Designing Flexible, Fast CAMs with Virtex Family FPGAs

XAPP203, September 23, 1999 (Version 1.1)

Application Note: Jean-Louis Brelet & Bernie New

Summary

Content Addressable Memories (CAM) allow a fast search for specific data in a memory. Each application has different CAM requirements. A CAM design implemented in Virtex™ Family slices offers a flexible approach to CAM depth and width based upon LUTs configured as Shift Registers. This application note describes a fast CAM design finding a match in a single clock cycle. The application note XAPP201 “An overview of Multiple CAM designs in Virtex Family devices” discusses the diverse solutions available when implementing CAM and introduces the specific solution described in this application note.

Xilinx Family

Virtex™ and Virtex™-E FPGAs

Introduction

The application note XAPP201 covers the basic differences between a CAM versus a RAM along with comparing the density and performance of three CAM design solutions in Virtex Family devices.

A CAM allows a concurrent search of input data into the memory. This is the main advantage of a CAM over a RAM. The CAM outputs the corresponding address when a match is found.

This application note introduces a methodology for designing flexible, small to medium size CAMs, in Virtex or Virtex-E slices. By using shift register primitives built into a Virtex slice, a reconfigurable LUT (two LUTs per slice) is used to implement a single clock cycle read CAM. A 4-bit CAM word fits into each LUT. A 32-word by 16-bit CAM would require 128 LUTs. The write operation uses the shift register mode and requires 16 clock cycles.

General Description

Four input AND gates with optional inverters fit in a LUT and are cascadable through the dedicated carry-chain. Each Virtex slice has two LUTs and a fast carry-chain. A column of slices is a high performance decoder using the carry-chain to AND-wire each LUT output. The read operation is equivalent to decoding the input data through the LUT and the carry-chain.

The number of slices per column is dependent on the Virtex Family device. This slice column (or a partial column) is at the same time a comparator and a storage location for a CAM word when the LUTs are reconfigurable. The shift register mode (SRL16E primitive) is used in each LUT to allow a write operation into the decoder.

Several slices columns are equivalent to an array of CAM words. The number of columns also depends upon the Virtex Family device. The designer can easily find the correct Virtex or Virtex-E device to implement a particular CAM size.

The input data bus is distributed to each single word or LUT through the high performance routing capabilities of Virtex Family devices. A 32 word by 16-bit CAM has 16 high fanout nets of 32 end-points.

Design Overview

The reference design described in this application note has an adjustable word width and depth. The output of the basic CAM module is a decoded (or “one-hot”) address. Often the critical timing parameter is the access time to the CAM during a match operation. This design provides a single clock cycle decoded address as fast as 7.5 ns in a Virtex Family device (-6 speed grade). A decoded address is defined as one line per memory location. The number of output lines is the number of words in the CAM. If an output is high, the corresponding word matches the input data.

This reference design also proposes an encoding module to generate the output address. Because of the hierarchical VHDL code structure, it is easy to use this module for various CAM sizes according to the designers need. Usually the encoder requires only one additional clock cycle to generate both the output address and a match flag. A wide OR gate of all the decoded addresses creates this match flag.

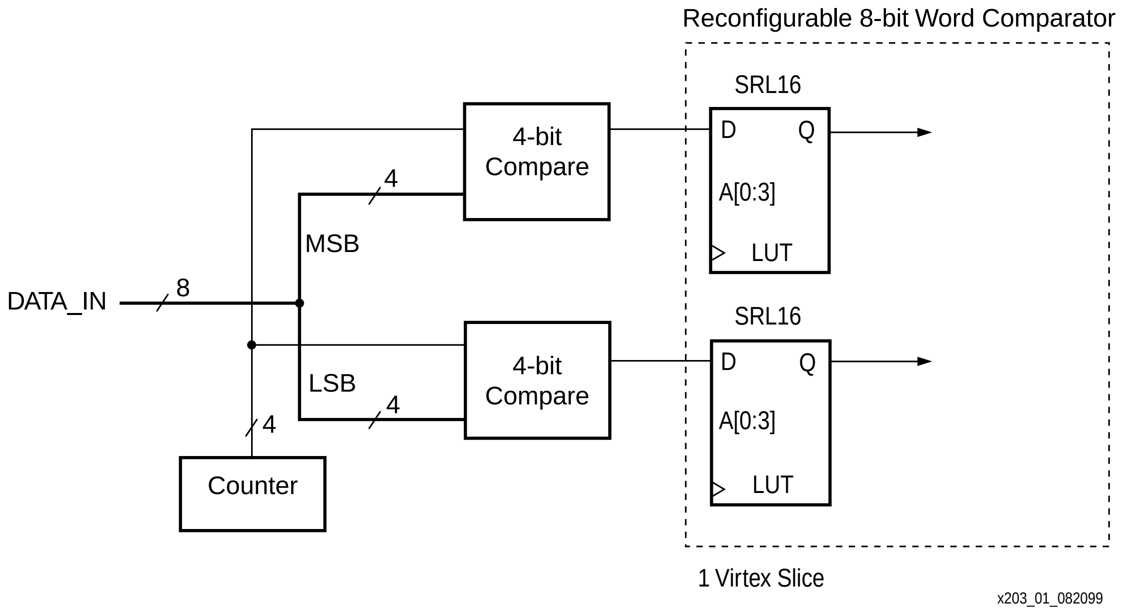

Figure 1: 8-bit CAM Word Write Operation (16 Clock Cycles)

Write Operation

The Shift Register mode is used to store new data in a location. 16 clock cycles shift the result of the comparison between the input data to be written and a 4-bit down counter (16 states). If the counter value equals the input data value then a "1" is shifted into the SRL16E primitive. Otherwise, a "0" is shifted in. The result is a 4-bit decoder in each LUT after the 16 clock cycles. Figure 1 is an 8-bit CAM word write operation.

Read Operation

The input data to be compared is used as an address of the Shift Register. Only one out of the 16 locations in the SRL16E has a "1" corresponding to the data stored previously. If the input data addresses this location, a match is found. The carry-chain will propagate this "1" (wide AND configuration) and if all the SRL16E in a particular slice column output a "1", a match is found. The global output of the carry-chain is one line of the CAM decoded address.

