LogiCORE™

One Dimensional RAM-Based Correlator

February 8, 1998 | Product Specification

XILINX®

Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
E-mail: dsp@xilinx.com
URL: www.xilinx.com

Features

- Data can be loaded in either serial or parallel (broadside) fashion
- Supports data words from 4 to 128 bits in steps of 4
- Cascadable to any length
- Easily extendible to 2 dimensions
- High performance implementation utilizing efficient RAM design
- Requires relatively few CLBs
- Single level of logic per pipeline stage
- Drop-in modules for the XC4000E, EX, and XL families
- Density and performance guaranteed through Relationally Placed Macro (RPM) mapping and placement technology
- Available in Xilinx CORE Generator

Functional Description

This parameterized core accepts data in a serial or parallel fashion and looks for a predetermined bit pattern that is stored in RAM. The output is a binary representation of the number of bits that match the desired bit pattern.

The match register, the mask register, and all of the logic to implement the compare is implemented in look-up table (LUT) logic so the function can be realized as a series of small LUTs (one for each 4 bits of data) followed by a small adder tree.

Unlike the ROM-based correlator, the match pattern is not specified in the GUI. The RAMs must be explicity loaded before the correlator can be used. The WE signal controls when data is written into the RAM: when WE is high, the DIN input (for parallel) or the last **W** input bits of SDIN (for serial, where **W** is the data width) act as the address to the RAM. Each 4 bits of input address a 16x3 RAM. The data to be stored in the RAM must come in at the LDATA bus. The width of LDATA is 3/4 the width of the correlator -- 4 bit slices of DIN are used as an address to store the corresponding 3 bit slices of LDATA into individual 16x3 RAMs. It takes 16 clock cycles to fully load the RAM after which WE must go low. After that, the correlator functions like a ROM correlator. The RAM can be reloaded with a new pattern at any time by making WE high and going through 16 clock cycles with the appropriate inputs at DIN and LDATA.

