HB0842

TCAM v2.0

Handbook

August 2018

Microsemi

a Microchip company

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.0

Revision 1.0 is the first publication of this document. Created for TCAM v2.0.

Contents

Figures

Tables

2 Introduction

2.1 Overview

Content Addressable Memory (CAM) allows user to enter a search **word** (or binary IP address) and search the entire memory in a single operation, which returns one (or more) matches. Ternary Content Addressable Memory (TCAM) is a more flexible type of CAM and allows the user to store a full prefix or mask.

TCAM is designed in such a way that the user supplies a data word and the CAM searches its entire memory to see whether that data word is stored anywhere in it. If the data word is found, then the CAM returns a list of one or more storage addresses where the word was found.

2.2 Features

TCAM IP supports the following features:

- CAM tables:
  - CAM tables provide only two results: 0 (True) or 1 (False)
  - CAM is useful for building tables that search on exact matches such as MAC address tables
- Ternary modes
  - Standard Ternary Mode: TCAM provides three results: 0, 1, and "X" (do not care bit)
- The output match address can be:
  - Binary encoded
  - One hot encoded
  - Multi-match (many-hot encoded)
- Supports multiple match address
- Supports configurable CAM data, mask, and address widths
- High throughput:
  - One lookup per clock cycle with a fixed latency of six clock cycles
- Is implemented using LSRAMs

2.3 Core Version

This handbook is for TCAM IP version 2.0.

2.4 Supported Families

- PolarFire®

2.5 Device Utilization and Performance

Table 1 lists the resource utilization of TCAM IP for TCAM 64x32 configuration when RAM_MODE is set to 0 (zero).

Table 1. TCAM IP Device Utilization

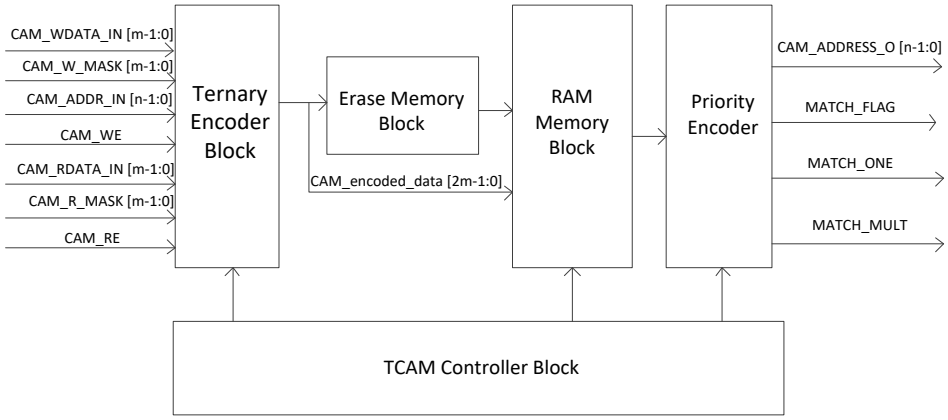
Family	FPGA Resources				Utilization		Performance
	4LUT	DFF	uSRAM	LSRAM	Device	Percentage	
PolarFire	4456	3162	6	16	MPF300T-1FCG1152E	1.49	125Mhz

Note: The data listed in this table is indicative only. The overall device utilization and performance of the core is system dependent.

3 Functional Description

This section describes functional description for TCAM in detail.

Figure 1 TCAM IP Block Diagram



For any write operation, if the ternary CAM mode is enabled and the CAM_WE signal is set to high, then the ternary encoder encodes the incoming CAM_WDATA_IN. The incoming CAM_WDATA_IN is then stored in the Erase Memory at the address location CAM_ADDR_IN.

At the same time, the erase memory erases the CAM write address in the RAM memory block, so that no two data are written at the same location. On the next clock cycle, CAM_WDATA_IN is written at the CAM write address (CAM_ADDR_IN) to the RAM memory block. CAM write address is written to the RAM memory block.

For the read operation, if the CAM read enable (CAM_RE) signal is high and the ternary CAM mode is enabled and the write operation is not in progress, then the incoming CAM_RDATA_IN will pass through ternary encoder block, where the data is encoded. The encoded data is then read from the RAM memory block and the CAM address is returned in the next clock cycle.

The CAM address is then passed to the priority encoder block. If the same data is stored at multiple locations, then the address with the highest priority is sent as an output of the priority encoder block.

Following section describes the internal blocks of the TCAM IP:

- RAM Block
- Erase RAM
- Ternary Encoder Block
- Priority Encoder Block
- TCAM Controller

3.1 RAM Block

For TCAM IP, the RAM block is implemented in two port LSRAMs. Each memory block behaves like a matrix, where each possible data or address combination is represented by one RAM-bit, as CAM address is implemented using one-hot encoding in RAM block.

For example, to implement a Binary CAM of 8x3, where data is 3-bit wide (n=3-bit wide and m=8 words deep), a memory of aspect ratio of 8x8 is required. Hence, a CAM 2^m x n whose input data is n-bit wide will require an LSRAM of 2n words deep to support all possible combinations of input data. The CAM address is implemented in the form of one hot encoding. Similarly, to support all possible combinations of CAM address, an LSRAM of bit width 2^m is required.

Note: Any CAM data can be stored in multiple address locations while a single address cannot be mapped to multiple CAM data.

3.2 Erase RAM

The write operation is completed in two phases:

- **First Phase:** Erase operation
- **Second Phase:** Write operation

The erase function is performed using an Erase RAM, which is implemented using µSRAMs. The Erase RAM stores the CAM data. The address of the Erase RAM is the CAM address.

3.3 Ternary Encoder Block

In the ternary mode, CAM supports 1, 0, and X values. For ternary CAM support, incoming CAM read or write data is used with the CAM read or write mask input. For an n-bit input data there is an n-bit CAM mask.

The relation between CAM data and CAM mask is shown in Table 2.