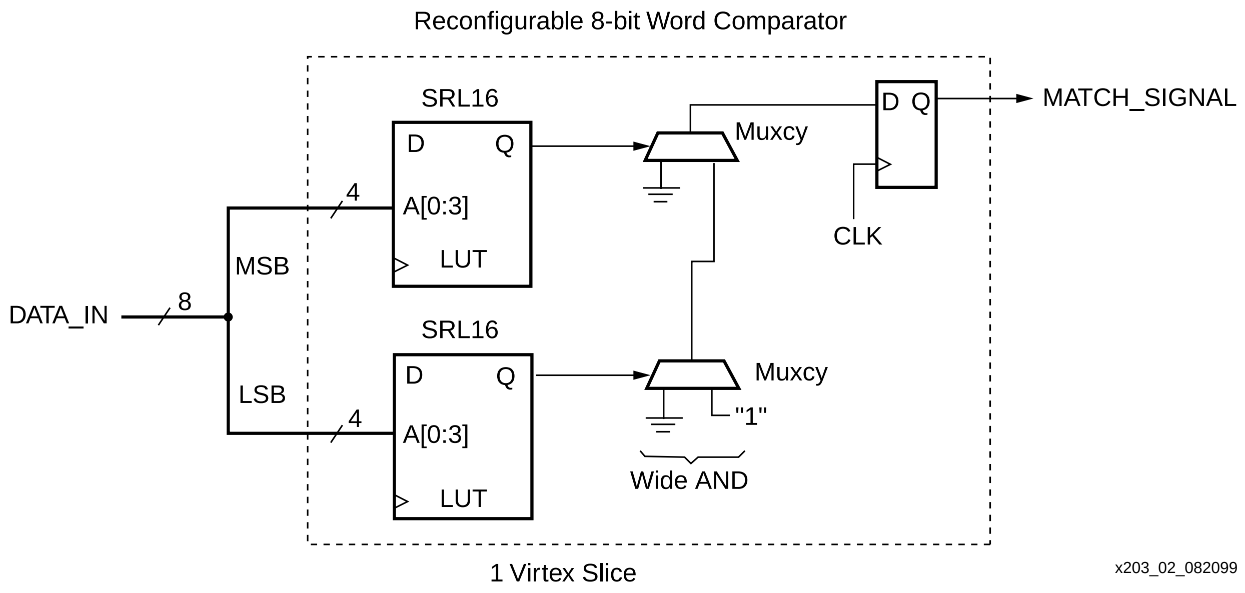

Figure 2: 8-bit CAM Word Read Operation (One Clock Cycle)

Designing a CAM in Virtex Family SRL16E Primitives

When using this application note, along with the reference design file XAPP203.zip, the designer will be able to produce fast, flexible CAMs using Virtex or Virtex-E slices. The first example of a CAM design in Virtex Family devices is illustrated in the hierarchical HDL reference desig. See Appendix A page 11.

Features:

- High performance one read clock cycle or match access time.
- 16 write clock cycles.
- Generic “word_width” from four bits up to any multiple by four bit value, only limited by the number of slices in a column in a particular Virtex Family device. As an example, a XCV50 has 16 slices per column or 32 LUT allowing implementation up to a 128-bit word. A XCV300 has 32 slices per column or 64 LUT, implementing up to a 256-bit word.
- Generic “nb_cam_16words” defining the number of basic 16 word blocks. The CAM depth is a multiple by 16 word value. The smallest required Virtex Family device depends on both the word width and the depth. Each LUT implements a four bit basic block CAM.
- Generic “addr_width” defining the number of address lines directly tied to the “nb_cam_16words”. Some examples would be a 32-word CAM requiring five address lines, a 64-word CAM requiring six address lines, or a 128-word CAM requiring seven address lines and onward.

Basic building block: CAM_SRL16

The fast match CAM implementation is composed of a shift register SRL16E to store and compare four CAM bits and the associated carry-chain MUXCY, used in building a wide AND gate by cascading this basic block.

The generic word width is a multiple of four (each LUT contains 4 bits). A word width of 8 bits requires one Virtex slice (2 x SRL16E and 2 x MUXCY), a word width of 16 requires two Virtex slices, 32 requires four, 64 requires eight, ... and so on.

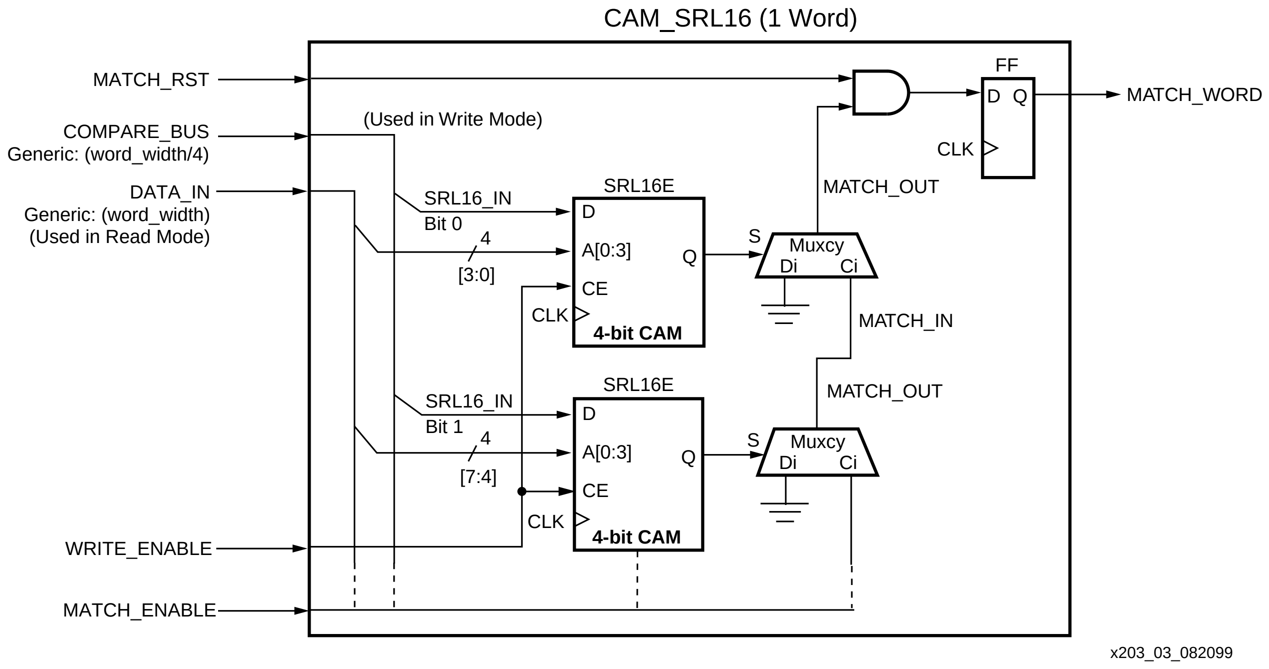

Figure 3: 8-bit Word (CAM_SRL16) in One Virtex Slice (Cascadable)

One main advantage of this approach is that the control logic is independent of the word width and the design hierarchy.

