VTT001 (v1.0) 24 July 2000

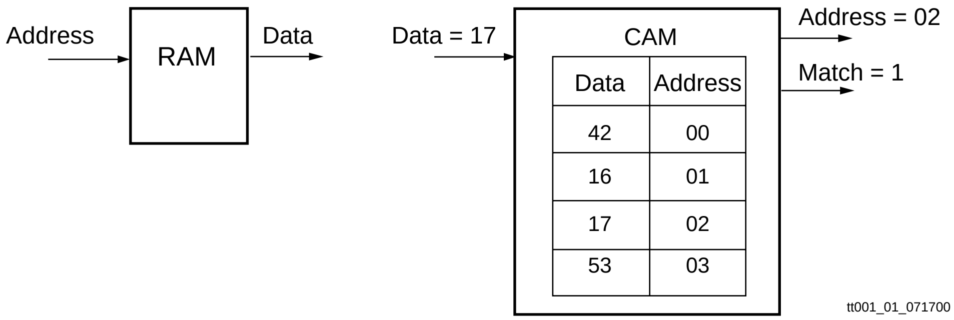
Content Addressable Memory

Introduction

CAM enables accelerated data searches to be performed in a storage array. CAM is well suited for many applications, including table look-up, pattern recognition, cache tags, Ethernet addressing, and ATM switching. Virtex, Virtex-E, Virtex-EM devices offer both on-chip and off-chip solutions customized for each application. Smaller CAM applications are configurable in Virtex devices through the use of internal SRL shift registers, True Dual-Port™ block RAM, and distributed RAM. For larger-scale CAM applications, Virtex devices use SelectI/O™ and delay-lock loop (DLL) technologies to provide a speedy and robust interface to external CAM designs.

CAM Definition

CAM is designed to enhance data retrieval speed from a particular location in a storage array. Instead of using an address to read the data, such as a RAM, the data is supplied as an input to locate the address, as shown in Figure 1. CAM determines if the data is found within a storage array. When a match is found, the CAM outputs the address location in the array and the match signal output value is set High.



Data	Address
42	00
16	01
17	02
53	03

Figure 1: **CAM Compared to RAM in Read Mode**

A basic CAM element consists of input data, a storage location, and a comparator. It is characterized by the following features:

- word size (width)
- number of words or locations (depth)
- match or compare time (read)
- write speed
- clock frequency
- ternary or binary CAM
- decoded and/or encoded address outputs
- multi-match support

Figure 2 is a CAM read waveform example illustrating the interplay of different CAM features. DATA_IN is provided as the input to be searched within a storage array. During the match cycle, DATA_IN is compared against all locations within the CAM storage array. If a match to DATA_IN is found, the matched location is provided as the DECODED_ADDRESS. During the encode_cycle, the MATCH_OK signal goes High and the address location (ADDR) is provided at the output.

