traffic when the switch is learning. When the destination device receives the frame and sends a reply frame to the source device, the MAC device reads the MAC address of the destination device and adds it to the MAC address table, which is the function of the learning process.

Depending on the implementation of the MAC table, the entry size in the TCAM could be of different widths. One can always envision adding more features like ageing and VLAN IDs to the MAC table. It is also possible for a system designer to come up with a compression scheme to eliminate a few columns of TCAM cells. This approach would cost additional clock cycles and logic to compress the entries. If a system designer cannot afford this clock-cycle overhead, he must widen the CAM to whatever width is required. This is also true for the depth of entries. A MAC aggregator takes different flavors depending on the platform. A MAC aggregator present in a switching or aggregation application at the access network will certainly require a much lower number of entries as compared to applications at the edge or core of the network.

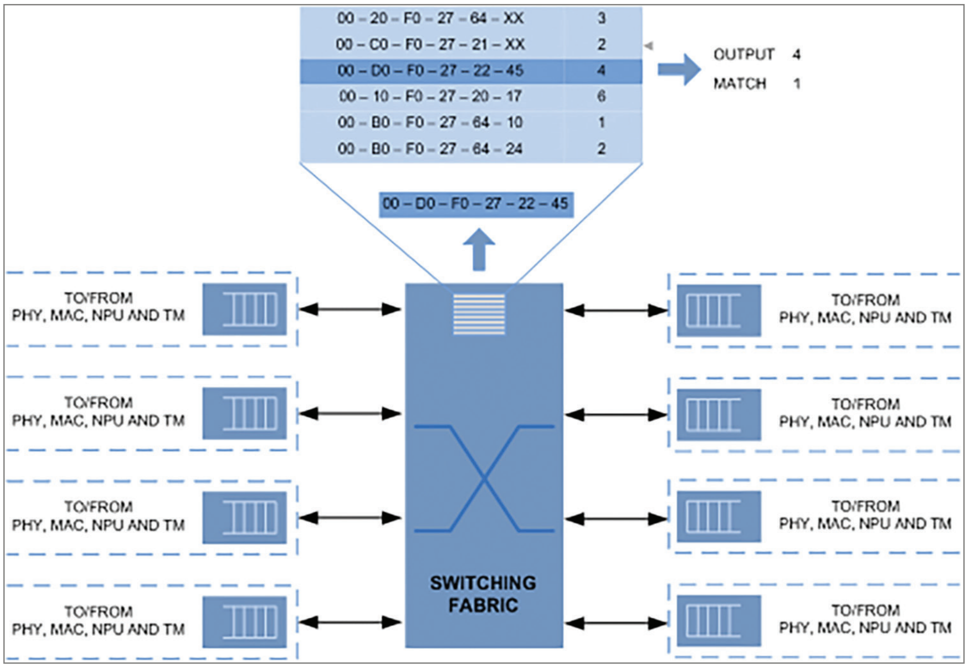
Through customization, embedded TCAMs offer the optimal configuration to fit the architecture. System architects can now focus on defining architecture that will differentiate their products and then pick the best configuration of the TCAM, rather than the other way around. Additionally, expensive TCAM cells are fully utilized in an embedded TCAM because of customization; hence, further system cost reduction can be realized.

Embedded TCAM in Switching Fabric

Switching fabric is responsible for directing received frames from each input port to the appropriate output port. It must also be able to handle a broadcast to all output ports. In general, there are two ways to build switching fabric: crossbar switch or dynamic switching. The crossbar switch needs to be statically configured with pre-defined input to output connections. There is no requirement for lookups in crossbar switches.

As the name implies, in the case of a dynamic switch, the switching decision is made by interpreting the switching control word in the received frame. To avoid head-of-line blocking issues, packets are fragmented into fixed-size frames and switching headers are appended to it even before it arrives at the switching fabric. This process is typically performed either by the traffic manager or the switching interface device. The switching control word provides critical information that allows the switching fabric to make correct forwarding decisions.

Figure 4. Switching Fabric



00 – 20 – F0 – 27 – 64 – XX	3
00 – C0 – F0 – 27 – 21 – XX	2
00 – D0 – F0 – 27 – 22 – 45	4
00 – 10 – F0 – 27 – 20 – 17	6
00 – B0 – F0 – 27 – 64 – 10	1
00 – B0 – F0 – 27 – 64 – 24	2

OUTPUT 4
MATCH 1
00 – D0 – F0 – 27 – 22 – 45
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM
SWITCHING FABRIC
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM
TO/FROM PHY, MAC, NPU AND TM

The switching fabric receives the frames and stores the frames in a small buffer. Buffering lowers the rate of collisions and allows the switching fabric some room for short periods of time without losing data. The core switching element within the switching fabric needs to process the frames in a multiplexed fashion at much higher rates than the individual port rates in the dynamic switching fabric. The multiplexed bus effectively makes one input-output connection at a time, with each input port processed one at a time. When many ports are multiplexed in this fashion, the data rate at the core switching element must be much faster than the individual port. For example, on a 10-port switch, with each port running at 10Gbps, the core switching element would need to operate at 100Gbps.