A large word width (128 or more bits) has a low impact on performance because of the high performance Virtex carry-chain. The wide AND built on MUXCY is the only path which is variable with the word width. (Each four bits adds only a MUXCY delay).

Module: CAM_16WORDS

Single CAM words are grouped in one 16 word module. A simple 4:16 decoder provides the WRITE_ENABLE of the selected CAM word in WRITE mode. Another module, COMPARE_4 compares the input data to a four-bit counter and generates the COMPARE_BUS.

The COMPARE_BUS is shifted into this CAM word (connected to the D input of the SRL16E). The 16 clock cycles write operation shifts this input value in the SRL16E to the right position.

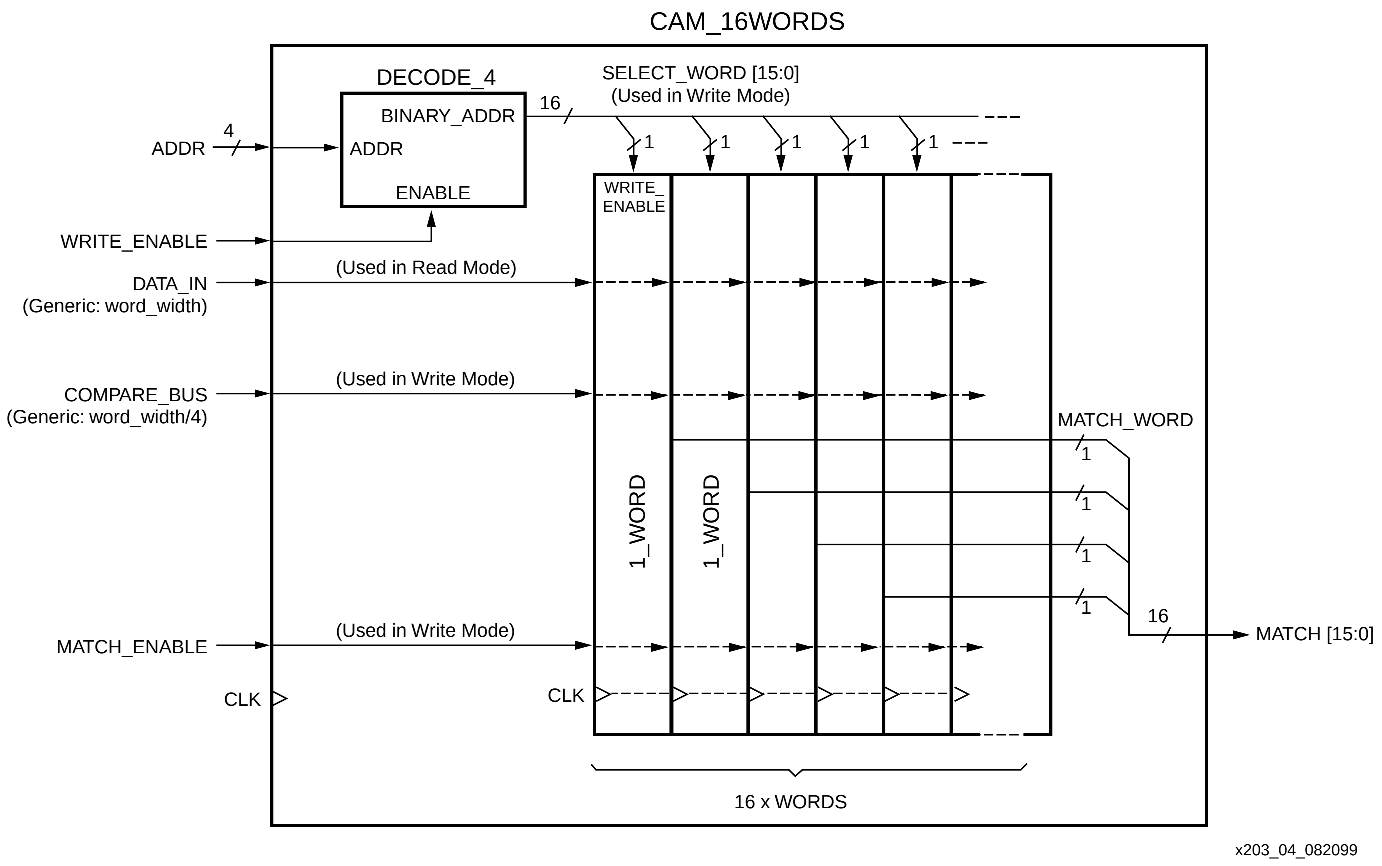

Figure 4: CAM_16WORDS Block Diagram

Read Cycle Mode:

In the read mode, the DATA_IN is compared in parallel to each single CAM word which generates a 16-bit MATCH bus. When using an 8-bit MATCH bus, for example, if the DATA_IN is found in WORD 2, the MATCH bus is "00000100" (decoded address). If no match is found the MATCH bus is "00000000". Some additional logic can be added to handle masks or multiple matches (A DATA_IN is found in more than one location).