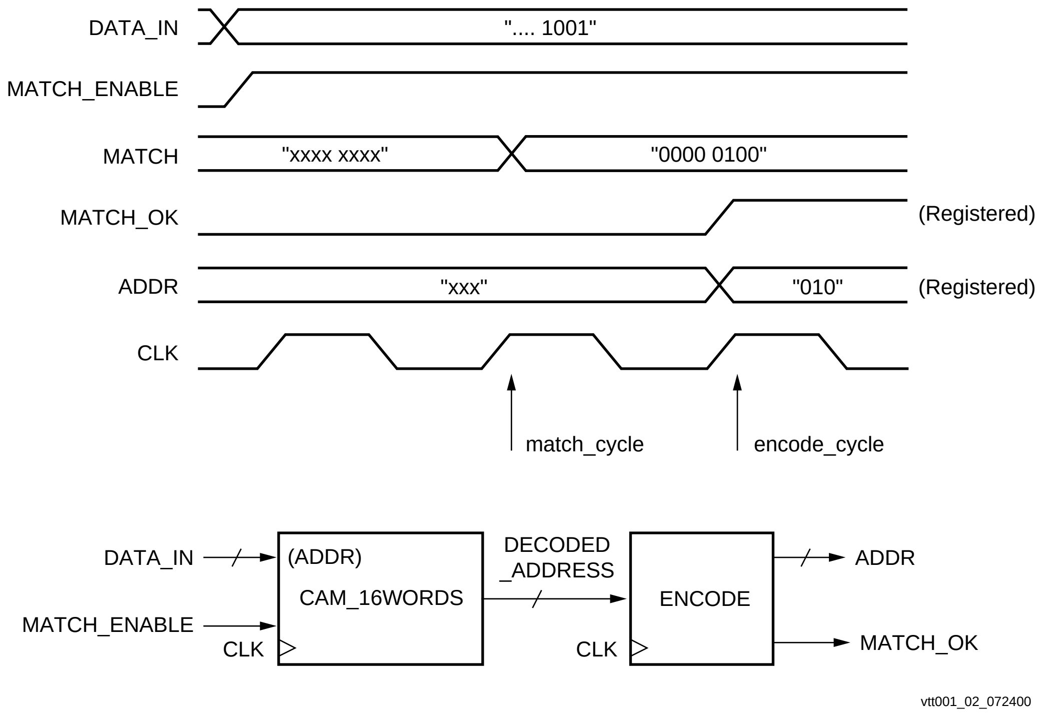


Figure 2: Waveform of a CAM Read Operation

CAM Modes

There are several CAM modes: single-match, multiple-match, and ternary CAM. In single-match mode, only one location in the storage array contains input data.

If the input data can be contained in multiple locations within the storage array, it is called multiple-match mode. In multiple-match mode, the decoded address is required to assign a priority that determines which particular address/location is selected.

Ternary CAM supports "don't care" bit(s) as input data. For example, if the second bit of the data is a "don't care" (Data = 00X1), the address containing either "0011" and "0001" matches the input data.

CAM Applications

CAM can be used in many applications including table look-up, Ethernet address, ATM switches, data compression, IP address resolution, and recognition algorithm. In this section, applications of network switch and ATM switch are illustrated.

Network Switches

In network switch applications, CAM is used to store switch port numbers and network addresses. For each incoming packet, the destination-port address is located, in one clock cycle, by searching for the port number in CAM. The CAM output is the corresponding port address as shown in Figure 3.

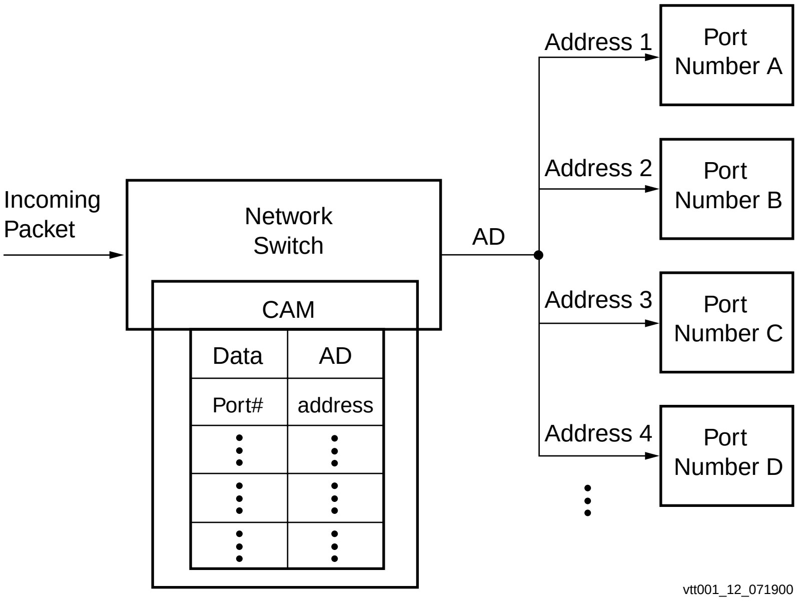


Figure 3: **Network Switch Application**

ATM Switches

Due to the connection-based protocol, ATM switches must translate each ATM cell address at every point along the routing path. Each ATM cell address is contained in a 5-byte, 2-field header. The switch maintains a table of the current ATM cell addresses with the corresponding RAM addresses. For example, for one thousand active ATM connections, the cell address can be stored in a 1024-location CAM.

The output of the CAM is used to address a RAM, as illustrated in Figure 4. More details are available in application note **XAPP202** "Content Addressable Memory in ATM applications".