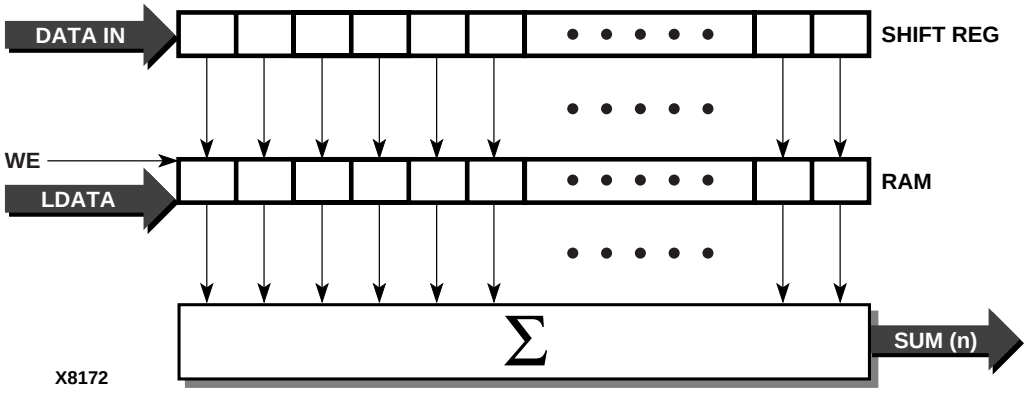

Figure 1: Serial In Correlator Block Diagram

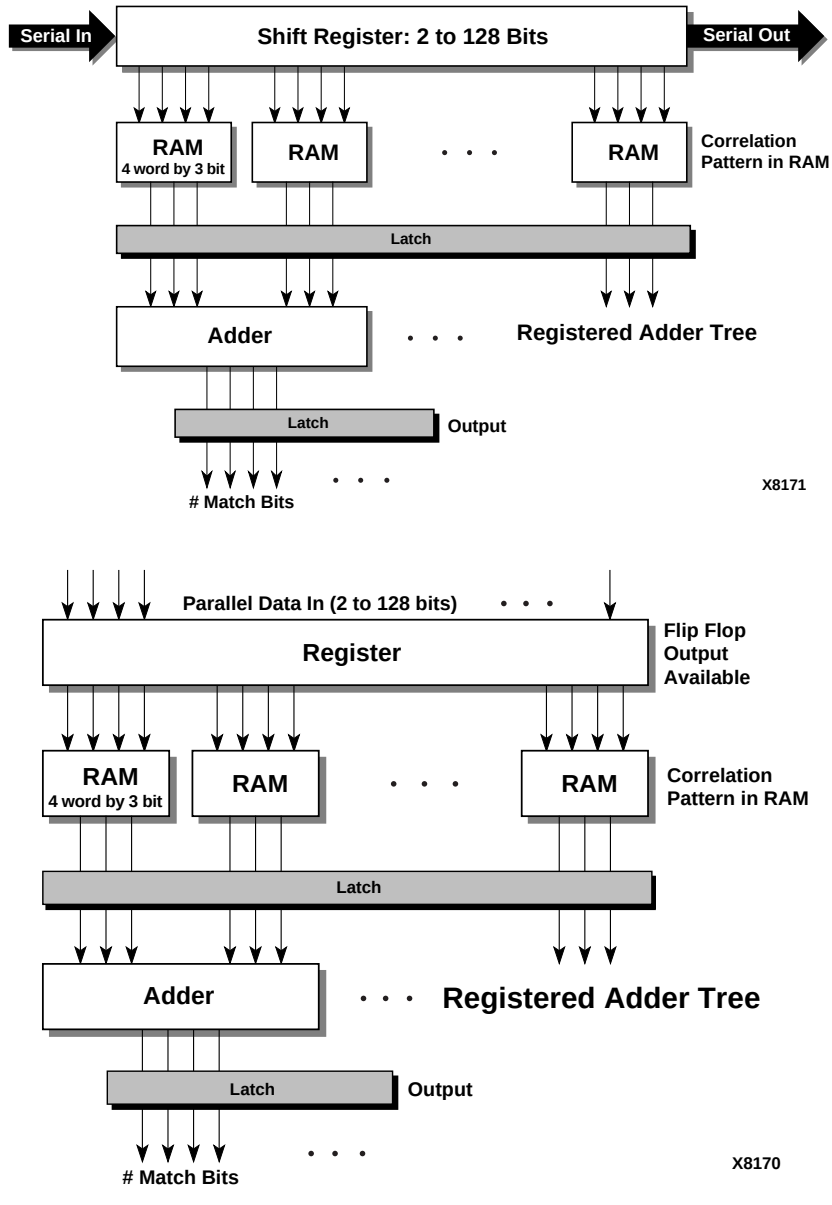

Figure 2: CORE Generator Implementation for Serial Data (top) and Parallel Data (bottom)

Latency is equal to one input buffer plus one pipeline register for each level of the 4-bit adder tree. For example, if the match register is 24 bits, the adder tree is 3 levels deep with a 5-bit wide output. The total latency is two (registers) + three (adder trees) = five.

This function is used in data communications to establish synchronization in a serial bit stream, and in any application that requires pattern recognition.

Using RAM for Implementing Correlators

Table 1 illustrates what the user must calculate to produce the contents of the distributed RAM. For a 4-bit correlator with an input search pattern of 1101 and a mask pattern of 1111 (include all bits), the RAM contains the following data.

Any n-stage correlator can be decomposed into (n/4) 4-stage correlators. The RAMs contain all potential outputs for each 4-stage correlation. For example, the correlation pattern 1101 stores 4 at address D in the RAM (all four bits match) and 3 at addresses 5, 9, C, and F in the RAM (three bits match).

The user must compute the contents of the RAMs for the desired “match” pattern and load the RAMs before doing any comparisons.

Table 1: RAM Contents

Address	Data
0000	001 (1 bit matches)
0001	010 (2 bits match)
0010	000 (0 bits match)
0011	001 (1 bit matches)
0100	010 (2 bits match)
0101	011 (3 bits match)
0110	001 (1 bit matches)
0111	010 (2 bits match)
1000	010 (2 bits match)
1001	011 (3 bits match)
1010	001 (1 bit matches)
1011	010 (2 bits match)
1100	011 (3 bits match)
1101	100 (4 bits match)
1110	010 (2 bits match)
1111	011 (3 bits match)

Combining 4-Input Sections

Each four-input correlator section can be combined by summing the outputs with an adder tree. The RAMs are 16 words (4 address lines) by 3 bits wide each. The adder tree grows by one bit for each level and the resulting output is a binary number representing the number of matches in the input data word after the mask register has been applied.