Table 2 Ternary DATA

CAM Data	CAM Mask	Ternary Mode Data
0	0	0
1	0	1
0	1	X (0, 1)
1	1	X (0, 1)

As shown in Table 2, a **0** on CAM mask bit will pass the incoming CAM data as is, while a **1** on CAM mask applies mask to the incoming read or write data and designates the CAM data as X, hence, it can either be: **0** or **1**.

If CAM is configured in ternary mode, then the CAM data needs to be encoded before passing the data to the RAM memory block.

The following example shows the ternary encoding for a 1-bit CAM data and 1-bit CAM mask.

For a 1-bit CAM data in the ternary mode, there will be two bits of encoded data, as the ternary encoder uses both CAM data and CAM mask parameters.

Table 3 shows the ternary encoding of the CAM data. For 1-bit CAM data, if the value of CAM data is **0** and CAM mask is **0**, then bit **1** of the encoded data output A0 is **1** and bit-0 B0 of encoded data output is **0** and encoded output is AB[1:0] is 10.

Similarly, if CAM data is **1** and CAM mask is **0**, then bit A0 is **0** and bit-0 B0 is **1** and the encoded data output is **01**. When the ternary data is in **X** state, then the value of CAM mask is **1**. It indicates that CAM data is masked and its value can be either **0** or **1**, hence, in this case both bits: bit A0 and bit B0 of the encoded CAM will be **1** and the encoded data output is **11**.

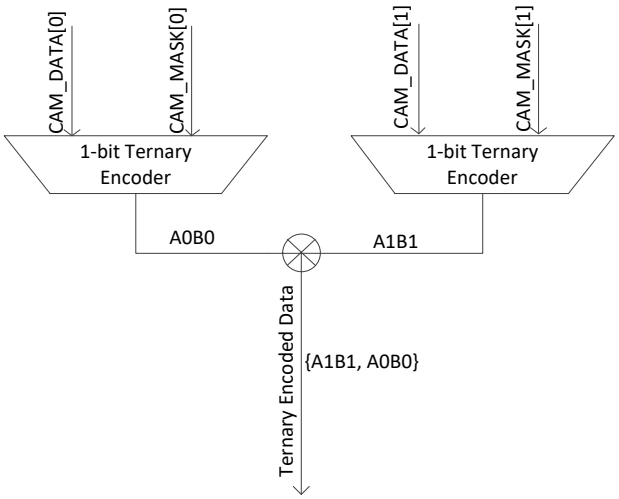
Table 3 Ternary Encoded Data for 1-bit Input Data and 1-bit Mask

Input Data			Ternary Encoded CAM Data		
Data in Ternary Mode	**CAM Data**	**CAM Mask**	**A_0(0)**	**B_0(1)**	**AB[1:0] Encoded Output**
0	0	0	1	0	10
1	1	0	0	1	01
X	0, 1	1	1	1	11

The ternary encoding is performed on each bit independently and then the encoded data of all the bits is appended to generate the ternary encoded data as shown in Figure 2.

The ternary CAM requires twice the RAM size as compared to the binary CAMs as shown in Figure 2.

Figure 2 2-bit Ternary Encoder



3.4 Priority Encoder Block

The priority encoder block receives the one hot encoded address, and converts it to the binary encoded form. For any read operation, if the CAM data returns more than one read address, then based on the priority set in the design, only one address is sent as an output of this block. If the priority encoder is configured such that the highest address is given the highest priority, then the highest CAM address is returned. Otherwise, if the Priority Encoder block is configured for lowest address, then the lowest address is given the highest priority and is transmitted as an output.

3.5 TCAM Controller

The TCAM controller controls all the internal blocks of the TCAM IP.

If the user wants to use the Ternary CAM, then TCAM controller enables the Ternary Encoder block, where the input CAM data is ternary encoded and is passed to the Memory Block.

If the user configures the binary CAM, the TCAM controller bypasses the ternary encoder block and the input CAM data is send to the memory block as is.

3.6 CAM Configuration using PolarFire RAM1K20

TCAM IP supports the following two most efficient CAM configurations for implementing the CAM in single RAM1K20 LSRAM macro.

1. **If RAM_MODE configuration parameter is 0:** In this mode, the smallest CAM supported is of aspect Ratio 32x9 (n=32-bit wide and m=29 words deep), where CAM input data is 9-bit wide, while CAM_address width is 32-bit wide.
2. **If RAM_MODE configuration parameter is 1:** In this mode, the smallest CAM supported is of aspect Ratio 16x10 (n=16-bit wide and m=210 words deep), where CAM input data is 10-bit wide, while CAM_address width is 32-bit wide.

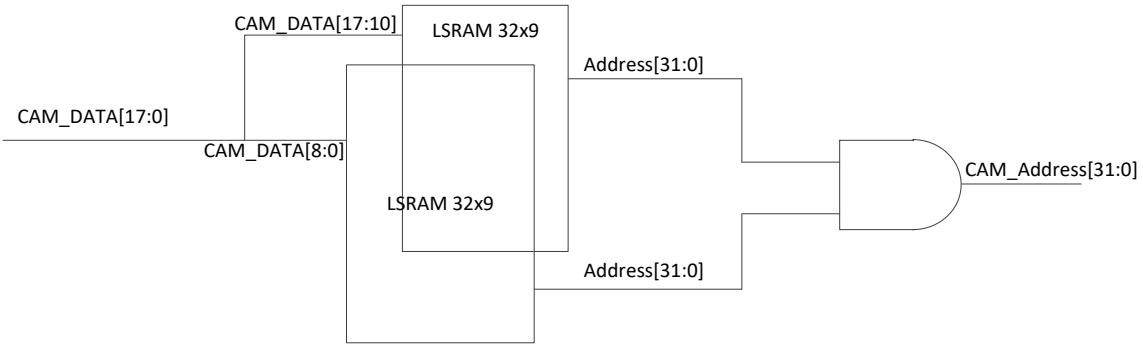
Both the configurations are implemented using a Two Port RAM, where data is written to the RAM through port A, while read from Port B.

The smaller CAMs are cascaded to implement deeper and wider CAMs.