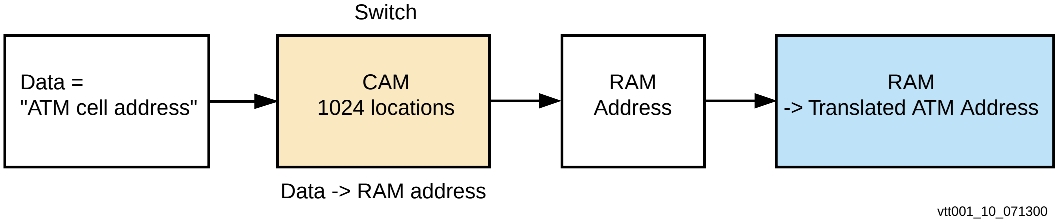


Figure 4: **ATM Switch Application**

CAM Configurations Using Virtex Devices

With Virtex devices, CAM configurations are customizable to fit the requirements of the different applications. CAM operates at maximum efficiency when matches are done in one clock cycle. An on-chip CAM eliminates the delay that occurs when going from on-chip to off-chip. On the other hand, dedicated external CAMs serve best in applications where larger data storage is required. When connecting to external CAMs, speed and efficiency of the interface are the most critical requirements of an integrated CAM design.

The following CAM configurations are implemented in Virtex devices:

- Embedded Wide CAM
- Embedded Deep CAM
- Virtex Series Interfacing to External CAM

Virtex series CAM implementations are classified as follows:

CAM Implementation	Width	Depth	Size
Embedded wide CAMs	up to 272-bit	up to 128 locations	34 K bits
Embedded deep CAMs	8-bit to 40-bit	up to 1024 locations	35 K bits
External CAM	32-bit to 272-bit	up to 256 locations	2M to 8M bits

Embedded Wide CAM

Virtex SRL shift register (16-bit shift registers in a LUT) is best suited for embedded wide CAMs. CAM implementations with SRL shift registers have the following features:

- Fast match and longer write access
- Configure a 1-word depth by 4-bit width CAM per LUT
- Match in one clock cycle
- Write in 16 clock cycles
- Provide both encoded and decoded address
- Support single and multiple-match modes
- Cascadable with no additional logic required

Figure 5 is an illustration of an 8-bit implementation using SRL shift registers. The carry-chain is used to link up the SRL shift registers to configure wider CAMs. Since each additional 4 bits in CAM width requires an additional SRL shift register and is connected by the same carry-chain, a wider CAM has minimal penalty in performance, because only one carry-chain delay is added. The number of words are equivalent to the number of SRL shift register columns used. CAM configured using SRL shift registers is therefore very flexible and expandable.

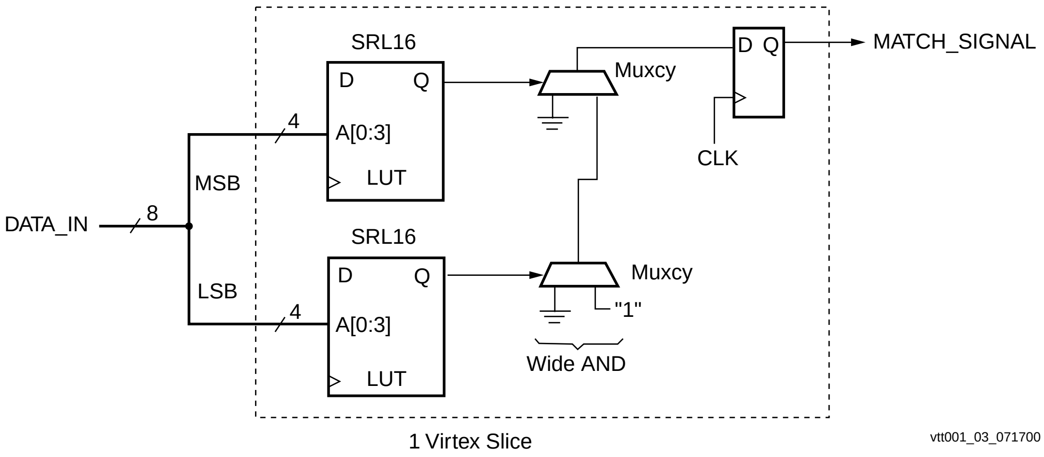


Figure 5: SRL Shift Register Implementation in an 8-bit CAM

Embedded Deep CAM

Deep CAM is best configured using Virtex series block RAM. CAM implemented using block RAM has the following features:

