XAPP254 (v1.0) January 12, 2001

# The Virtex-II SiberBridge

Author: Ratima Kataria & the SiberCore Applications Engineering Group


## Summary

Designed to be implemented in a Virtex™-II FPGA, the Virtex-II SiberBridge is a register transfer logic (RTL) design example demonstrating a reference interface between a 32-bit host (typically a network processor) and the SiberCAM™ device, or a cascade of SiberCAM devices. The SiberCAM device is a large capacity content addressable memory (CAM) product of **SiberCore Technologies**. The SiberBridge provides a way to initiate searches, obtain search results, and perform table maintenance operations for the SiberCAM, all using a single 32-bit synchronous SRAM or a ZBT SRAM interface. The SiberBridge is intended as a reference design having a low-gate count.


## Introduction

### SiberCAM Device Overview

The SiberCAM device is a Content Addressable Memory (CAM) for use with ternary data of variable widths. A CAM is a storage array with the capability to search the array contents to find the location of a particular stored value. By comparing the input against the data memory, a CAM determines if an input value matches one or more values stored in the array.

Search data is presented to the SiberCAM device on the search data port. After several clock cycles, an address containing the data that best matches the data inside the SiberCAM device becomes available on the search address (results) port.The SiberCAM is a ternary CAM and therefore allows the storage of ternary data ("0", "1" or "don't care"). Ternary CAMs are frequently used in longest prefix match applications. This often results in more than one match during any one search operation. The result of the search is the longest prefix match.

A SiberCAM device is configured and loaded with ternary data during maintenance operations. Typically, maintenance operations are done using the separate 36-bit maintenance port. In 3-port mode, maintenance operations can be done in parallel with search operations that are initiated on the separate search data port. This allows for continuous search operations while maintenance operations are being performed.

For applications that do not require maintenance operations to be performed in parallel with search operations, the SiberCAM device can be used in 2-port mode. In this mode, the maintenance operations are performed with the same port used for search operations (Figure 1).

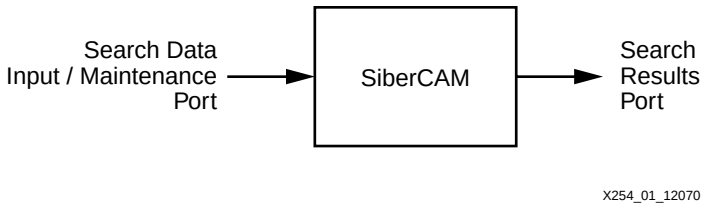


*Figure 1:* **SiberCAM Device in 2-Port Mode**

Whether using the separate maintenance port or the search data port in 2-port mode, the SiberCAM device expects maintenance operations to be performed in 18, 36, or 72-bit multiplexed quantities. The SiberBridge provides a mechanism to perform these operations using a 32-bit interface that resembles an SRAM.

## SiberBridge Description

The SiberBridge provides a host system with 32-bit access to the features of the SiberCAM device using a single SRAM-style interface. Figure 2 shows the interface signals between the SiberCAM device on one side of the SiberBridge and a network processor on the other.