3.6.1 Cascading RAMs when RAM_MODE is "0"

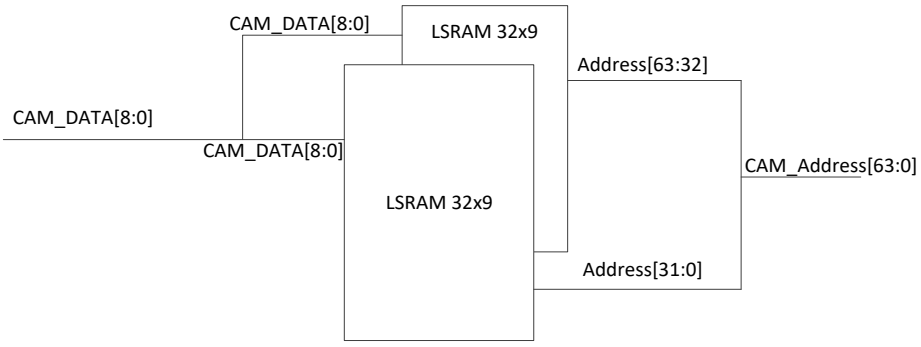
32x18 binary CAM (data width of CAM is 18-bits and address is 5-bit wide (2^5)) is implemented by cascading two LSRAMs as shown in Figure 3.

Figure 3 Deeper CAMs in Cascading Mode (32x18 CAM in Binary Mode)



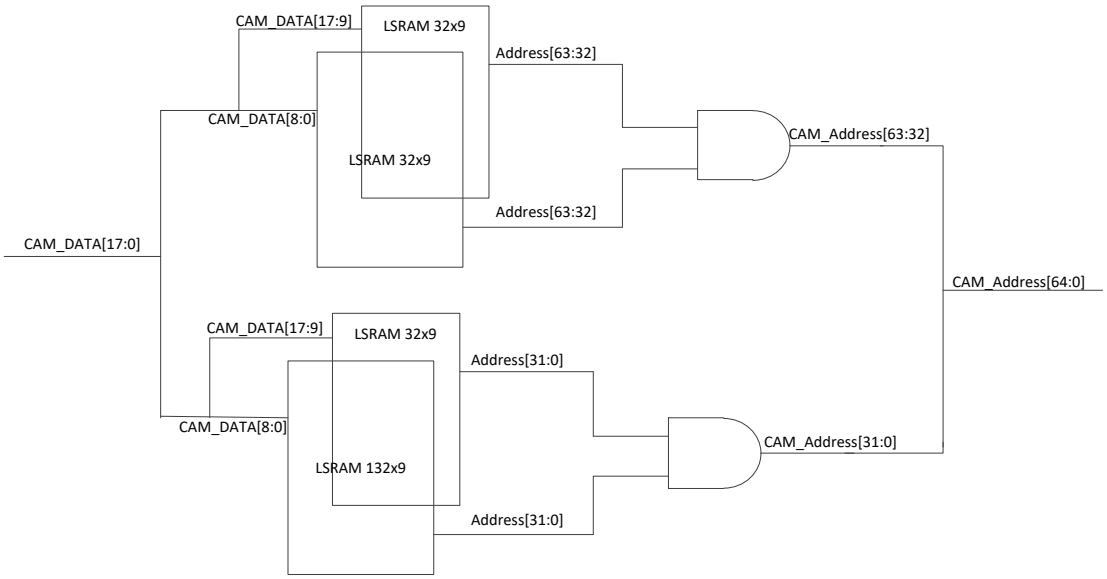
A 64x9 binary CAM is implemented by cascading two LSRAMs as shown in Figure 4.

Figure 4 Wider CAMs in Cascading Mode (64x9 CAM in Binary Mode)



A 18-bit wide and 64 locations deep binary CAM is implemented by cascading LSRAMS as shown in Figure 5.

Figure 5 Wider and Deeper CAM in Cascading Mode (64x18 CAM in Binary Mode)

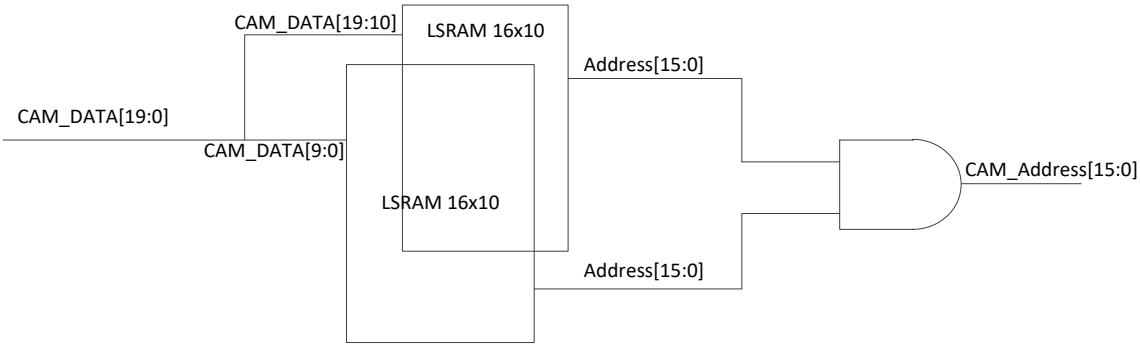


3.6.2 Cascading RAMs when RAM_MODE is "1"

The smaller CAMs are cascaded to implement deeper and wider CAMs.

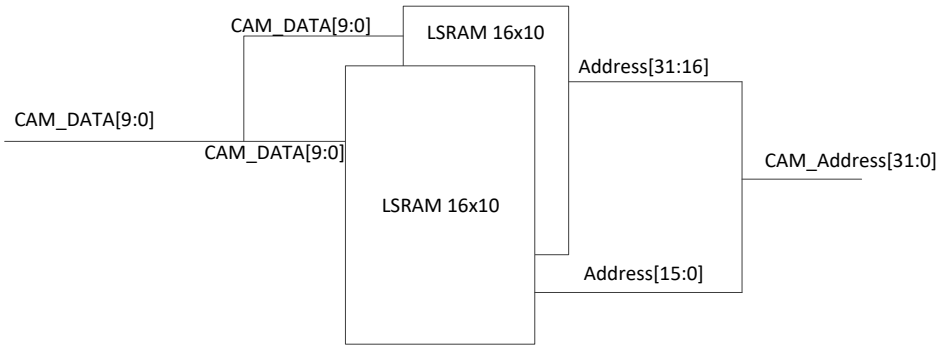
16x20 binary CAM (data width of CAM is 20-bits and address is 4-bit wide (2^4)) is implemented by cascading two LSRAMs as shown in Figure 6.