The core switching element receives the frame from the multiplexed bus and extracts the switching-control header. A portion of the switching-control header is dedicated to the destination address. The core switching element launches a lookup to an embedded CAM to determine the destination port.

In the figure above, “00-D0-F0-27-22-45” is the switching-control header received from one of the input ports. This information is extracted by the switching element and a lookup is launched and destination port 4 is obtained as a result from the CAM. The entire frame is then forwarded to output port 4.

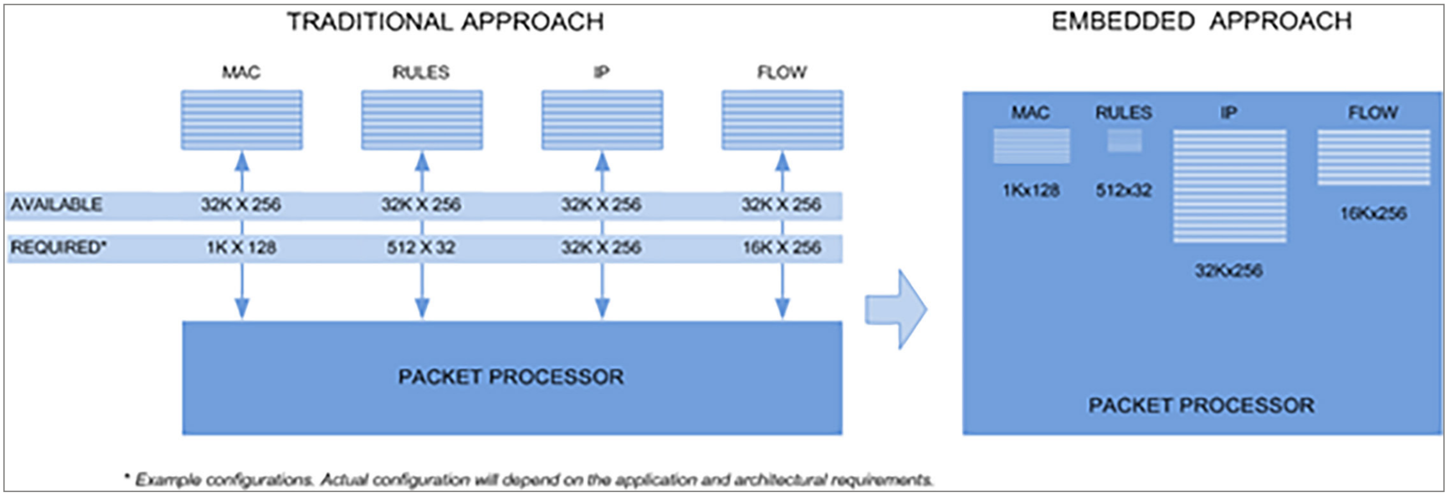
As discussed in the beginning of this paper, 100Gbps line cards are in the mainstream today, which means that each line card needs to support a bandwidth of at least 100Gbps for wire-speed operation. This, in turn, is pushing the limits of the switching-fabric technology, and in fact, terabit switching fabrics have already been productized.

From a lookup perspective, switching fabrics require high performance and reasonably shallow CAMs. In addition, because of limited I/O availability and bandwidth, discrete TCAMs are not an option for this particular application. Embedded TCAMs are well suited for switching-fabric design.

Embedded TCAM in Deep-Packet Classification

Traditional classification systems could afford to perform searches one by one in a pipelined fashion. With high-bandwidth requirements in many high-speed network processing systems, several searches must occur simultaneously in order for the

Figure 5. Packet Processors



	MAC	RULES	IP	FLOW
AVAILABLE	32K X 256	32K X 256	32K X 256	32K X 256
REQUIRED*	1K X 128	512 X 32	32K X 256	16K X 256

* Example configurations. Actual configuration will depend on the application and architectural requirements.

equipment to guarantee deep-packet inspection and processing at wire speeds.

Carriers want to support differentiated services with fine-grained QoS based on applications. This means that the search engine that supports the classifier must be able to produce results within extremely short amounts of time. In such applications, several tables need to be accessed at the same time.

For example, a MAC table, an IP table, a rules table, and a flow-management table must all be consulted in parallel as shown in the figure above. A traditional approach would need multiple discrete TCAMs of various widths and depths. This brute-force approach would need four separate TCAMs with four search interfaces, each holding one of the tables. Also, if one of the tables isn't large enough to fill up the complete CAM—like a layer 2 MAC table, which may require only 1024 entries depending on the platform—then the rest of TCAM cells are wasted. The unused space cannot be used for another table, no matter how small, since all the tables must be searched in parallel.