- Fast match and write access
- Configure a 16-word depth by 8-bit width CAM per block RAM
- Match in one clock cycle
- Write in one clock cycle (and erase in one clock cycle)
- Provide both encoded and decoded address
- Support single- and multiple-match modes
- Fully synchronous match and write ports (independent)

Figure 6 provides an illustration of a cascadable CAM implementation using block RAM. Several 16 x 8 block RAM elements can be linked together to implement a deeper CAM. The outputs of the different 16 x 8 block RAM elements are merged as the global decoded address. A 64 x 8 CAM consists of four 16 x 8 block RAM elements, with a 64-bit decoded address. The first 16 x 8 block RAM element corresponds to the LSB (position 0 to 15) of the 64-bit CAM. The second 16 x 8 block RAM element corresponds to the next 16-bit (position 16 to 31), and so on.

Similar to the 16 x 8 CAM, the 64 x 8 CAM (or any deeper size CAM) takes one clock cycle to generate the decoded address. With each addition 16 x 8 block RAM element, the address width of the encoder becomes larger.

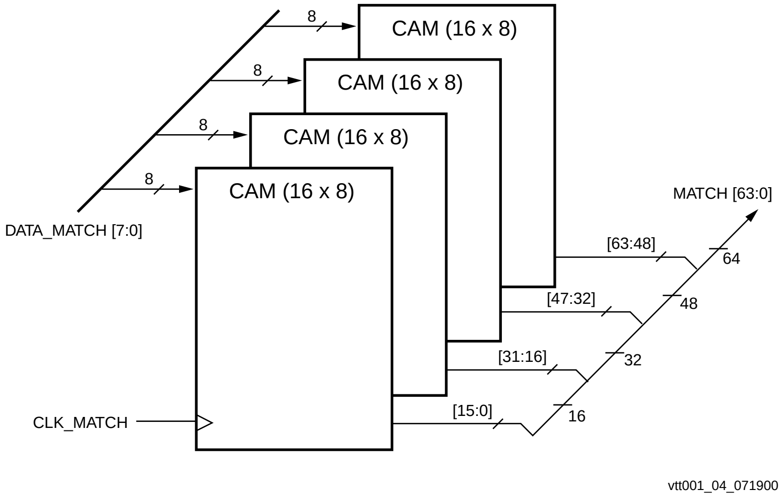


Figure 6: 64-word x 8-bit CAM Implementation in Block RAM

Virtex Series Interfacing to External CAM

For large-scale CAM applications, Virtex devices can be interfaced to external CAMs via SelectI/O technology. Virtex devices provide a robust interface to read, write, and search external CAMs for large, intensive data processing. Many off-the-shelf CAM devices can interface seamlessly with Virtex devices, including Lara Networks, Netlogic Microsystems, SiberCore Technologies, and Music Semiconductors.

Virtex Series Advantages

This section examines differences between the CAM implementation in Xilinx Virtex devices and Altera APEX E devices.

Virtex devices support single- and multiple-match modes, using the same model for CAM configurations of any width or depth via SRL shift registers or block RAM. As illustrated in Figure 7, a decoded address, which supports multiple-match mode, is provided on the first clock cycle. An encoded address and a match flag are generated on the second clock cycle.

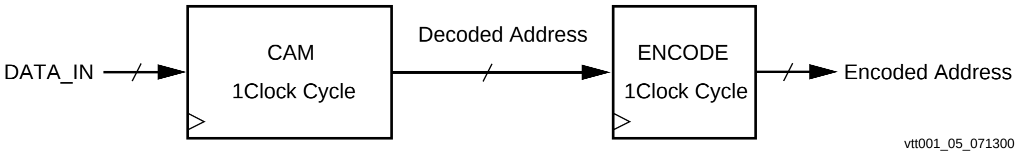


Figure 7: Xilinx Solutions

The Altera APEX E CAM solution is based on a fixed-size Embedded System Block (ESB). Depending on single-match or multiple-match mode, a different model implementation is required. As shown in Figure 8, single-match mode is used only with widths no larger than 32 bits. Single-match mode provides an encoded address on the first clock cycle and no decoded address. One ESB implements a 32-word x 32-bit CAM in single-match mode.