Figure 6 Deeper CAMs in Cascading Mode (16x20 CAM in Binary Mode)



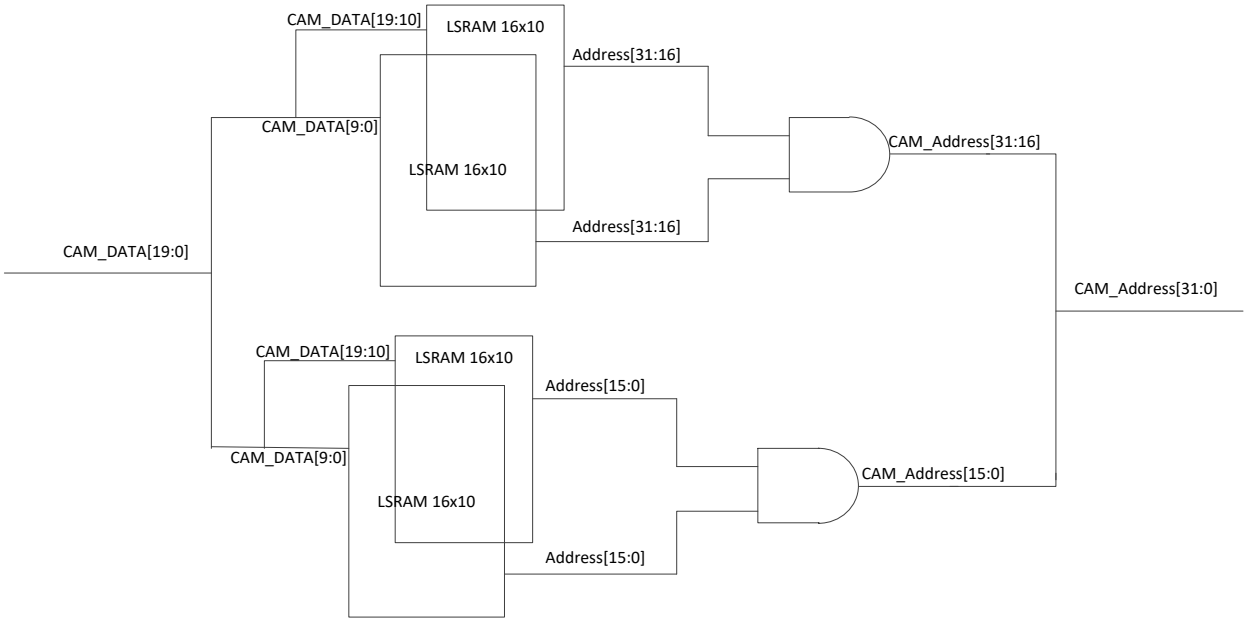
A 32x10 binary CAM is implemented by cascading two LSRAMs as shown in Figure 7.

Figure 7 Wider CAM in Cascading Mode (32x10 CAM in Binary Mode)



A 20-bit wide and 32 locations deep binary CAM is implemented by cascading LSRAMS as shown in Figure 8.

Figure 8 Wider and Deeper CAM in Cascading Mode (32x20 CAM in Binary Mode)



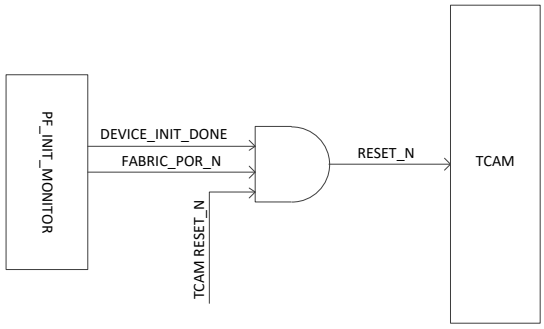
3.7 TCAM IP Clocking Structure

TCAM IP operates in the single clock domain. The clock input to the TCAM IP is provided on the input port CLK_IN.

3.8 TCAM IP Reset Structure

All the reset signals coming as an input to the TCAM IP are active low. Each incoming reset must be qualified with the FABRIC_POR_N and DEVICE_INIT_DONE signals from the PF_INIT_MONITOR block as shown in Figure 9.