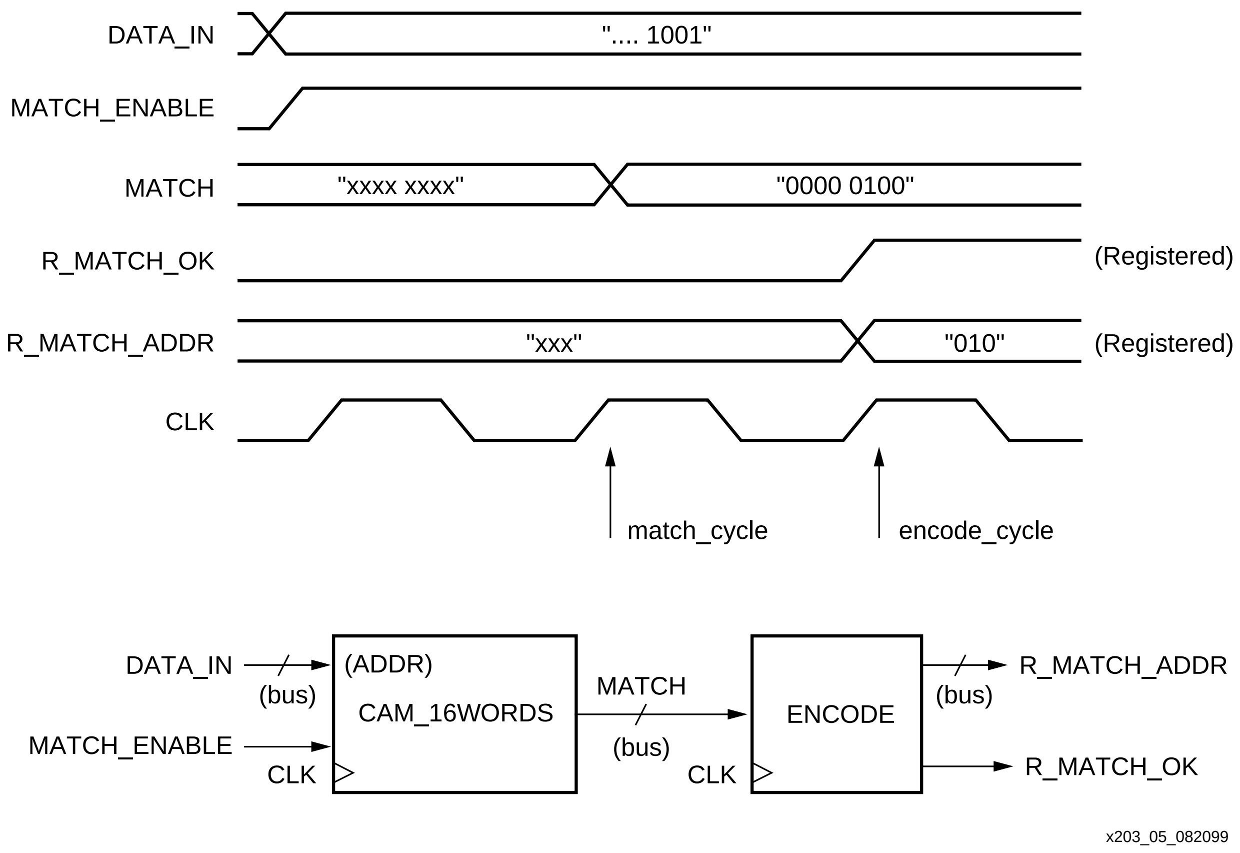

Figure 5: CAM Read Mode Waveforms

Module: CAM_generic_word in Read Mode

To build the desired CAM size, several CAM_16WORDS modules are instantiated in the CAM_generic_word module. The CAM depth is a generic value "nb_cam_16words" and should be a multiple by 16 words. A single CAM_16WORDS requires a 4-bit MATCH address output bus. The generic value "addr_width" corresponds to the number of CAM_16WORDS or the CAM depth. A 32-word CAM requires five address lines, a 64-word CAM requires six address lines, ...

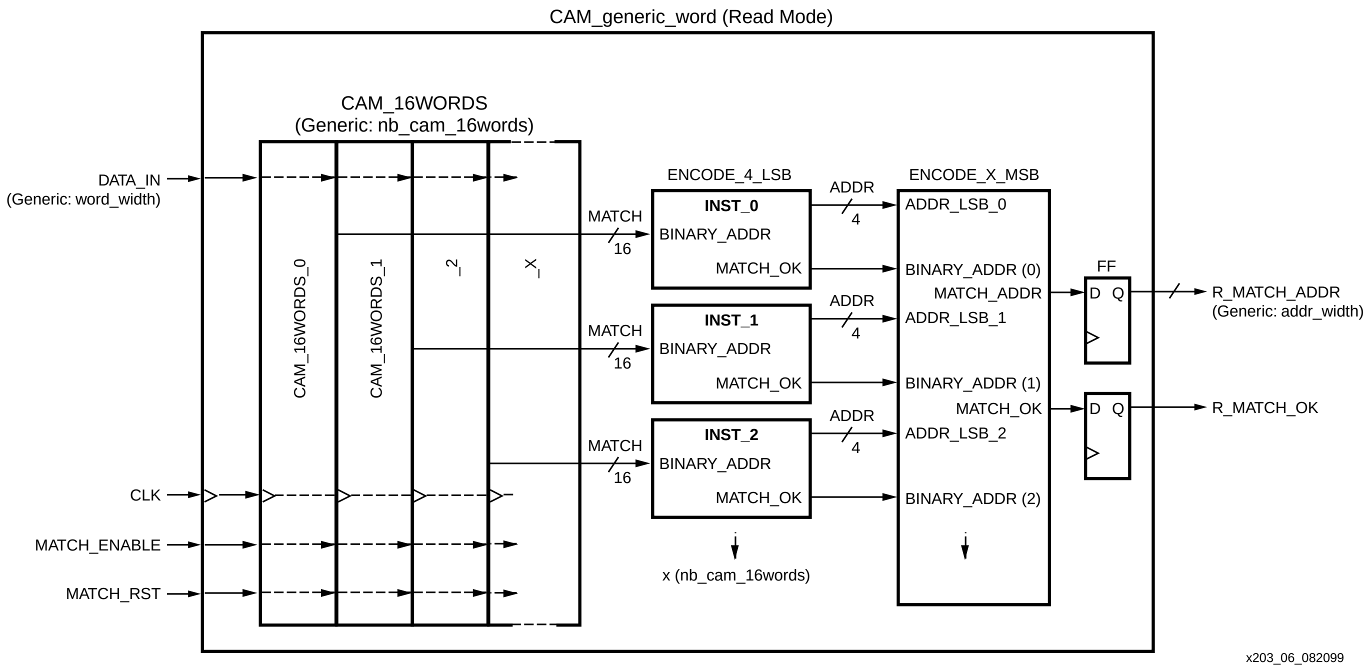

Figure 6: CAM Read Path (1 Clock Cycle)

The MATCH busses (CAM_16WORDS outputs) are encoded to generate both the match address output MATCH_ADDR and a MATCH_OK signal (High when a match is found).

Write Cycle:

A WRITE_ENABLE signal starts a 4-bit counter ("1111" down to "0000") and the write enable signal WRITE_SRL16 remains asserted for 16 clock cycles. Each counter value is compared to each 4-bit DATA_IN bus (data to be written) in parallel. The COMPARE output is connected to each D input of the SRL16E array.