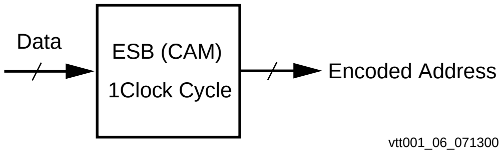


Figure 8: **Altera Solution for Widths < 32 Bits or Single-Match Mode**

Multiple-match mode is required for CAM widths greater than 32 bits or when the decoded address is needed. In this mode, the input data is limited to 31 bits per ESB, because one bit is used for the select feature. Two clock cycles (twice the amount of time required by Virtex series solutions) are needed to get the decoded address, and more clock cycles are required to obtain the encoded address, as illustrated in Figure 9.

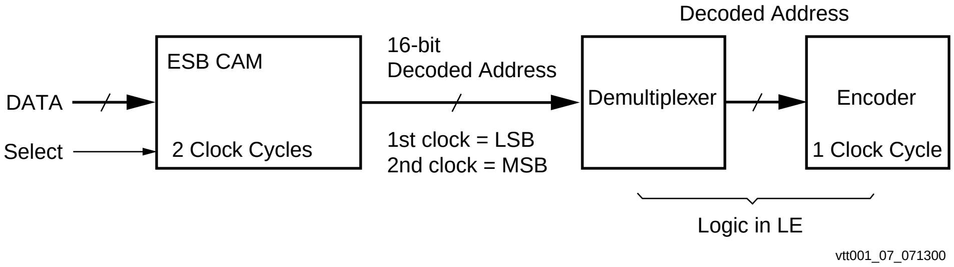


Figure 9: **Altera Solution for Widths > 32 Bits or Multiple-Match Mode**

In Table 1 details of Virtex and APEX E timing models are examined. The timing output of Virtex devices for any width or depth in the SRL shift register or block RAM implementation is shown.

Table 1: **Virtex CAM Timing**

Clock cycle	**1**	**2**	**3**	**4**	**5**	**6**
Input data	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6
Output decoded address		Result_1	Result_2	Result_3	Result_4	Result_5
Output encoded address			Result_1	Result_2	Result_3	Result_4

A new search is supported at each clock cycle. After the input data (Data_1) is supplied on clock cycle 1, the output decoded address (Result_1) is available on clock cycle 2. At the same time, a new search of Data_2 can begin. On clock cycle 3, the output decoded address is then encoded to generate the output encoded address (Result_1) and the match signal.

In comparison, a new search in APEX E begins only every 2 clock cycles. Table 2 illustrates the timing of the Altera CAM implementation for CAM width or depth of > 32 or in multiple match mode. APEX E processes only 16 bits at a time and requires two clock cycles to de-multiplex the 16-bit odd and 16-bit even output decoded address.

To search Data_1, 16 even bits (even_1) are processed on clock cycle 2 and 16 odd bits (odd_1) are processed on clock cycle 3. The output encoded address (Result_1) is not available until clock cycle 4. A new search of Data_2 cannot begin until clock cycle 3. This shows that APEX E CAM has only half the speed efficiency of Virtex implementations. Furthermore, additional logic resources are necessary to cascade the multiple ESB blocks to implement CAM sizes larger than 32-bit or 32-word.

Table 2: **Apex E CAM for Widths > 32 Bits or Multiple-Match Mode**

Clock cycle	1	2	3	4	5	6
Input data	Data_1	xxxxx	Data_2	xxxx	Data_3	xxxx
Output decoded address		even_1	odd_1	even_2	odd_2	even_3
Output encoded address				Result_1	xxxx	Result_2

Virtex and APEX E Performance Comparison

To measure how Xilinx CAM implementations are compared against Altera CAM solutions, the following CAM sizes were implemented with the Xilinx Foundation™ tools and the Altera MegaWizard compiler with multiple match, decoded outputs, and registered I/Os. Table 3 summarizes the performance comparison.

Table 3: **Performance Comparison**

CAM Size	Xilinx (Block RAM)	Altera (ESB)
16 x 8-bit	144 MHz	100 MHz
32 x 8-bit	118 MHz	95 MHz
64 x 8-bit	91 MHz	85 MHz
128 x 8-bit	84 MHz	57 MHz
256 x 8-bit	83 MHz	40 MHz
CAM Size	**Xilinx (SRL Shift Register)**	**Altera (ESB)**
32 x 16-bit	125 MHz	94 MHz
64 x 32-bit	100 MHz	75 MHz
128 x 40-bit	83 MHz	55 MHz
256 x 48-bit	57 MHz	32 MHz
1024 x 16-bit	55 MHz	12 MHz

Figure 10 illustrates how superior CAM performance levels are achieved with Virtex devices using block RAM.