Figure 9 PF_INIT_MONITOR Block

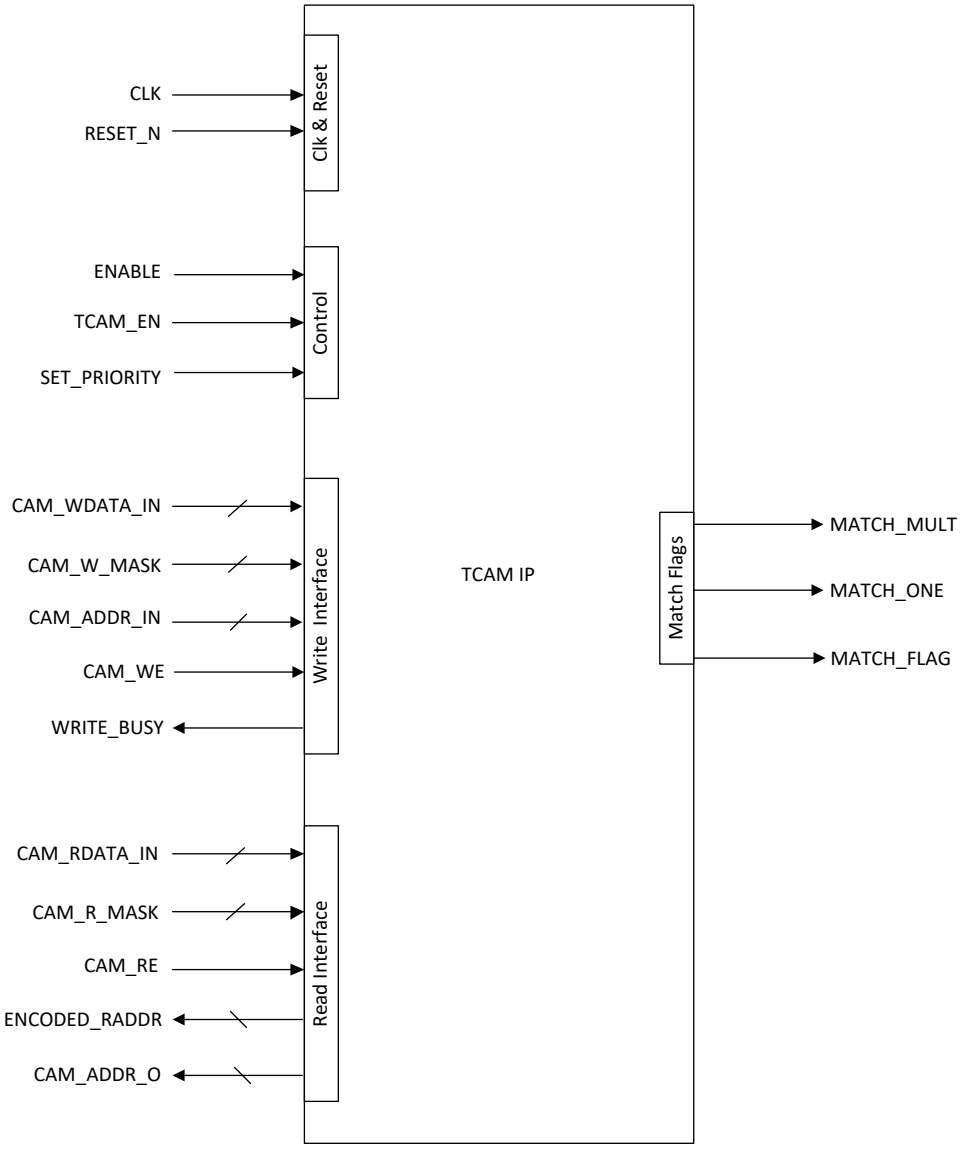


Note: The reset structure shown in above figure is applicable to all the reset signals coming as an input to TCAM IP.

4 Interface

Figure 10 shows the interface signals of the TCAM IP.

Figure 10 TCAM IP Interface Signal Description



4.1 Configuration Parameters

Table 4 lists the configuration parameters used in the hardware implementation process of TCAM IP.

Table 4 Configuration Parameters

Name	Default	Description
RAM_MODE	0	If set to **0**, then configure the Two Port RAM in 512x32. If set to **1**, then configure the Two Port RAM in 1024x16.
TCAM	1	If set to **1**, then configure the IP in TCAM mode. If set to **0**, then configure the IP in Binary CAM mode.
CAM_D_WIDTH	32	CAM Data Width.
CAM_A_WIDTH	6	CAM Address Width.

4.2 Ports

Table 5 lists the port descriptions.

Table 5 Port Description

Name	WIDTH	Type	Description
CLK	1	Input	Input clock signal.
RESET_N	1	Input	Active Low. Asynchronous input reset signal.
ENABLE	1	Input	If set to **1**, then TCAM IP is enabled.
TCAM_EN	1	Input	If set to **1**, then TCAM IP works in Ternary CAM mode. If set to **0**, then TCAM IP works in Binary CAM mode.
CAM_WE	1	Input	If set to **1**, then CAM write transaction is initiated.
CAM_WDATA_IN	CAM_D_WIDTH	Input	CAM_WDATA_IN is the data that is to be written to the CAM during any write operation.
CAM_W_MASK	CAM_D_WIDTH	Input	CAM_W_WASK defines CAM write mask. This signal masks the corresponding CAM_WDATA bits for any ternary CAM write operation. **Note:** This signal is only valid for Ternary CAM operations.
CAM_ADDR_IN	CAM_A_WIDTH	Input	CAM_ADDR_IN defines the CAM write address where incoming CAM write data resides.
CAM_RE	1	Input	If set to **1**, then CAM read transaction is initiated **Note:** CAM_RE should be initiated by the user only when WRITE_BUSY signal is low.
CAM_RDATA_IN	CAM_D_WIDTH	Input	CAM_RDATA_IN is the data to be searched from the CAM memory during any read operation.

Name	WIDTH	Type	Description
CAM_R_MASK	CAM_D_WIDTH	Input	CAM_R_MASK defines CAM read mask. This signal masks the corresponding CAM read data bits for any ternary CAM read operation. **Note:** This signal is only valid for Ternary CAM operations.
CAM_ADDR_O	CAM_A_WIDTH	Output	CAM_ADDR_O signal indicates the CAM output address where the incoming CAM_RDATA resides.
WR_BUSY	1	Output	If set to **1**, then this signal indicates the TCAM write is in progress. TCAM read transaction should not be initiated until the write_busy signal is low.
MATCH_FLAG	1	Output	If set to **1**, then this signal indicates that the incoming CAM_RDATA_IN resides in atleast one of the CAM address.
MATCH_ONE	1	Output	If set to **1**, then this signal indicates that the CAM_RDATA_IN is available at only one address location.
MATCH_MULT	1	Output	If set to **1**, then this signal indicates that incoming CAM_RDATA_IN is available at multiple address locations.
ENCODED_RADDR	$2^{CAM\_A\_WIDTH}$	Output	One hot encoded CAM output address.
SET_PRIORITY	1	Input	If set to **1**, then the highest order address will be transmitted. If set to **0**, then the lowest order address will be transmitted.