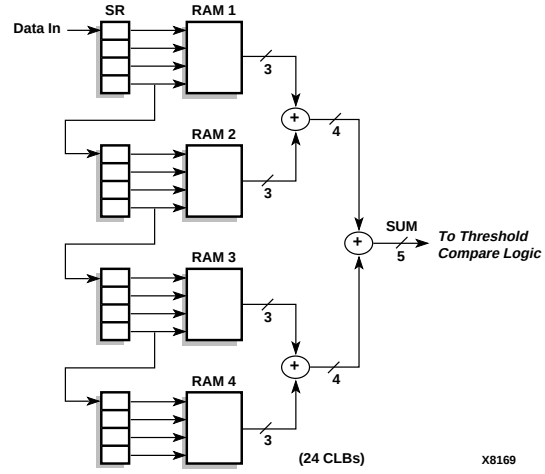

Figure 3: Combining 4-Input Sections

Pinout

Signal names for the schematic symbols are shown in Figures 4 and 5, and described in Tables 2 and 3.

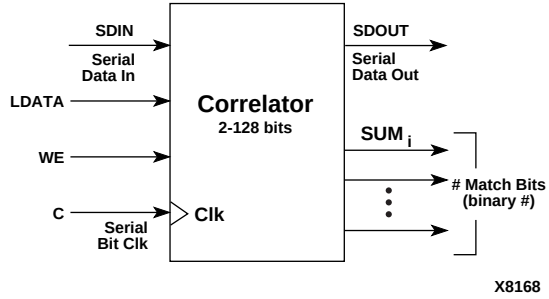

Figure 4: Core Schematic Symbol – Serial Data In

Table 2: Core Signal Pinout – Serial Data In

Signal	Signal Direction	Description
SDIN	Input	Serial Data In – Serial data stream for correlation.
C	Input	Serial Data Clock
WE	Input	Write Enable – to load the RAM.
LDATA	Input	The data that gets written into all the RAMs at the address locations in the input shift register.

Table 2: Core Signal Pinout – Serial Data In (cont.)

Signal	Signal Direction	Description
SDOUT	Output	Serial Data Out – SDIN delayed N clocks.
SUM_i	Output	Binary representation of the number of bits that match the correlation pattern excluding any bits defined in the mask register.

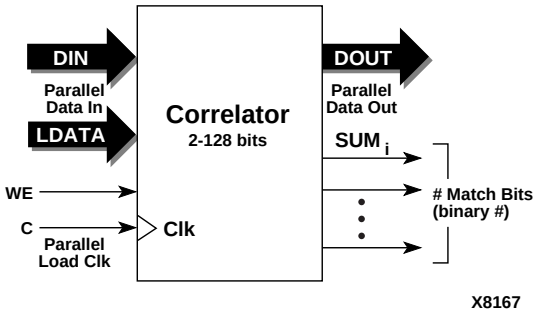

Figure 5: Core Schematic Symbol – Parallel Data In

Table 3: Core Signal Pinout – Parallel Data In

Signal	Signal Direction	Description
DIN	Input	Parallel Data In – 2 to 128 bits
C	Input	Clock – Parallel load clock
WE	Input	Write Enable – to load the RAM.
LDATA	Input	The data that gets written into all the RAMs at the address locations in the input shift register.
DOUT	Output	Parallel Data Out – DIN delayed 1 clock.
SUM_i	Output	Binary representation of the number of bits that match the correlation pattern excluding any bits defined in the mask register.

The CORE Generator accepts the parameters entered through the dialog box and creates the specific design from the values entered using a parameterized VHDL recipe. VHDL instantiation code and a schematic symbol are created along with the netlist for the design.

CORE Generator Parameters

This macro has two CORE Generator dialog boxes. The one for the parallel correlator is shown in Figure 6. The serial correlator dialog box has a different title, but the parameters are the same. The parameters are as follows:

- **Component Name**: Enter a name for the component.
- **Data Width**: Select an input width from the pull-down menu for the data string to be compared. The valid range is 4-128 in multiples of 4.

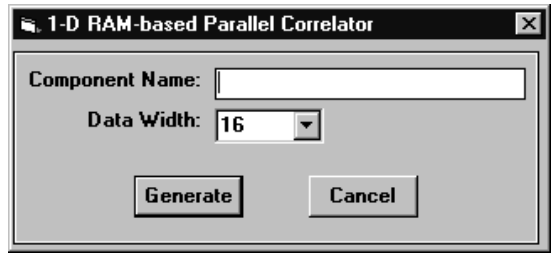

Figure 6: CORE Generator Dialog Box

Bit Width and CLB Count

Table 4 lists the number of CLBs required for example bit widths. The maximum speed is for XC4000E-1 devices.

Table 4: Bit Width versus CLB Count

Bit Width	CLB Count
4	4
8	14
12	20
16	25
20	36
32	54

Ordering Information

This macro comes free with the Xilinx CORE Generator. For additional information contact your local Xilinx sales representative, or e-mail requests to dsp@xilinx.com.