Now, as shown in the figure above, by embedding the four TCAMs in the SoC, the system designers overcome all of the limitations identified above. TCAM cells will not be wasted, because the tables would be custom built according to specified requirements. The I/O requirement for the SoC would drop dramatically because there is no requirement to access pin-intensive discrete TCAMs. This approach will also free up significant real estate for other strategic components on the PCB and provides significant power savings.

Embedded TCAM in Active Control List (ACL)

ACL is a filter that controls network traffic in and out of a network. Based on these filters, packets are either permitted or denied access to specific ports or specific types of services. Normally these filters are manually configured by network administrators to filter traffic and to provide security for the networks. TCAMs are used for ACL management. An ACL engine looks up the rule tables (masks) maintained and stored in the ACL TCAMs to determine how a packet will be handled. ACL entry typically consists of five fields: source and destination IP addresses, port addresses, protocol type, and access action (permit/deny). Each entry is programmed as masks in TCAM tables. Packets are filtered by TCAMs based on the masks programmed. The packet might be permitted or denied access to the network/services based on the result returned by the TCAMs.

ACL rules for a typical network gateway can consist of just a few to tens of thousands of entries. For example, an ACL rule table of 2K depth and a 288 width, which can handle both IPv4 and IPv6 standards, can be implemented by cascading eight off-the-shelf embedded TCAMs 512x144 in size.

Embedded TCAM power-management features are very useful in ACL implementations. The "row valid" feature is widely used by a TCAM-based ACL if there are limited numbers of rules, i.e., there are empty row entries in a TCAM. To save power, one can dynamically program a TCAM so that a block of rows (typically 16) can be excluded from the compare operation.

Customer Success Stories

eSilicon has supplied embedded TCAMs to networking and communications customers for a number of years. We have silicon-proven CAMs (BCAM & TCAM) from 180nm to 14/16nm. The 14nm and 16nm embedded TCAM macros are available now for integration.

The table below shows a small sample of our customers who have successfully used eSilicon embedded TCAMs in their applications. Each TCAM macro is optimized per the customer's requirements (custom features, various table width or depth, etc.).

Table 2. Customer Success Stories

Customer	Process	Applications	CAM Functions	Customized Features	CAM Size in Chip
A	14LPP	Enterprise/ Data Center Networking Server	ACL, IP, MAC Lookup	Priority Encoding, ECC	120Mb
B	16FF+LL	Router	ACL, MAC Lookup	Power Management	80Mb
C	16FF+GL	Enterprise/ Data Center Networking Server	ACL, IP, MAC Lookup	2-Port	40Mb
D	28HPM	Ethernet Switch, Servers	ACL	Pseudo-2-Port, Power Management	20Mb
E	28HPM	Network Processor	ACL	Priority Encoding	44Mb
F	28HPC+	Ethernet Switch	ACL	Optimized registered array architecture	40Mb
G	40G	InfiniBand Switch	ACL	Power Management	1.8Mb
H	40G	Enterprise/ Data Center Networking Server	ACL, IP, MAC Lookup		15Mb

Table Note: eSilicon developed the foundry-approved TCAM bitcells for 40nm and 28nm technologies

Conclusion

Embedded TCAMs allow system designers to define various configurations using multiple TCAMs to match architectural requirements to achieve optimal performance, power and cost of the system. Discrete TCAMs, on the other hand, force designers to compromise the system design to accommodate the discrete devices.

In addition to the inherent power savings achieved through embedded TCAMs, further savings in power could be attained through creative system design. For example, designers can intelligently partition TCAM blocks, such that unused blocks could be dynamically shut off to achieve significant power savings.

Last but not least, embedded TCAMs enable system OEMs to focus on system innovations through creative system architectures and differentiation, and also allow them to have control of their network application intellectual property (proprietary technology). Embedded TCAMs also provide an opportunity to lower overall system costs and power consumption. Discrete TCAMs are restrictive in all the above fronts.

About eSilicon

eSilicon guides customers through a fast, accurate, transparent, low-risk ASIC journey, from concept to volume production. Explore your options online with eSilicon STAR tools, engage with eSilicon experts, and take advantage of eSilicon semiconductor design, custom IP and IC manufacturing solutions through a flexible engagement model. eSilicon serves a wide variety of markets including the communications, computer, consumer, industrial products and medical segments. Get the data, decision-making power and technology you need for first-time-right results. The right chip. Right now™

Contact

Please contact eSilicon at ipbu@esilicon.com for more information, silicon quality results, white papers or complete data sheets. www.esilicon.com

eSilicon Headquarters
2130 Gold Street, Suite 100
San Jose, CA 95002
www.esilicon.com
sales@esilicon.com
ipbu@esilicon.com

Phone: 1.408.635.6300
U.S. & Canada Toll Free:
1.877.7.MY-CHIP
(1.877.769.2447)