5 Timing Diagrams

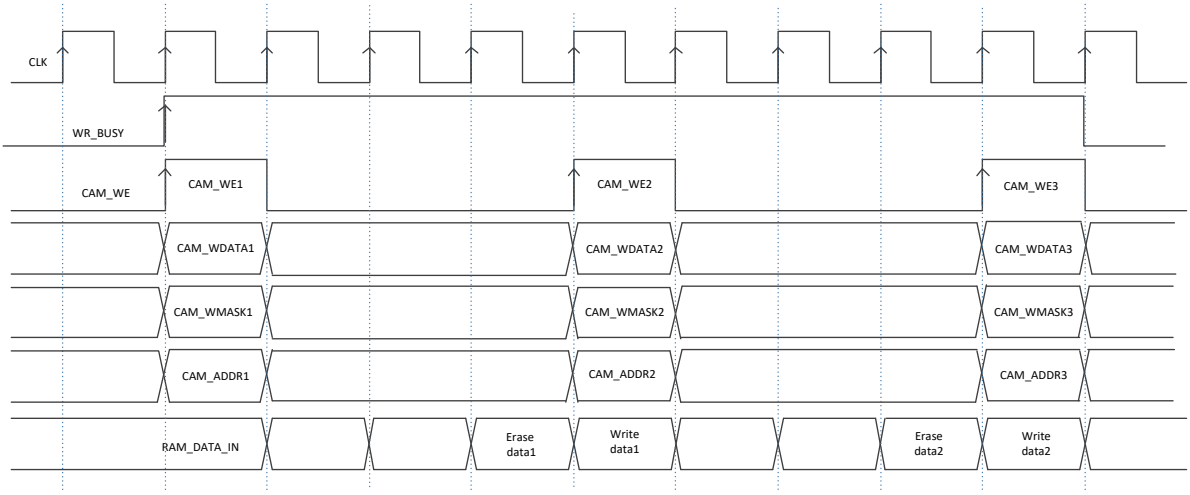
5.1 Memory Write

A write to the memory is completed in two phases:

- **Erase Operation:** In this phase, the data stored previously at the CAM_ADDR_IN (incoming write address) is cleared by writing 0 to that particular location in the memory
- **Write Phase:** In this phase, the incoming CAM_WDATA_IN is stored at the CAM address bit location

Figure 11 shows the timing diagram of the TCAM write interface.

Figure 11 TCAM IP Write Interface



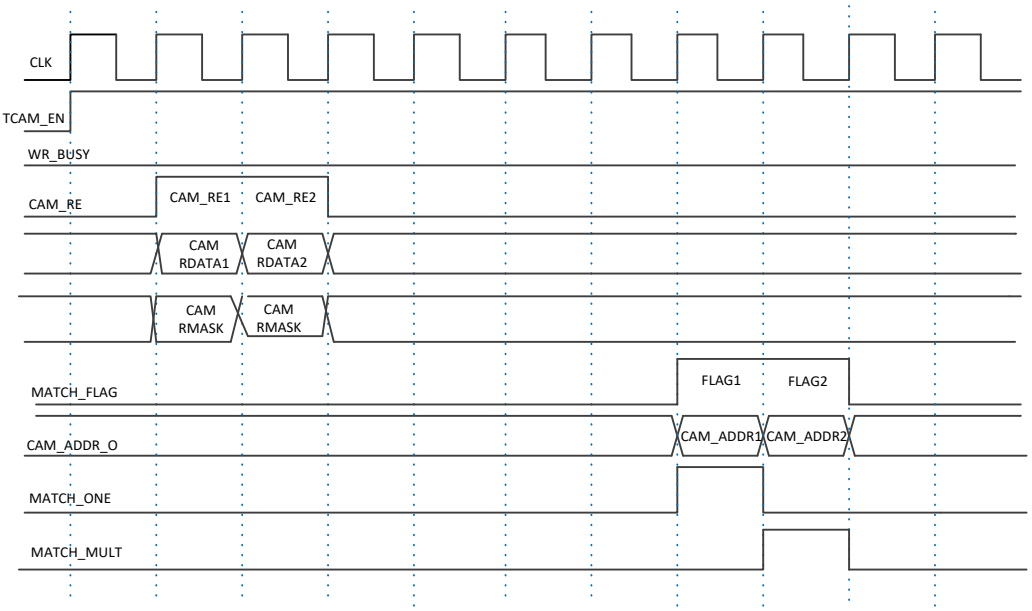
Note: As shown in Figure 11, TCAM write interface timing diagram, the write operation takes four clock cycles. Hence, there should be a gap of four clock cycles between two write transactions initiated by the user. For example, if the first write transaction is initiated at clock cycle T0, then the second write transaction should be initiated after four clock cycles, which is T4, and the third write transaction should be initiated at T8. The IP may not operate properly if the write transaction is initiated before four clock cycles.

5.2 Memory Read

For the read operation, the CAM address is returned from the memory location for the input CAM read data CAM_RDATA_IN. If any of the bits of the one hot encoded CAM address returns a value **1**, then it indicates that the CAM data is stored at the matched CAM address and the Match flag is set high.

If any of the memory location read returns the data value as **0**, then it indicates that the input data is not stored at any CAM address, and match flag is not set high. TCAM IP supports consecutive read outputs with an initial read latency of six clock cycles as shown in the Figure 12.

Figure 12 TCAM IP Read Interface



Note: User should not initiate the CAM Read transaction, when WR_BUSY signal is high. The WR_BUSY is the output signal from the TCAM IP, if high, then this signal indicates that write transaction is in process, and no read transaction should be initiated by the user.

6 Testbench

A testbench is provided to check the functionality of the TCAM IP. Figure 13 shows the top-level block diagram of the TCAM testbench. The CAM Write Master block initiates the write transactions to the TCAM IP while the CAM Read Master block initiates the TCAM read transactions.

For any read transaction, the Scoreboard block verifies whether the CAM read address is as that of expected. If the CAM read address matches the expected address, then the testcase passes, else the test fails.

Figure 13 TCAM IP Top Level Testbench

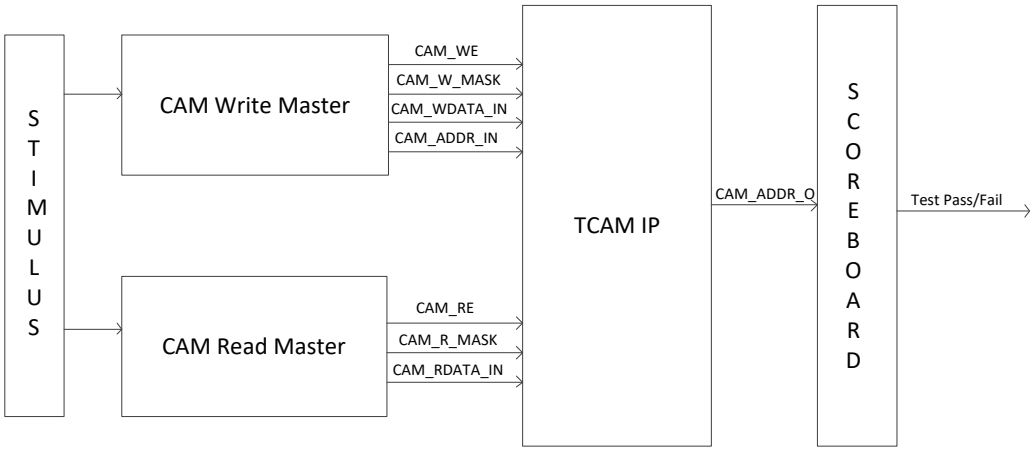


Table 6 lists the testbench parameters that are configured according to the application. All the parameters used in the TCAM IP testbench are located in the `core_parameter.v` file, which is included in the top level TCAM testbench.

The file is available in *//component/Microsemi/Solutions-WiredComms/TCAM/2.0.0/stimulus* directory.

Table 6 TCAM IP Testbench Parameters

Port Name	Default	Description
RAM_MODE	0	If set to **0**, then configure the Two Port RAM in 512x32. If set to **1**, then configure the Two Port RAM in 1024x16.
TCAM	1	If set to **1**, then configure the IP in TCAM mode. If set to **0**, then configure the IP in Binary CAM mode.
CAM_D_WIDTH	32	CAM data width.
CAM_A_WIDTH	6	CAM address width.
NUM_OF_TRANS	10	The values defines the number of transactions for any read or write transfer.
CAM_START_ADDR	6'h00	Start address for any write transaction.
CAM_START_MASK	32'h00ff00ff	Mask value for any read or write transaction.
CAM_START_DATA	32'h55aa55aa	Data value for any read or write transaction.

7 Tool Flow

7.1 License

TCAM IP clear RTL is license locked and the encrypted RTL is freely available.

7.1.1 Encrypted

- Complete RTL code is provided for the core, enabling the core to be instantiated with SmartDesign
- Simulation, Synthesis, and Layout can be performed with Libero software. The RTL code for the core is encrypted using the IP encryption solution.

7.1.2 RTL

Complete RTL source code is provided for the core and testbenches.

7.2 SmartDesign

The TCAM IP is available for download in the Libero IP catalog, Solutions-WiredComms. Once it is listed in the catalog, then the core can be instantiated using the SmartDesign flow. For more information, on how to use SmartDesign to configure, connect, and generate cores, see to the Libero SoC online help.

After configuring and generating the core instance, basic functionality can be simulated using the testbench supplied with the TCAM IP. The testbench parameters can be updated in the `core_parameter.v` file, which is available in:

//component/Microsemi/Solutions-WiredComms/TCAM/2.0.0/stimulus directory.

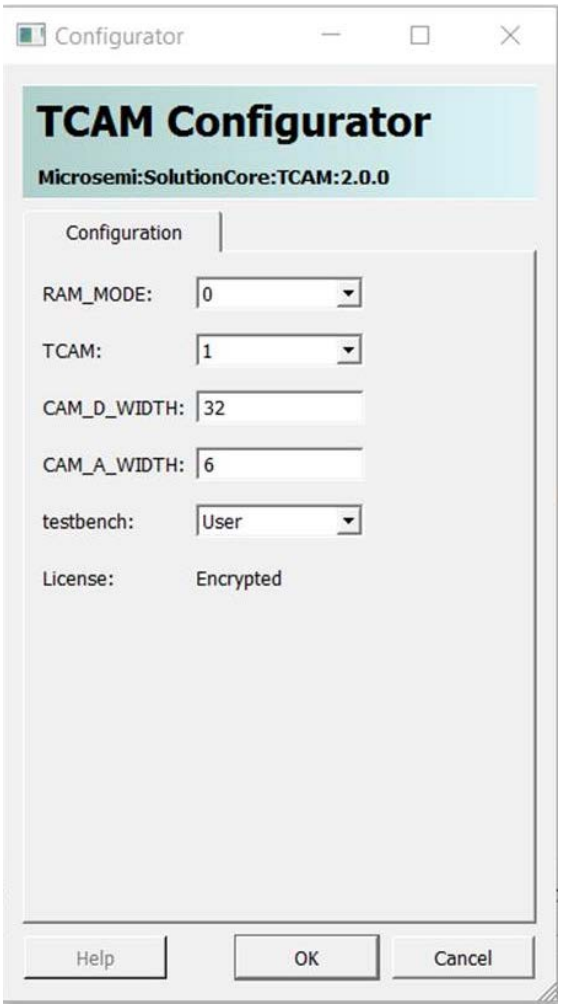
For more information on testbench parameters see, Table 6. The TCAM IP can be instantiated as a component of a larger design. TCAM IP is compatible with Libero SoC PolarFire.

7.3 Configuring TCAM IP in PolarFire

The TCAM IP is configured using the configuration user interface within SmartDesign.

An example of the user interface for the PolarFire family is as shown in Figure 14.

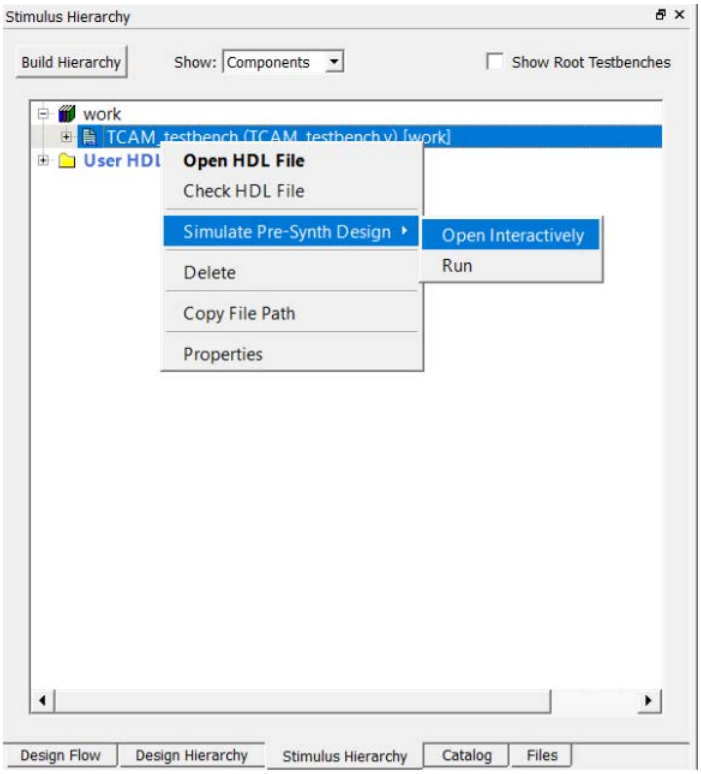
Figure 14 Configuring TCAM IP in PolarFire



7.4 Simulation Flows

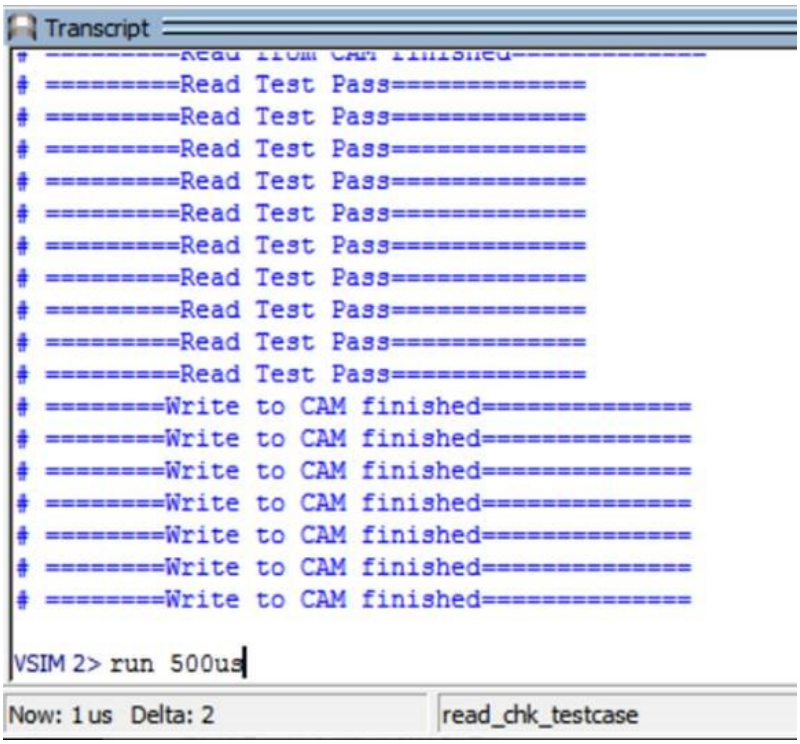
Following are the steps to simulate TCAM IP using ModelSim ME 10.5c through Libero SoC Software. To Execute TCAM IP testbench, Right-Click `TCAM_testbench.v`, from **Simultaion Hierarchy** Tab, select **Simulate Pre-Synth Design > Open Interactively** as shown in Figure 15.

Figure 15 Simulating the Pre-Synthesis Design Option



ModelSim starts the testbench execution. Run the testbench for 500us. To do this, enter the command `run 500us` in Transcript window as shown in Figure 16.

Figure 16 Command Window



See the results in transcript window, to verify if TCAM IP passed or failed. When the received data matches the expected data, TCAM IP test passes. Otherwise, the TCAM IP test fails.

Figure 17 Test Result

7.5 Synthesis in Libero

Click **Synthesis** in the Libero software. The Synthesis window displays the Synplify® project. Set Synplify to use the Verilog 2001 standard, if Verilog is being used. To run the synthesis, click **Run**.

7.6 Place-and-Route in Libero

Click **Layout** in the Libero software to invoke the Designer. TCAM IP does not require any special place-and-route settings.

8 Ordering Information

8.1 Ordering Codes

Order TCAM IP through your local Microsemi sales representative.

Use the following number convention when ordering: TCAM IP -XX. XX is listed in Table 7.

Table 7 Ordering Codes

XX	Description
RM	RTL multi-use multiple-site license

9 Appendix A

9.1 Use Case: Binary CAM 8X3

Table 8 lists the mapping of the CAM data and address in the memory block for the implementation of 8x3 CAM.

Table 8 Mapping CAM Data with CAM Address using LSRAM

	RAM Data / CAM Address							
RAM Address / CAM Data	0	1	2	3	4	5	6	7
000	0	0	0	0	0	0	0	1
001	0	0	0	1	0	1	0	0
010	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
100	0	1	0	0	0	0	0	0
101	1	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0
111	0	0	0	0	1	0	0	0

As shown in Table 8, a CAM with a data width of 3-bits will require eight memory locations to support all possible combinations of CAM data, while a CAM with a depth of eight address locations requires 8-bit wide memory to support all combinations of CAM address.

The CAM data stored in the memory block is interpreted as:

- CAM data 000 is mapped to CAM address 7
- CAM data 001 is mapped to CAM address 3 and 5
- CAM data 010 is not mapped to any address
- CAM data 011 is not mapped to any address
- CAM data 100 is mapped to address 1
- CAM data 101 is mapped to address 0
- CAM data 110 is not mapped to any address
- CAM data 111 is mapped to address 4

Microsemi Headquarters
One Enterprise, Aliso Viejo, CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
email: sales.support@microsemi.com
www.microsemi.com

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services.
Learn more at www.microsemi.com.

50200842-1.08/18