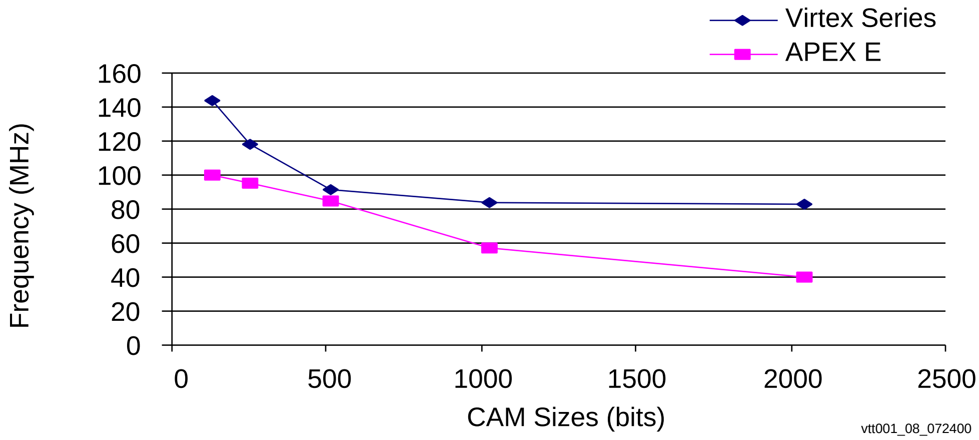


Figure 10: **APEX E and Virtex Series (with block RAM) CAM Performance Comparison**

Figure 11 illustrates how superior CAM performance levels are achieved with Virtex devices using shift registers.

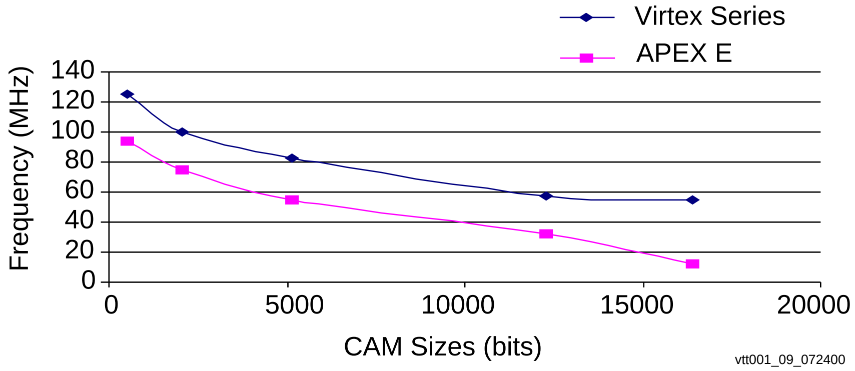


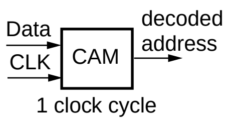
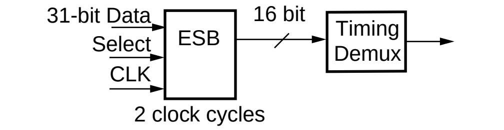
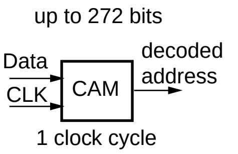
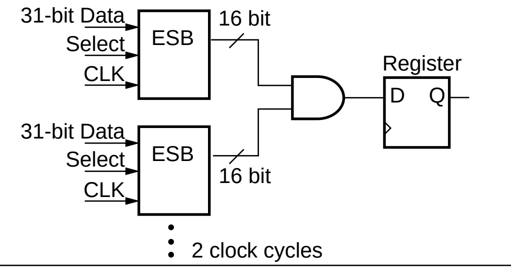
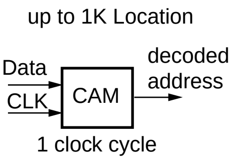
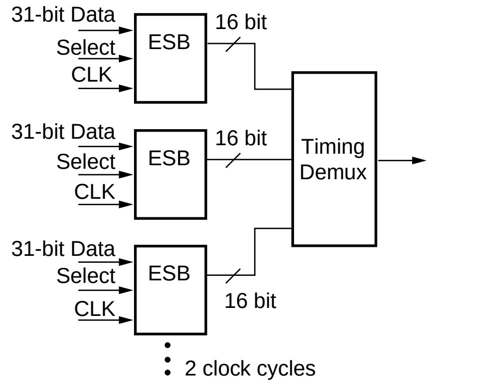
Figure 11: APEX E and Virtex Series (with SRL Shift Registers) CAM Performance Comparison

With Virtex devices in either shift register or block RAM implementation, their performance outperforms APEX E CAM. In addition, as shown in Table 4, Virtex series has significantly more memory remaining that can be used for other functions. In 16 x 8 and 64 x 8 CAM configurations, XCV100E devices have more than twice the block RAM remaining, compared to EP20K60E plus distributed RAM.