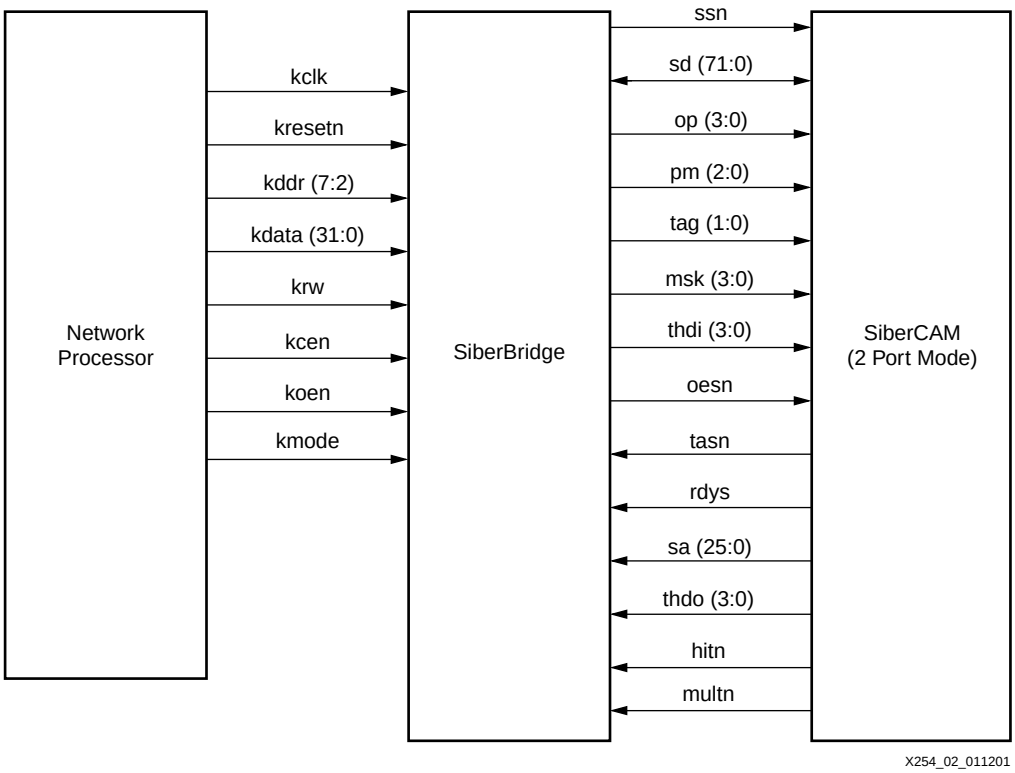


*Figure 2:* **Network Processor - SiberCAM Interface Using SiberBridge**

## One-Port Operation

Overall, the optimal performance of the SiberCAM device is achieved when the device is used in its native 3-port mode or 2-port mode. However, for some applications, it is desirable to perform maintenance operations, initiate search operations, and retrieve search results from a single 32-bit interface (e.g., when the SiberCAM device is used as a co-processor with a network processor). The SiberBridge is a reference design for an application where a SiberBridge permits a SiberCAM device or a cascade of SiberCAM devices to connect to a single 32-bit interface.

SiberCAM devices are generally used for fast look-ups of address information and classification in Internet routing equipment. Although a search operation in the SiberCAM device can be performed on every cycle, there is latency until the search result is available. If look-ups with the SiberCAM device are performed with a single processing thread, the processing of information is reduced due to the latency of the SiberCAM device. If multiple processing threads are performing lookups, it allows interleaving of search requests. Though each individual thread will still experience latency, multiple threads allow greater utilization of the SRAM bus.

The SiberBridge allows a host to use the SiberCAM device with multiple threads of operation. A thread can initiate a search with the SiberBridge and then poll for its result in a unique results register. The SiberBridge permits up to 32 threads of operation.

## RAM Interface

The SiberBridge interfaces to a host system using a 32-bit RAM interface. To the host system the RAM interface appears to be a synchronous SRAM or a ZBT SRAM. The SRAM mode is selected using a static input pin and simply changes the timing requirements for the write cycles.

To the host system, the SiberBridge appears to be a memory mapped device on the SRAM bus. Registers inside the RAM can be read or written to by the host system. The memory inside the SiberBridge is accessed using opcodes as described in the SiberCAM device documentation. The SiberBridge does not map all the memory addresses inside the SiberCAM device to the host systems memory map.

Since the SiberBridge has an SRAM interface, it does not cause wait states to occur nor does it assert a retry signal while it is busy performing an operation. Therefore, it is necessary to pole the state of the SiberBridge before performing certain operations. The state of the SiberBridge can be determined by reading the STATUS register.

## How the SiberBridge Works

The SiberBridge enables maintenance and search operations to be initiated using 32-bit registers and to organize search results for up to 32 different contexts.

The SiberBridge contains a large register that can be written to in 32-bit quantities. The contents of this register is written to the SiberCAM device in 72-bit quantities and initiated by writing to the GO register. For example, consider a maintenance write operation where the opcode, address, and data are all written to the write data register. After this register has been loaded, the data is written to the GO register as expected. Search operations can be initiated in a similar manner.

For maintenance operations that get data from the SiberCAM device, the SiberBridge captures this information and presents it to the SiberCAM device in 72-bit quantities, in a series of 32-bit registers. These registers can then be read from the SiberBridge.

The SiberBridge does not decode the data written to the write data register and, therefore, knows nothing about the opcodes it is supplying to the SiberCAM device. This requires the designer to identify which maintenance operations return read data. This puts a slightly greater burden on the software, but significantly reduces the complexity of the SiberBridge. The block diagram in Figure 4 illustrates the registers that write data and capture data from the SiberCAM device.

For one-port operation, the SiberBridge stores the results of searches in internal result registers. This allows multiple contexts to initiate searches and obtain search results in any order.

Multiple search contexts are supported as follows: the user writes the context number in the GO register at the time it initiates a search operation. This context number "follows" the search as it propagates through the SiberCAM device. When the search result becomes available, the result is written into the context result register associated with the selected context. There is one context result register for each of the 32 contexts.

A status bit is contained in each context result register. The status bit is cleared when the context is written into the GO register initiating a search operation. When the context result register is updated, the status bit is set. Figure 4 illustrates the flow of information in the SiberBridge implements the one-port mechanism.

## Semaphore Register

The SiberBridge is a simple mechanism to allow a network processor to perform searches using different threads of operation. It is designed to be implemented in relatively few gates. In order to keep the design simple, different threads of operation must share the registers of the SiberBridge. Ownership of the registers is determined by the semaphore register.

For an application with multiple threads of operations, the SiberBridge is used as follows:

- Initially, the semaphore register has a value of "0."
- Each thread of operation using the registers of the SiberBridge does the following:
  - A thread writes to the semaphore register a unique non-zero value. The thread reads back the semaphore register. If the value of the semaphore register is the value that was written, then the thread has ownership of the registers. If the value read back from the semaphore register is some value other than the value that was written, then the thread continues to write to and read from the semaphore register until it contains the value written.
  - Once ownership of the registers is obtained, the thread programs the WDATA register and writes to the GO register with a unique context value. Then the thread writes a value of zero to the semaphore register. After freeing the semaphore register, the thread reads from the result registers associated with the value written to the context register. The value in the result register is valid if, and only if, bit 31 of the register is a value of "1."

## Using Digital Clock Managers (DCMs) in the Reference Design

The Virtex II Digital Clock Manager (DCM) provides a complete on-chip and off-chip clock(s) generator, offering a wide range of powerful clock management features.

The DCM utilizes fully digital delay lines allowing robust high-precision control of clock phase and frequency. The DCM consists of Delay Locked Loop (DLL), Digital Phase Shifter (DPS), Digital Frequency Synthesizer (DFS) and the Digital Spread Spectrum (DSS). In this reference design, DCM is used for the DLL functionality.

Two DCMs are used in this reference design for clock de-skewing. DCM_ext is used as an external DCM (i.e., the CLK0 output drives an OBUF and is an output of the FPGA). This output is looped back externally and connected to the CLKFB input of the same DCM through an IBUFG. The SiberCAM device is clocked by the same sys_clk. This ensures clock de-skewing.

DCM_int is used as an internal DCM. The CLK0 and CLK180 outputs are the clocks used by the logic inside the FPGA. Figure 3 shows DCM usage.

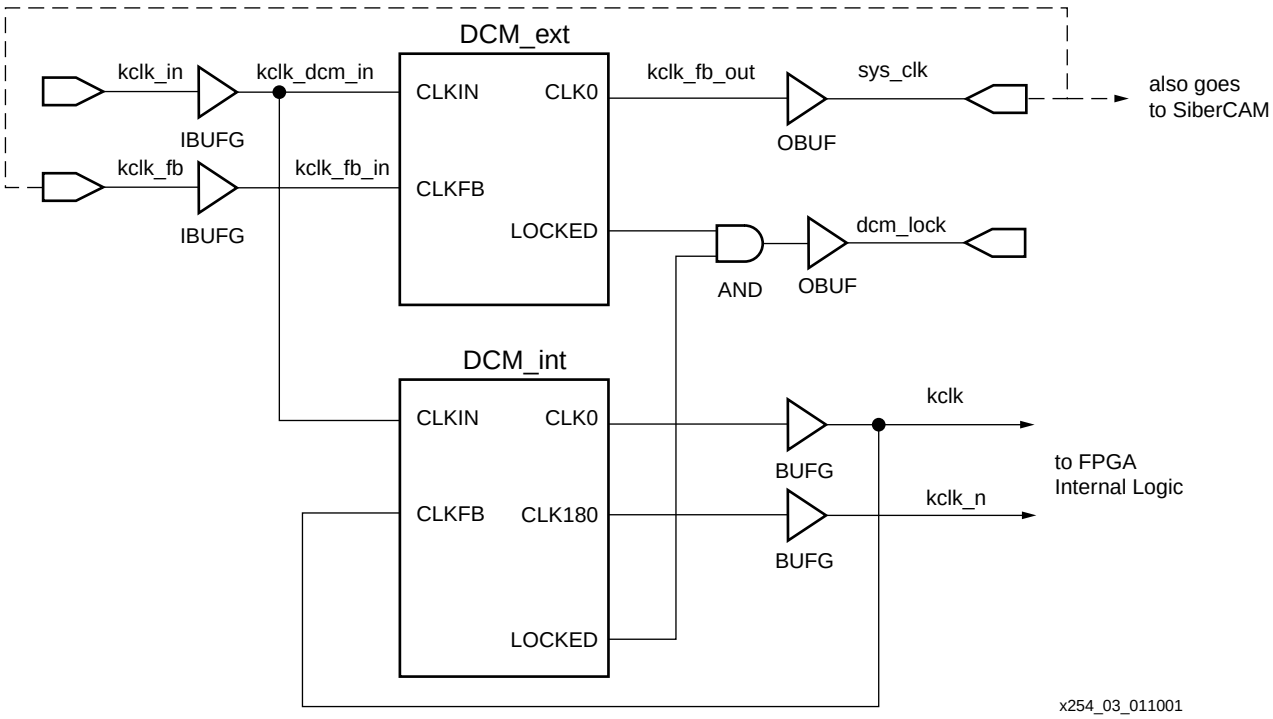


*Figure 3:* **DCM Usage**

### SiberBridge Block Diagram

Figure 4 shows a block diagram of the SiberBridge flow of information.

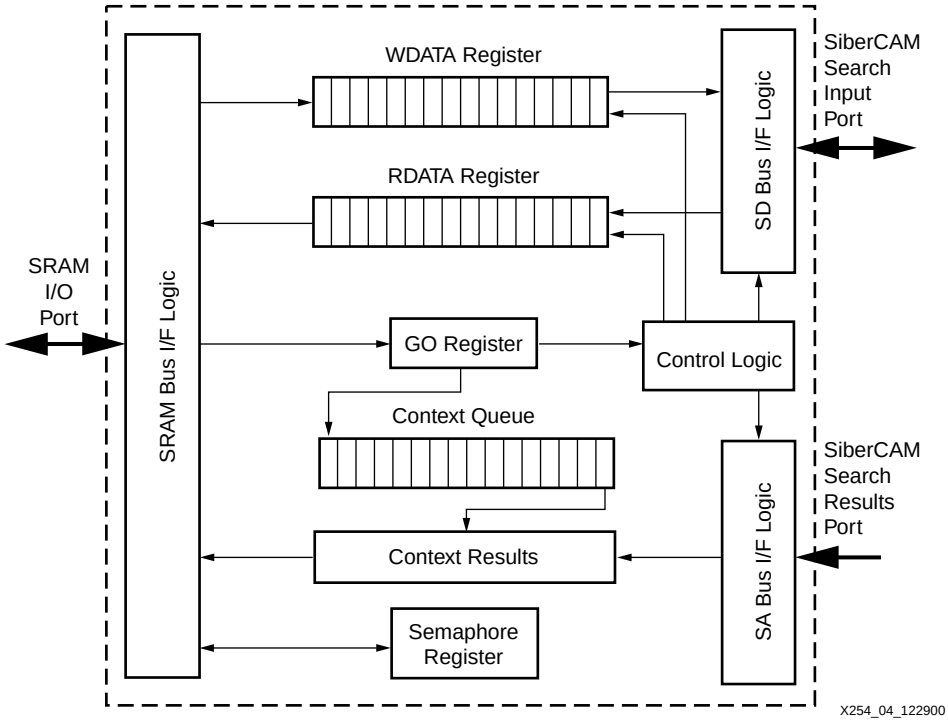


*Figure 4:* **SiberBridge Block Diagram**

To perform a maintenance operation with the SiberCAM device, the designer writes to the registers of the SiberBridge and then to the special GO register. This causes the SiberBridge to perform the maintenance operation using the expected protocol.

The SiberBridge is used with a SiberCAM device operating in 2-port mode. The SiberBridge communicates with the SiberCAM device over the 72-bit search data (SD) bus using double data rate (DDR) signaling. The SiberBridge accepts requests for data in 32-bit quantities and packages the data into the 36-bit/72-bit multiplexed bursts expected by the SiberCAM device.

## SiberBridge Interfaces

### Host Interface

The SiberBridge connects to a host as a 32-bit SRAM. The signals that make up this interface are described in Table 1.

*Table 1:* **Processor Interface Signal Descriptions**

| Signal | Type | Description |
|---|---|---|
| KCLK_IN | Input | Rising edge active. This input is connected to the same clock that is connected to the CK input of the SiberCAM device. Since the SiberBridge uses both edges of the clock, it is important that the clock have a 50% duty cycle. |
| KRESETN | Input | Active-low, asynchronous reset. This input is connected to the same signal that is connected to the RSTN input of the SiberCAM device. |
| KRW | Input | 1 = Read, 0 = Write. KRW must be in the correct state when KCEN is asserted. |

*Table 1:* **Processor Interface Signal Descriptions *(Continued)***

| Signal | Type | Description |
|---|---|---|
| KADDR[7:2] | Input | Address Bus. KADDR is used to select registers inside the SiberBridge. KADDR must be in the correct state when KCEN is asserted. Since all registers in the SiberBridge are 32-bits wide and 32-bits aligned, there are no address bits 1 and 0. The least significant bit of the address bus is bit 2. |
| KCEN | Input | Active-low, Device Select. KCEN is used to qualify the address signals. |
| KDATA[31:0] | Input/output | Bidirectional Data Bus. KDATA is used to move data in and out of the SiberBridge. |
| KOEN | Input | Active-low output enable. |
| KMODE | Input | 0 = normal SRAM timing, 1 = ZBT SRAM timing |

## SiberCAM Interface

The SiberBridge signals that connect to the SiberCAM device are listed in Table 2.

*Table 2:* **SiberCAM Interface Signal Descriptions**

| Signal | Type | Description |
|---|---|---|
| SD[71:0] | Input/Output | Bidirectional Search Data bus. These signals are connected to the SD pins of the SiberCAM device. |
| SSN | Output | Search Port Select. This signal is connected to the SSN pin of the SiberCAM device. |
| OESN | Output | Output Enable. This signal is connected to the SSN pin of the SiberCAM device. |
| OP[3:0] | Output | Search Port Operation Select. These signals are connected to the OP pins of the SiberCAM device. |
| TAG[1:0] | Output | Search Tag. These signals are connected to the TAG pins of the SiberCAM device. |
| MSK[3:0] | Output | Search Mask. These signals are connected to the MSK pins of the SiberCAM device. |
| THDI[3:0] | Output | Search Thread. These signals are connected to the THDI pins of the SiberCAM device. |
| PM[2:0] | Output | Power Management. These signals are included to support future power management features in future SiberCAM products. |
| SA[25:0] | Input | Search Address Bus. Result of search operation. Qualified by HITN and MULTN. |
| HITN | Input | Hit result output. Asserted when search results in a hit. |
| MULTN | Input | Multiple hit output. Asserted with HITN when multiple entries in the CAM match the search data. |

*Table 2:* **SiberCAM Interface Signal Descriptions *(Continued)***

| Signal | Type | Description |
|---|---|---|
| THDO[3:0] | Input | Thread Outputs. |
| TASN | Input | SiberCAM Transfer Acknowledge. This signal is connected to the TASN pin of the SiberCAM device. |
| RDYS | Input | SiberCAM Ready. This signal is connected to the RDYS pin of the SiberCAM device. |

# SiberBridge Bus Operations

## Host Read/Write Operation (kmode=0)

Figure 5 shows a typical read and write cycle on the host interface when KMODE is set to a value of "0" enabling normal synchronous SRAM timing.

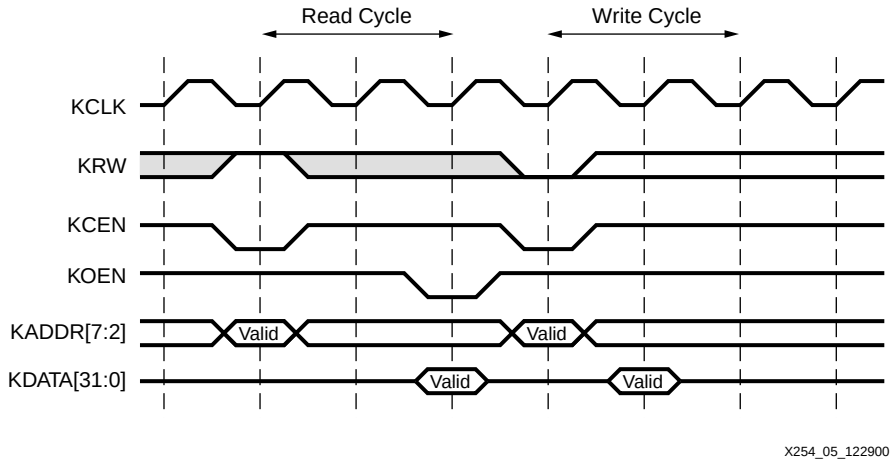


*Figure 5:* **Synchronous SRAM Mode Timing Diagram**

## Host Read/Write Operation (kmode=1)

Figure 6 shows a typical write cycle on the host interface when KMODE is a value of "1."

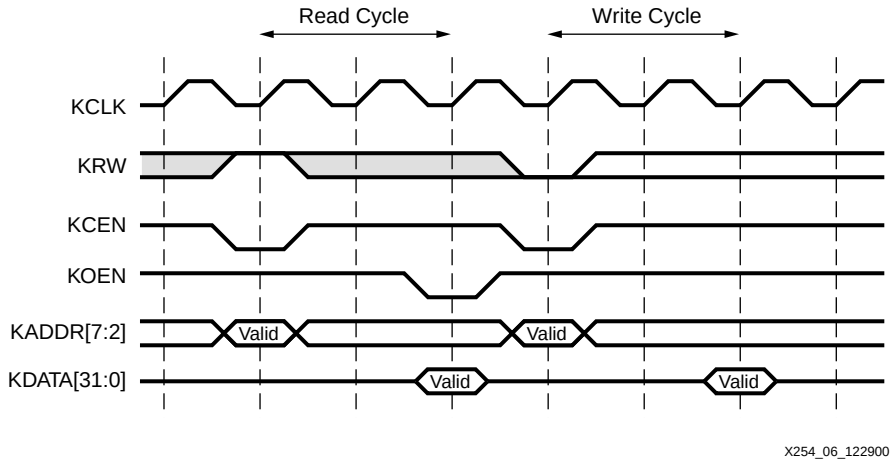


*Figure 6:* **ZBT SRAM Mode Timing Diagram**

## Registers

The registers, as outlined in Table 3, inside the SiberBridge are all 32-bits wide and 32-bits aligned and must be accessed with 32-bit transfers. Accesses to registers with 8-bit and 16-bit transfers are not supported.

*Table 3:* **SiberBridge Registers**

| Offset | Register Name | Read/Write | Description |
|---|---|---|---|
| 0x00 | Unused | N/A | Reserved. |
| 0x04 | CONFIG | Read-Write | Reserved. |
| 0x08 | GO | Read-Write | Contains bit-fields that control the bursting and reading operations performed by the SiberBridge on the SiberCAM interface.<br>Bits[1:0] – burst length<br>Bits[3:2] – read size<br>Bits[4] – read = 1/write = 0<br>Bits[7:6] – TAG[1:0]<br>Bits[11:8] – MSK[3:0]<br>Bits[15:12] – OP[3:0]<br>Bits[19:16] – THDI[3:0]<br>Bits[22:20] – BS[2:0]<br>Bits[29:24] – Context[4:0]<br>The burst length bit fields control the amount of data that is burst onto the SiberCAM bus.<br>The read size bit fields determine the number of 72-bit quantities that are to be read when the read/write bit is a value of "1."<br>The bit fields TAG[1:0], MSK[3:0], OP[3:0], THDI[3:0], and BS[2:0] are the values driven to the SiberCAM bus when SSN is asserted.<br>The Context[4:0] bits select which of the context result registers will contain the result. |
| 0x0C | STATUS | Read-only | Bit[0] – busy<br>The busy bit is a value of "1," while the SiberBridge is performing transfers with the SiberCAM device. While this bit is a value of "1", the contents of the WDATA register should not be modified. |
| 0x10 | WDATA | Read-Write | Bits[31:0] of WDATA. WDATA is burst onto the SiberCAM device in 72-bit quantities depending upon the settings in the GO register. |
| 0x14 | WDATA | Read-Write | Bits[63:32] of WDATA. |
| 0x18 | WDATA | Read-Write | Bits[95:64] of WDATA. |
| 0x1C | WDATA | Read-Write | Bits[127:96] of WDATA. |
| 0x20 | WDATA | Read-Write | Bits[159:128] of WDATA. |
| 0x24 | WDATA | Read-Write | Bits[191:160] of WDATA. |
| 0x28 | WDATA | Read-Write | Bits[223:192] of WDATA. |
| 0x2C | WDATA | Read-Write | Bits[255:224] of WDATA. |
| 0x30 | WDATA | Read-Write | Bits[287:256] of WDATA. |
| 0x34 | WDATA | Read-Write | Bits[319:288] of WDATA. |
| 0x38 | WDATA | Read-Write | Bits[351:320] of WDATA. |
| 0x3C | WDATA | