Content Addressable Memory (CAM) Applications for ispXPLD™ Devices

July 2002 Application Note AN8071

Introduction

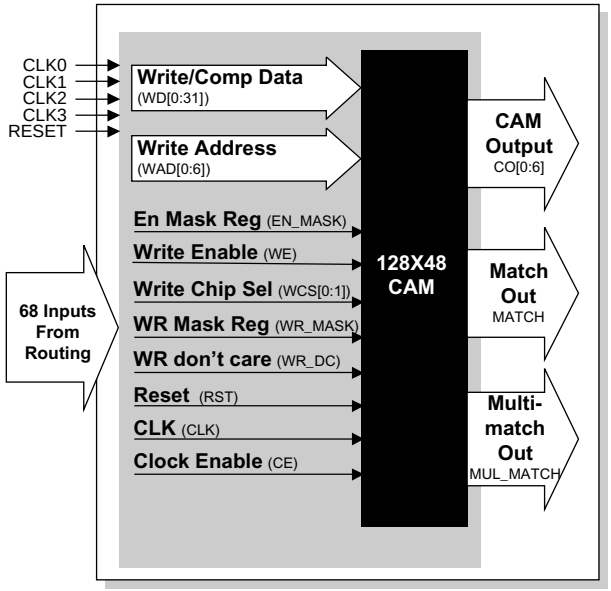
Content Addressable Memory (CAM) is a type of memory that compares the input data with the preloaded contents of the CAM block and generates a given output depending on the kind of CAM. This kind of memory provides a distinct speed advantage over RAM in systems requiring quick address comparison or retrieval. Typical RAM applications utilize a counter to load the addresses into the RAM at a rate of one address per clock cycle. The data from the RAM block would then be compared with the expected data using XOR logic. When a match is found, the address becomes the valid output. The number of clock cycles required for this whole process is the number of addresses. For most applications, this would take more than one clock cycle. Because CAM can take the input (data) and compare it with all of the preloaded contents in the CAM array simultaneously, it can execute the entire lookup in a single clock cycle. This speed is ideally suited for network applications such as address lookups and filtering, packet encryption, and firewalls.

The ispXPLD 5000MX supports designers who want to implement CAM within their system. This application note describes the CAM itself and three applications: data compression and encryption, IP addressing, and Multi-Protocol Label Switching (MPLS).

ispXPLD 5000MX CAM Description

In the ispXPLD 5000MX device family a Multi-Function Block (MFB) can be transformed into a CAM memory with 128 entries of 48 bits in width. Each bit can have one of three values: 0,1, or X (don't care). This ternary CAM has 64 inputs and nine outputs. Figure 1 depicts a simplified block diagram of the ispXPLD 5000MX CAM.

Figure 1. ispXPLD 5000MX CAM Block Diagram



The inputs into the CAM memory are for data and control. Other input signals include the enable mask register, write address, write enable, write chip selects, write mask register, write don't care, reset, clock, and clock enable. The outputs consist of seven address and two match flag signals.

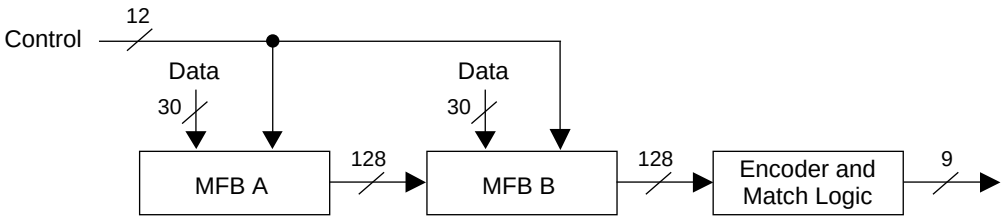
The CAM performs two types of functions: compare (read) and update (write). Upon power-up, the device loads the contents of the CAM with user-specified initialization data. After power-up, the user can load additional data into the CAM using the data address and write enable signals. A mask register allows writes to partial words. Asserting

the enable mask register signal enables the mask register to be written to using the data and write mask register signals. Don't cares can be written to a location by asserting the write don't care signal.

The input data to be compared is first loaded into the registers of the CAM. Next, the data is encoded into a word in the compare and write data encoders before approaching the CAM block. If a single match is found, the 128-bit CAM output word is encoded into an address, which is one of the outputs of the CAM. In addition, the two match flags MATCH and MUL_MATCH receive values of 1 and 0, respectively. If more than one match is located, priority is given to the lower address (0 is the lowest). In this case, MATCH and MULT_MATCH both equal 1.

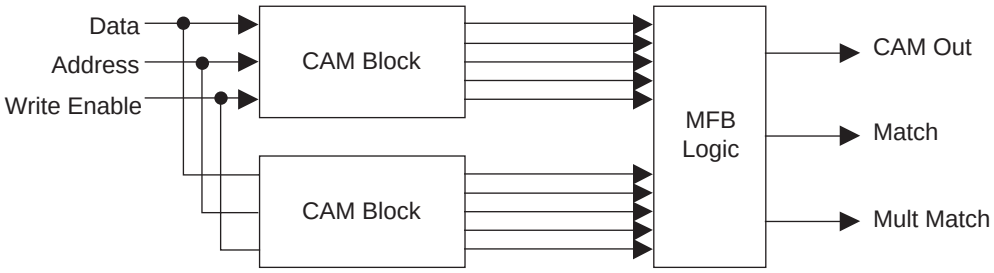
The ispXPLD 5000MX CAM can be cascaded to expand its width or depth for larger applications. Built-in logic allows up to four CAM arrays to be in cascade for wider CAM word sizes. For a width that is not a multiple of 48 bits, a larger CAM can be comprised of multiple MFBs with the number of bits divided throughout the whole memory. For example, a 60-bit CAM can consists of two adjacent MFBs cascaded together with 30 bits each as shown in Figure 2.

Figure 2. 60-Bit CAM Using Two Adjacent MFBs



To increase the depth of the CAM, the inputs can be configured for simple writing to four CAM blocks. The compare word would be loaded into the compare/update data register for all the CAMs. Combination of CAM output is implemented in additional MFB logic. The individual addresses are compared, and an overall address and the match flags are generated for the entire CAM. Figure 3 illustrates the concept of depth cascading.

Figure 3. Diagram of Depth Cascading



The main aspect of the ispXPLD 5000MX CAM is its speed for lookup tables and algorithms for designers. Data compression and encryption is one such CAM application.

Data Compression and Encryption

Data compression is accomplished by finding the redundant patterns in the data and replacing them with shorter symbols called tokens to reduce the amount of original data. This increases the throughput of the data being sent over a fixed bandwidth. Encryption takes the expected patterns in the data and converts them to a hidden meaning based on an encryption algorithm. Compression normally precedes encryption. If encryption occurs before data compression, the outcome would most likely be a larger amount of data. This would effectively decrease the possible throughput of data across the network. Compression and encryption are handled to save money when transmitting information and for increasing privacy.

Lossless compression is one category of compression algorithms. Decompression of files, such as code, text, and numeric data, using a lossless compression algorithm results in the same exact file as the original. One kind of

lossless compression algorithm is dictionary-based. The purpose of a dictionary-based compression algorithm results in an efficient usage of both processor cycles and memory space. LZ78 is a dictionary-based compression technique developed by Lempel and Ziv in 1978. This algorithm creates a dictionary of common strings found in the data and replaces them with a corresponding index to the dictionary.

Once the data is compacted, encryption is the next step. Encryption takes the data and applies mathematical functions to change the appearance of the text beyond recognition. Substitution and transposition are used in modern encryption. The sender encrypts the plaintext message using an encryption algorithm before transmitting the ciphertext message over the network to the receiver who decrypts the message to get the original plaintext sent. The sender and receiver have at least one key depending on the type of cryptographic algorithm to ensure the identity of the participants and/or to encrypt/decrypt the message. One type of cryptographic algorithm is the secret key algorithm or symmetric key algorithm. Both the send and receiver share the identical key for this algorithm.

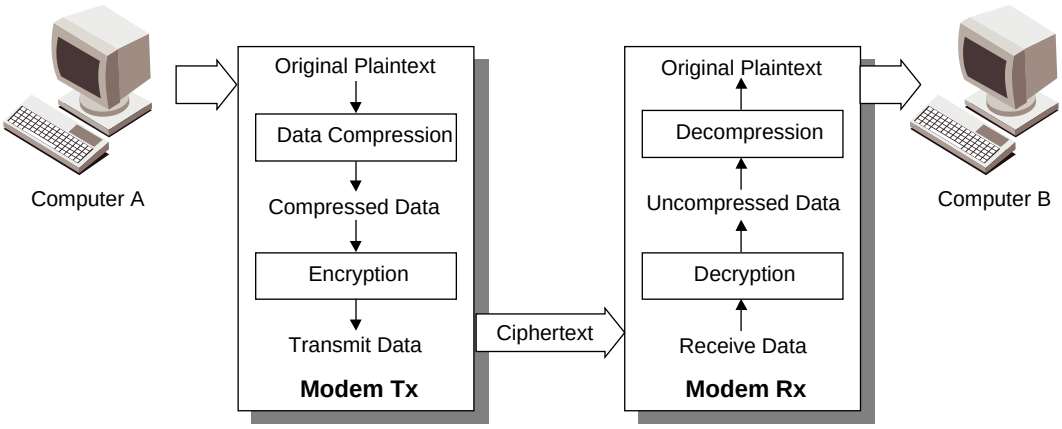
Data Encryption Standard (DES) is a symmetric key algorithm. It was originally developed by IBM Corporation and was published by an U.S. federal agency that develops standards called NIST in 1977. This algorithm divides the plaintext into 64-bit blocks and enciphers them one block at a time utilizing the key. This is called block cipher. The key implemented is 64 bits long; however, only 56 bits are utilized. The eighth bit of each byte is the parity bit for that byte. DES involves permutation, substitution, and bit-wise XOR operations. The algorithm has three main stages: the 64 bits in the block are shuffled, 16 rounds of a function jumble the key and data, and the inverse of the original permutation occurs. The Electronic Codebook is one of the DES operation modes. To ensure privacy and authentication, DES can be applied three times using three or two unlike keys. This is called Triple DES (TDES).

Example

Below is an example of implementing data compression and encryption using the ispXPLD 5000MX CAM.

Computer A wants to send data over the network to Computer B. Computer A first compresses the data (original plaintext) using the LZ78 algorithm to reduce the data in a serial bit stream. Subsequently, the compressed text is encrypted using DES and the exact secret key as Computer B to avoid any third-party user from altering or reading the contents of the message. Once the total output (ciphertext) has been transmitted from Modem Tx to Modem Rx, Computer B decrypts the incoming message based on the reverse of the same encryption algorithm to produce the compressed text. Then it decompresses the output to get the same data that Computer A started with (original plaintext). Figure 4 demonstrates the process.

Figure 4. Block Diagram of Data Compression and Encryption



For data compression, the repetitive patterns are loaded into the ispXPLD 5000MX CAM. Next, the input bit stream is compared with the contents of the dictionary loaded into the CAM. If a match is located, a token (lowest address of match in CAM) replaces the pattern and shortens the overall output data. If a match is not found, the next word is searched. Compression continues until the end of the input data. Below is an example of data used. The words in the following section are represented in symbol form as opposed to ASCII form.

Input Data ABDGABCDBDEGABAABE...
Output Data ABD42B34AABE...

Dictionary 1) ABC
2) CD
3) DE
4) GAB
•
•
•

The output is stored in the ispXPLD 5000MX. This data compression algorithm condensed the input data above by 40%. Consequently, less information will be sent over the network, resulting in a greater throughput of data across the network.

Once the compressed text has been generated, the DES algorithm is applied to begin the encryption process. The user writes the logic equations in the software to shift and permutate the blocks of 64-bit data and generate the 16 subkeys for one round of DES encryption.

To obtain the encrypted data, the CAM is used in conjunction with the RAM. Two ispXPLD 5000MX CAMs are needed for this encryption algorithm because a 64-bit block is permutated at a time. These CAMs store the compressed data and keys without permutation being applied. The CAM gives the address of the data. This address is utilized to locate the encrypted data in the RAM. Table 1 illustrates the use of CAM and RAM to get encrypted information.

Table 1. Example of CAM and RAM Encryption

CAM

Data	Address
ABD	0
EFG	1
ABE	2

RAM

Address	Data
0	DFG
1	IOP
2	RET

The compressed data is ABD42B34AABE. The encrypted data is DFG42B34ARET. This new data is not recognizable. To have TDES, DES is applied three times using the same key for the first and third DES and a different key for the second DES.

IP Addressing

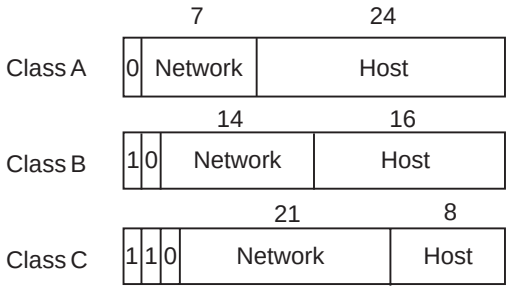
When transmitting information from one device to another across a network, IP addresses are utilized to identify each device. Each network can have any of the following items: a router, a physical segment, a gateway, and/or a subnet. A physical segment connects the routers to the network. A gateway, or router, interconnects the networks. Any device, such as a router, printer, or computer, has a unique IP address per network interface card.

This section of the application note covers IP version 4 (IPv4), which is the present version of Internet Protocol. Its addresses are 32 bits long. For convention, the binary number is converted to decimal notation and a dot separates each byte (e.g. 193.2.41.53). IP addresses are classified under five distinct classes ranging from Class A to Class E. The primary classes are Class A, Class B, and Class C. The most significant bits of each IP address determine

its class. For instance, Class A addresses have 1 as the first MSB. Class B addresses have 10 as the first two MSBs. Class C addresses have 110 as the first three MSBs.

IP addresses are hierarchical addresses that contain network and host subaddresses. The network part of the address signifies which network is connected to the host. Each host has its own unique network address. Because the network is part of a larger internet, the network can access outside its own physical segment. The host part of the address represents each host on the network uniquely. The following description for each class is for IPv4 addresses. Class A addresses have the next seven bits for the network ID and 24 bits for the host ID. Class B addresses have the next 14 bits for the network ID and 16 bits for the host ID. Class C addresses have the next 21 bits for the network ID and eight bits for the host ID. Class A supports the most number of networks. The number of networks supported decreases when moving up the IP address classes. Figure 5 explains the IP addresses in detail.

Figure 5. Classes for IP Addresses



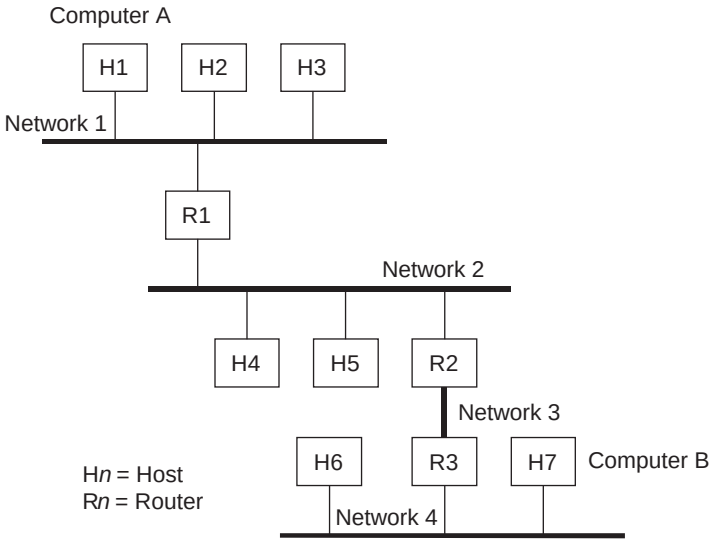
Packets of data are delivered with the IP address of the desired destination. The host or router of the source first compares the network ID of the destination address against the network ID of other interfaces on that physical network. Hosts have only one address. On the contrary, routers and gateways have one or more addresses since they are connected to at least one network. If a match is located, the host and routers on that physical network send frames over that network. Otherwise, the packet is sent to the router of that physical network that chooses the next best destination for the packet using a forwarding table, which can be implemented in a CAM. This forwarding process continues until the packet reaches its destination. Once the packet arrives at the desired host, a corresponding table converts the IP address to its Media Access Control (MAC) address, which is the computer's unique hardware number on a local area network or the Ethernet address on an Ethernet. The packet has fully gone through its delivery and reached the destination after conversion.

Example

Below is an example of a data packet, which includes an IP address, being delivered from the source to the destination host.

Computer A (H1 on Network 1) wants to send information to Computer B (H7 on Network 4). H1 sends the packet of data with the IP address of H7, which is 193.25.9.53. This is a Class C IP address. Figure 6 shows the internetwork for this situation.

Figure 6. A Simple Internetwork



Each node in the internetwork has a forwarding table of the direct interfaces. For instance, refer to Table 2 for the routing table for R2. This shows the next destination based on which network the packet should travel to. This is implemented using logic equations.

Table 2. Routing Table for Router R2

NetworkNum	NextHop
1	R1
2	Interface 0[1]
3	R3
4	Interface 1[2]

1. Interface for Network 2.
2. Interface for Network 4.

The user loads the IP address of the hosts and routers of each network into a separate CAM. The addresses of all the networks cannot be loaded into the same CAM because the CAM can produce a match for an address not on the same network. Therefore, the next hop can be wrong for that particular condition.

H1 first compares the network ID of 193.25.9 with that of H2 and H3 on the same physical network by comparing the address with the CAM's contents. No match is found because it is not on the network. As a result, H1 sends the packet to Router R1 because it is the only available path to another network, Network 2, based on the routing table being implemented in logic equations. The address of the destination host is compared with the address of each network in the CAM until the packet reaches H7. R1 cannot deliver directly to H7 because none of the interfaces on R1 is on the same network as H7. Then the packet is delivered to Router R2, which selects the next best destination according to the forwarding table for R2. The packet is forwarded to R3 through Network 3. A match is located, meaning that the host is on Network 4. The packet is forwarded to H7. The address of the location in the CAM is output, and MATCH and MUT_MATCH have a value of 1 and 0, respectively. The address from the CAM is utilized in the RAM, which contains the matching addresses' MAC address. Table 3 lists the contents of a CAM and RAM for an IP addressing application.

Table 3. Example of CAM and RAM for IP Addressing

CAM

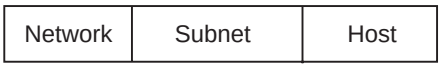
IP Address	Address
193.25.9.25	0
193.25.9.42	1
193.25.9.34	2

RAM

Address	MAC Address
0	00-01-03-28-97-A6
1	00-05-07-36-83-A8
2	00-05-03-23-82-A4

When the number of logic networks associated with an IP address is not enough to support the network, a subnet is used to expand the number of logic networks within a class of networks in a private network. Subnet addresses are unique and cover many physical networks. Subnets have an identical network-prefix as its IP address. The same method of delivery for the IP addresses is applied to the subnet addresses. The usage of a subnet mask introduces another level of hierarchy into the IP address. After the packet reaches the correct subnet environment, the router of the subnet environment compares the extended network prefix, which includes the network prefix and subnet number, to route the packet to the correct subnet. Figure 7 illustrates the subnet address format.

Figure 7. Subnet Address Hierarchy



For an IP addressing application without the usage of CAMs, a bit-by-bit comparison is applied to the subnet mask and Internet address to locate the subnet. It classifies the subnet number for the specific subnet. Instead of using this approach, an individual CAM can be utilized to store the subnet numbers for that particular host of the IP address. When a match is found, the output of the CAM represents which subnet CAM to refer to next. This CAM contains the host numbers for that specific subnet. The same technique as locating the IP address to the MAC address is applied here. Table 4 shows the contents of a CAM and RAM for a subnet application.

Table 4. Example of CAM and RAM for Subnetting

CAM

Subnet Mask	Address
255.255.255.128	0
255.255.255.7	1
255.255.38.63	2

CAM

Host	Address
193.25.9.25	0
193.25.9.42	1
193.25.9.34	2

RAM

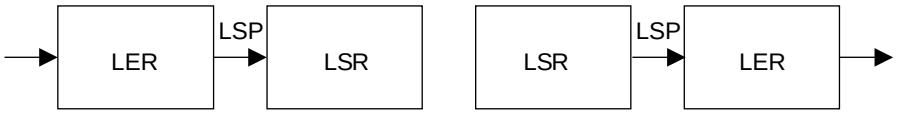
Address	MAC Address
0	00-01-03-32-97-A6
1	00-05-07-63-83-A8
2	00-05-03-42-82-A4

MPLS

Multi-Protocol Label Switching (MPLS) is a standard technology that overlays a protocol on top of IP networks to combine two Layer-2 technologies such as Internet Protocol (IP), Asynchronous Transport Mode (ATM), and frame relay network networks. This eliminates the need for overlaying protocols specific to each Layer-2 technology. Because label switching removes the need for Layer-3 forwarding, MPLS works with Layer-3 routing and Layer-2 switching. MPLS forwards packets of information based on the contents of the labels instead of IP addresses like the traditional packet forwarding. Packet forwarding is implemented in the hardware instead of the software. These characteristics allow faster delivery of the packets. Some results of MPLS applications include network scalability, packet-forwarding performance, and network interoperability.

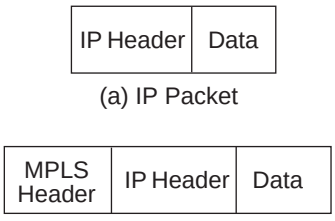
An MPLS network is composed of Label Edge Routers (LERs), Label Switch Paths (LSPs), and Label Switch Routers (LSRs). LSPs are the paths that the packets travel along to connect the LERs and LSRs. They are used to increase performance and avoid network congestion and jitter. Figure 8 shows a simple diagram of the path that the MPLS packets travel along.

Figure 8. Path of MPLS Packets



Upon entering the network, the LER converts the IP packets into MPLS packets. An MPLS shim header is attached to the beginning of the IP packet to forward it as an MPLS packet. This header is placed between Layer 2 and Layer 3 of the OSI model. Figure 9 illustrates the two types of packets.

Figure 9. IP and MPLS Packets



The ispXPLD 5000MX CAM can be utilized as a database to match the destination address to the label. The destination addresses are stored according to the label number in the CAM. Table 5 shows the contents in the CAM for an LER.

Table 5. Example of Database at an LER

CAM

Destination/IP	Address (Label)
125.50.5.2	36
125.50.5.1	40
125.50.5.3	49

Next, the packet travels to the LSR.

LSRs are non-edge routers that examine the packets by completing the instruction of the label and forwarding it to the next destination. They act as either MPLS switches or routers and can perform label-swapping operations. This is done using logic equations. LSRs deal with packets from Layer -2 to MPLS and MPLS to Layer -3.

The label of the MPLS packet is compared with the contents of the CAM. Each CAM can contain the acceptable labels for that router as the input and the following destination label as the output.

Table 6 illustrates the contents of the CAM for an LSR. The incoming labels are stored according to the output label of the next hop.

Table 6. Example of Database at an LSR

CAM

Label/In	Label/Out
234	5
40	25

After the packet arrives at the next hop, it follows this method of delivery until it reaches the egress label switch. The packet travels to the egress LER through the LSP. Once the packet arrives at the destination LER to leave the network, the MPLS label is removed. The LER converts the MPLS packet back into an IP packet.

Conclusion

The CAM is a device within the ispXPLD 5000MX that improves the speed of searching tables and algorithms. It can execute the entire lookup in a single clock cycle. This is suited for many applications like data compression and encryption, IP addressing, and MPLS. RAMs and CAMs can work together to provide certain functions for different applications.

Technical Support Assistance

Hotline: 1-800-LATTICE (Domestic)
1-408-826-6002 (International)
e-mail: techsupport@latticesemi.com