Table 4: **Memory Comparison**

	XCV100E-8		**EP20K60E-1**	
CAM Size	CAM use	Remaining	CAM use	Remaining
16 x 8-bit	4 kbits BRAM (1 BRAM)	76 kbits BRAM + Distributed RAM	2 kbits RAM (1 ESB)	30 kbits RAM
64 x 8-bit	16 kbits BRAM (4 BRAMs)	64 kbits BRAM + Distributed RAM	4 kbits RAM (2 ESBs)	28 kbits RAM
256 x 8-bit	64 kbits BRAM (16 BRAMs)	16 kbits BRAM + Distributed RAM	16 kbits RAM (8 ESBs)	16 kbits RAM

Figure 12 provides a summary of what wide and deep CAM implementations require in Virtex devices and APEX E devices.

Feature	Xilinx/Virtex Devices	Altera/Apex	Comments
Multi-Match CAM		 31-bit Data Select CLK ESB 16 bit Timing Demux 2 clock cycles	Virtex devices allow direct access to multi-match outputs.
Wide CAM	 up to 272 bits Data CLK CAM decoded address 1 clock cycle	 31-bit Data Select CLK ESB 16 bit Register D Q 31-bit Data Select CLK ESB 16 bit 2 clock cycles	Virtex shift register solutions automatically support wide CAM without extra logic.
Deep CAM	 up to 1K Location Data CLK CAM decoded address 1 clock cycle	 31-bit Data Select CLK ESB 16 bit 31-bit Data Select CLK ESB 16 bit 31-bit Data Select CLK ESB 16 bit Timing Demux 2 clock cycles	Virtex block RAM or shift register solutions automatically support deep CAM.

vtt001_011_072400

Figure 12: Wide and Deep CAM Implementation Comparison

Table 5 summarizes the advantages of Virtex CAM solutions over APEX E CAM implementations.

Table 5: **Virtex CAM Advantages Summary**

Features	Virtex, Virtex-E, Virtex-EM	APEX E
1 clock cycle multi-match	Yes	No
Width supporting 1 clock cycle match	> 272 bits	32 bits maximum
Depth supporting 1 clock cycle match	any size	32 locations maximum
Encoded address result (> 32 bits)	Each clock cycle (latency = 2)	Multiple clock cycles for each search
Fast CAM search (any width / depth)	Yes - CAM 128 x 40-bit in 12 ns	No - CAM 128 x 40-bit in 36 ns
Flexible solution	Yes	No

In conclusion, Virtex series advantages include:

- Greater flexibility in CAM designs and more efficient use of the implementation
- Support for both single- and multiple-match modes, regardless of CAM width and depth
- More remaining memory plus extra distributed RAM that can be used for other applications
- Support for both 1.8 V and 2.5 V CAM designs

References

Several application notes and reference designs are available to assist with your CAM designs. They can be accessed at the following web site:

http://www.xilinx.com/products/xaw/memory/embedded/cam.htm

- XAPP201, An Overview of Multiple CAM Designs in Virtex Devices
- XAPP202, Content Addressable Memory in ATM Applications
- XAPP203, Designing Flexible, Fast CAMs With Virtex Family FPGAs
- XAPP204, Using Block RAM for High Performance Read/Write CAMs
- XAPP242, Interfacing to the Lara Networks Search Engine Using Virtex Devices

For applications requiring larger size CAM implementations, more information can be obtained from the following vendors:

- Lara Networks (**http://www.laranetworks.com**)
- Netlogic Microsystems (**http://www.netlogicmicro.com**)
- SiberCore Technologies (**http://www.sibercore.com**)
- Music Semiconductors (**http://www.music-ic.com**)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/24/00	1.0	Initial Xilinx release.