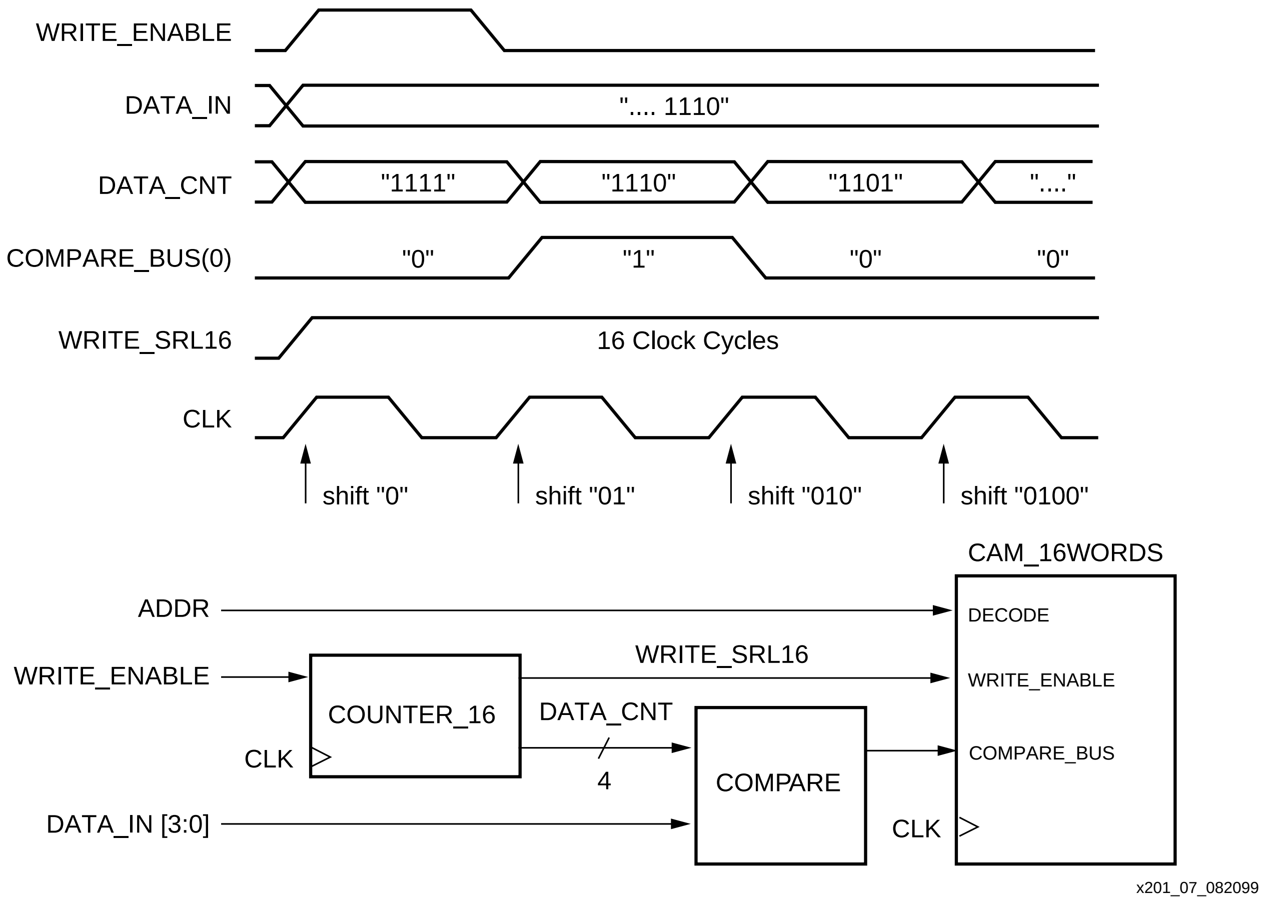

Figure 7: CAM Write Mode Waveforms

When a particular DATA_IN (4-bit) equals the counter value, a "1" is shifted, otherwise a "0" is shifted in the SRL16E of the selected CAM word. A standard address decoder selects the CAM word to be written (like in a standard RAM memory). If the 4-bit DATA_IN value is "1110", then a “0” is shifted in at the first clock cycle shift, a "1" is shifted in at the second clock cycle, a "0" is shifted in at the remaining 14 clock cycles.

The resulting SRL16E content is "0100000000000000". In the Read mode, when the DATA_IN bus is connected to A0:A3 inputs of this SRL16E, only the pattern A(3:0) = “1110” will yield a “1” (match). In the Read mode, the SRL16E output is asynchronous, equivalent to that of a LUT.

Module CAM_generic_word (Write Mode)

The write mode part of this module instantiates a single counter and a single comparator (COMPARE_4 x "word_width/4") for the complete design. A standard top level address decoder generates the select bus WRITE_ENABLE of each CAM_16WORDS.

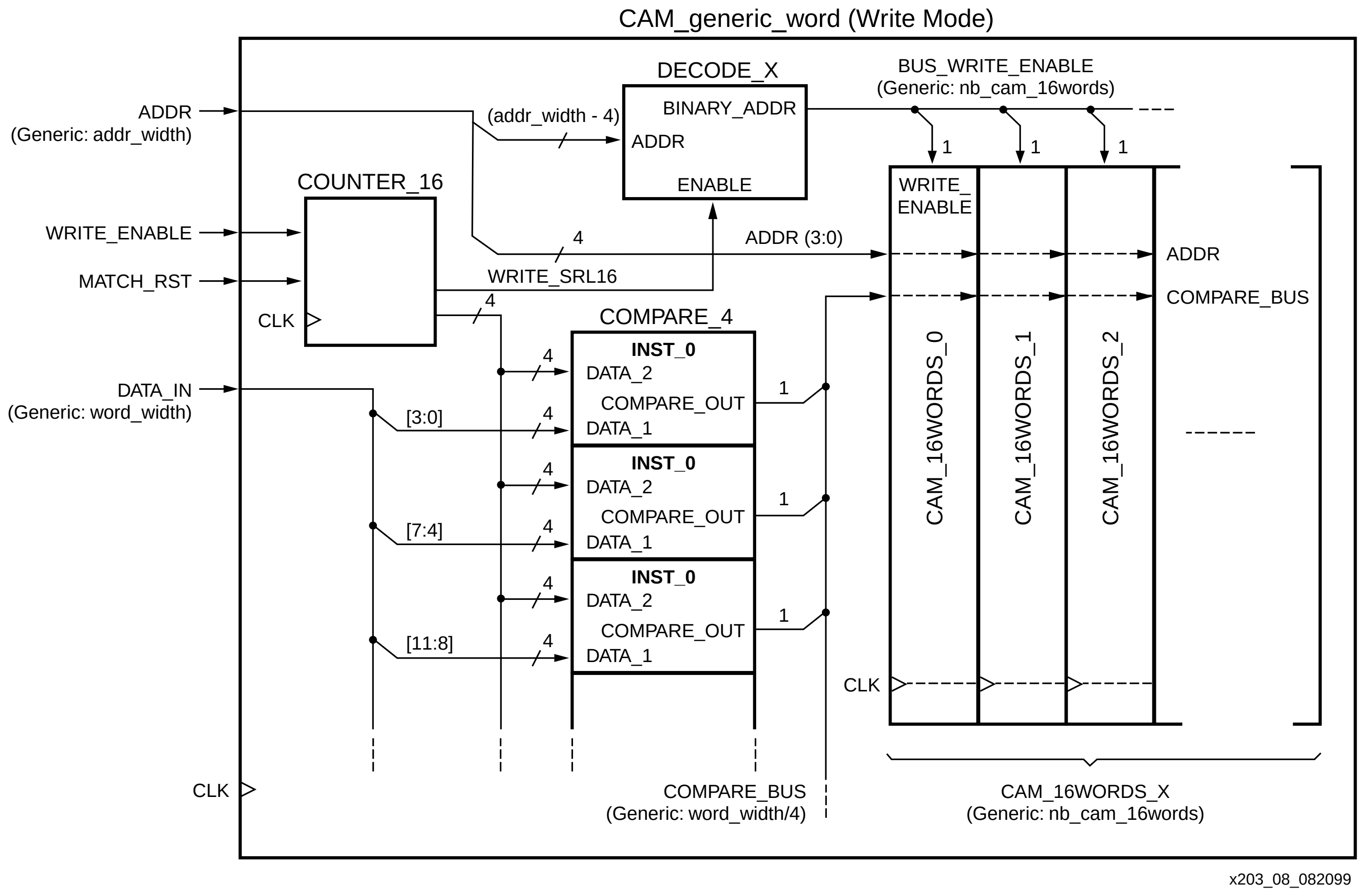

Figure 8: CAM Write Path (16 Clock Cycles to Count from 15 Down to 0)

Module: CAM_top

This module is a top level wrapper of the generic CAM. It registers all input and output signals.

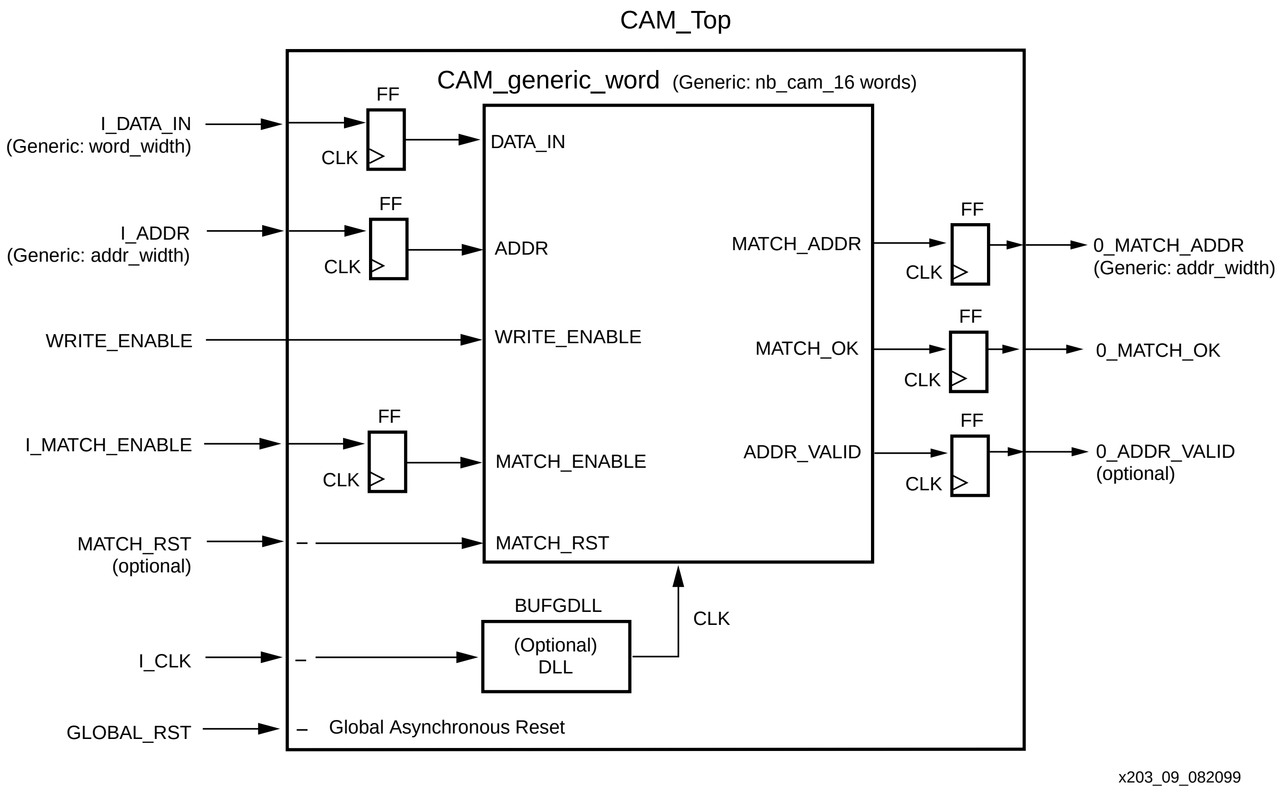

Figure 9: CAM Design Top Level Block Diagram

Pinout: (all I_XXX signals are inputs, O_XXX are outputs)

- I_DATA_IN is the generic data bus used by both read and write operations
- I_ADDR is the generic input address bus used to write only a new data into a selected location.
- WRITE_ENABLE is a one clock cycle signal to enable the 16 clock cycles write operation (active High).
- I_MATCH_ENABLE enables a read access (active High).
- MATCH_RST is a synchronous reset. It does not change the CAM content. (optional)
- I_CLK is the clock. As an option, It can be routed through a DLL.
- O_MATCH_ADDR is the generic output address valid only in read mode.
- O_MATCH_OK is High when a match is found (read operation).
- O_ADDR_VALID is an optional signal when multiple matches can occur. Active High if a single match occurred.

A recommendation when synthesizing this reference design is to keep the hierarchy for eventual floor planning and to facilitate static timing analysis. A constraint file (UCF) should define the clock period. Because of the high fanout nets, the UCF file also constrains the following nets. The DATA_IN bus and the internal output of the "COMPARE_4" module of the COMPARE_BUS are two busses where each line has a fanout equivalent to the CAM depth or more. For optimal performance, simple UCF constraints similar to the following example automatically constrain each line.

```
NET "DATA_IN<*>" MAXDELAY = 3ns;
```

```
NET "CAM_generic_word_1/COMPARE_BUS<*>"
MAXDELAY = 3ns
```

The reference design adjoining this application note can easily be adapted into different CAM modes.

Conclusion

Virtex and Virtex-E SRL16E based implementations are convenient for small to large word width with a minor impact on the performances. Each additional four-bit width adds one carry-chain MUXCY delay. The design structure remains identical. The only limitation to the word width, if any, is the number of slices per column of the Virtex Family device. However eight bits per slice allows very wide words. The Virtex XCV50 and the XCV50E have 16 slices per column, the XCV1000 and the XCV1000E have 64 slices per column, and the XCV3200E has 104 slices per column.

The CAM depth is also expandable (number of slices column) providing a real flexibility. All the decoded addresses or match bus are generated in a single clock cycle independently from the CAM size. The reference design proposes an address encoder and match flag generation in gates or in 3-state buffers. These options are open to the designer according to the final CAM implementation.

Because there are various application needs, there are diverse approaches to CAM solutions. The unique Virtex Family features are key advantages to providing these flexible solutions. In addition to the design presented in this application note, XAPP202 "Content Addressable Memory (CAM) in ATM Applications" and XAPP204 “Using Block SelectRAM+ for High-Performance Read/Write CAMs” offer complementary solutions.

APPENDIX A: Synthesizable HDL code Reference Design

This appendix describes a hierarchical, synthesizable design implementing a parametric word width and memory depth, CAM in Virtex slices. The complete HDL code is available as a reference design (File: xapp203.zip or xapp203.tar.z).

The header of each VHDL module is listed below:

Module: CAM_top.vhd

```
--
-- Module:       CAM_Top / Top Level
-- Design:       CAM_Top
-- VHDL code:    Hierarchical wrapper
--                            Instantiated CAM_generic_word (variable depth and word width)
--
-- Synthesis     Synopsys FPGA Express ver. 3.2 - Option = Preserve Hierarchy
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Instantiated a CAM implementation
--               Registered inputs and outputs (CAM internal timing analysis)
--
-- Device:       Virtex Families (Virtex and Virtex-E)
--
-- Created by: Jean-Louis BRELET / XILINX - Virtex Applications
-- Date: July 23, 1999
-- Version: 1.0
--
-- History:
--     1.
--
-- Disclaimer:   THESE DESIGNS ARE PROVIDED "AS IS" WITH NO WARRANTY
--               WHATSOEVER AND XILINX SPECIFICALLY DISCLAIMS ANY
--               IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR
--               A PARTICULAR PURPOSE, OR AGAINST INFRINGEMENT.
--
--  Copyright (c) 1999 Xilinx, Inc.  All rights reserved.
-------------------------------------------------------------------------------------------
```

Module: CAM_generic_word.vhd

```
--
-- Module:       CAM_generic_word
-- Design:       CAM_Top
-- VHDL code:    Hierarchical RTL
--               Instantiated COUNT_16
--               Instantiated COMPARE_4
--               Instantiated CAM_16WORDS
--                    Instantiated DECODE_4
--                    Instantiated CAM_SRL16
--                           Instantiated INIT_SRL16_AND
--                    Instantiated ENCODE_4_LSB
--                    Instantiated DECODE_X
--                    Instantiated ENCODE_X_MSB
--               Choose the right ENCODE and DECODE modules according to the number of
CAM_16WORDS
--               If "nb_cam_16words" = 2 then ENCODE_1_MSB and DECODE_1 must be used:
1 bit to decode 2 CAM_16WORDS

--               If "nb_cam_16words" = 4 then ENCODE_2_MSB and DECODE_2 must be used:
```

```
2 bits to decode 4 CAM_16WORDS
--                If "nb_cam_16words" = 8 then ENCODE_3_MSB and DECODE_3 must be used:
3 bits to decode 8 CAM_16WORDS
--                If "nb_cam_16words" = 16 then ENCODE_4_MSB and DECODE_4 must be used:
4 bits to decode 16 CAM_16WORDS
--                Note: synthesis tools do not support Configuration
--
-- Synthesis      Synopsys FPGA Express ver. 3.2  - Option = Preserve Hierarchy
--                Use of "pragma synthesis_off/on" and attributes
--
-- Description:   Instantiated "nb_cam_16words" CAM_16WORDS (see generic)
--                2 x 16 words depth x 8 bits width by default
--                1 clock cycle Read (or Match),
--                16 clock cycles Write
--                MATCH_OK indicates one or more matches is/are found.
--                MATCH_ADDR output the address of the match, if ONLY ONE is found
--                ADDR_VALID indicates when MATCH_ADDR is a valid address (Optional)
--
-- Device:        Virtex Families
--                modules CAM_16WORDS fits in 1 Virtex slices column
--                (4 CAM bits per LUT + global control logic)
--                If "nb_cam_16words" = 2 then  CAM32xWORD
--                If "nb_cam_16words" = 4 then  CAM64xWORD
--                If "nb_cam_16words" = 8 then  CAM128xWORD
--                If "nb_cam_16words" = 16 then  CAM256xWORD
-- ...
-----------------------------------------------------------------------------------------
```

Module: COUNT_16.vhd

```
--
-- Module:        COUNT_16
-- Design:        CAM_Top
-- VHDL code:     RTL
--
-- Synthesis      Synopsys FPGA Express ver. 3.2
--                Use of "pragma synthesis_off/on" and attributes
--
-- Description:   4 bits counter 15 downto 0
--                Generate a 16 clock cycle wide enable signal
-- Device:        Virtex Families
-- ...
-----------------------------------------------------------------------------------------
```

Module COMPARE_4.vhd:

```
--
-- Module:       COMPARE_4
-- Design:       CAM_Top
-- VHDL code:    Virtex primitives' instantiation
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Basic building block of a CAM to compare 2 x 4 bits busses
--               Combinatorial module
--
-- Device:       Virtex Families
--               Gates and MUXF5 fitting in 1 Logic Cell.
-- ...
-----------------------------------------------------------------------------------------
```

Module: CAM_16WORDS.vhd

```
--
-- Module:       CAM_16WORDS
-- Design:       CAM_Top
-- VHDL code:    Hierarchical RTL
--                     Instantiated DECODE_4
--                     Instantiated CAM_SRL16
--                             Instantiated INIT_SRL16_AND
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Building block of a CAM 16 words (variable word width)
--               1 word depth x "word_width" bits width
--               1 clock cycle Read (or Match), 16 clock cycles Write
--               If NO match is found, or MATCH_ENABLE is low, the output MATCH =
'0000000000000000'
--               Initialized SRL16E in low level module
--
-- Device:       Virtex Families
-- ...
-----------------------------------------------------------------------------------------
```

Module: CAM_SRL16.vhd

```
--
-- Module:       CAM_SRL16
-- Design:       CAM_Top
-- VHDL code:    Hierarchical RTL
--                    Instantiated INIT_SRL16_AND
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Basic building block of a CAM using 16-Bit Shift Register LUT
--               1 word depth x "word_width" bits width
--               1 clock cycle Read (or Match), 16 clock cycles Write
--               If NO match is found, or MATCH_ENABLE is low, the output MATCH_WORD = '0'
--               Initialized SRL16E in low level module
--
-- Device:       Virtex Families
-- ...
------------------------------------------------------------------------------------------
```

Module: INIT_SRL16_AND.vhd

```
--
-- Module:       INIT_SRL16_AND
-- Design:       CAM_Top
-- VHDL code:    Virtex primitives' instantiation
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Basic building block of a CAM using 16-Bit Shift Register LUT
--               4 bits per LUT
--               Asynchronous Read (or Match) by ADDR -> MATCH_OUT
--               Initialized SRL16E: attributes to constraint PAR and simulation
--
-- Device:       Virtex Families
--               SRL16E and MUXCY
-- ...
------------------------------------------------------------------------------------------
```

Module: ENCODE_4_LSB.vhd

```
--
-- Module:       ENCODE_4_LSB
-- Design:       CAM_Top
-- VHDL code:    RTL / Combinatorial
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Encode a 16 bits binary address into 4 bits and find if a match occurs
--               if BINARY_ADDR = "0000000000100000" => MATCH_ADDR = "0101" / MATCH_OK = 1
--               Optional ADDR_VALID = 1 when only one Match (If simultaneous matches can
occur)
--               However, the ADDR_VALID generation double the size of the combinatorial
logic !
--               if no match found => MATCH_OK = 0 / ADDR_VALID = 0 (MATCH_ADDR is not a
valid address)
--               if 2 or more matches found => MATCH_OK = 1 / ADDR_VALID = 0 (MATCH_ADDR is
not valid address)
--
-- Device:       Virtex Families
-- ...
------------------------------------------------------------------------------------------
```

Module: ENCODE_4_MSB.vhd

This module is available in one bit ENCODE_1_MSB (CAM 32 words), two bits ENCODE_2_MSB (CAM 64 words), three bits ENCODE_3_MSB (CAM 128 words) and four bits ENCODE_4_MSB (CAM 256 words). Additional modules for other CAM depths could be easily created following this model.

```
--
-- Module:       ENCODE_4_MSB
-- Design:       CAM_Top
-- VHDL code:    RTL / Combinatorial
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Encode a 16 bits binary address into 4 bits, map with the LSB address and
find if a match occurs
--               if BINARY_ADDR = "0000000000100000" => MATCH_ADDR = "0101" / MATCH_OK = 1
--               Optional ADDR_VALID = 1 when only one Match (If simultaneous matches can
occur)
--               However, the ADDR_VALID generation double the size of the combinatorial
logic !
--               if no match found => MATCH_OK = 0 / ADDR_VALID = 0 (MATCH_ADDR is not a
valid address)
--               if 2 or more matches found => MATCH_OK = 1 / ADDR_VALID = 0 (MATCH_ADDR is
not valid address)
--
--               Choice between GATES ONLY implementation or BUFT implementation. (See
comments)
--               Note: synthesis tools do not support Configuration
--
-- Device:       Virtex Families
-- ...
------------------------------------------------------------------------------------------
```

Module: DECODE_4.vhd

This module is available in one bit DECODE_1 (CAM 32 words), two bits DECODE_2 (CAM 64 words), three bits DECODE_3 (CAM 128 words) and four bits DECODE_4 (CAM 256 words). Additional modules for other CAM depths could be easily created following this model.

```
--
-- Module:       DECODE_4
-- Design:       CAM_Top
-- VHDL code:    RTL / Combinatorial
--
-- Synthesis     Synopsys FPGA Express ver. 3.2
--               Use of "pragma synthesis_off/on" and attributes
--
-- Description:  Decode 4 bits address into 16 binary bits
--               Generate an ENABLE bus
--
-- Device:       Virtex Families
-- ...
-----------------------------------------------------------------------------------------
```

Similar modules ENCODE_3_MSB, ENCODE_2_MSB, ENCODE_1_MSB, DECODE_4, DECODE_3, DECODE_2, and DECODE_1 are available in the reference design files.

```
End of Appendix A.
```

Revision History

Date	Revision	Activity
8/31/99	1.0	Initial Release
9/23/99	1.1	Initial Virtex-E update

The Programmable Logic Company℠

Headquarters

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
U.S.A.

Tel: 1 (800) 255-7778
or 1 (408) 559-7778
Fax: 1 (408) 559-7114

Net: hotline@xilinx.com
Web: http://www.xilinx.com

North America

Irvine, California
Tel: (949) 727-0780

Englewood, Colorado
Tel: (303) 220-7541

Sunnyvale, California
Tel: (408) 245-9850

Schaumburg, Illinois
Tel: (847) 605-1972

Nashua, New Hampshire
Tel: (603) 891-1098

Raleigh, North Carolina
Tel: (919) 846-3922

West Chester, Pennsylvania
Tel: (610) 430-3300

Dallas, Texas
Tel: (972) 960-1043

Europe

Xilinx Sarl
Jouy en Josas, France
Tel: (33) 1-34-63-01-01
Net: frhelp@xilinx.com

Xilinx GmbH
München, Germany
Tel: (49) 89-93088-0
Net: dlhelp@xilinx.com

Xilinx, Ltd.
Byfleet, United Kingdom
Tel: (44) 1-932-349401
Net: ukhelp@xilinx.com

Japan

Xilinx, K.K.
Tokyo, Japan
Tel: (81) 3-3297-9191
Net: jhotline@xilinx.com

Asia Pacific

Xilinx Asia Pacific
Hong Kong
Tel: (852) 2424-5200
Net: hongkong@xilinx.com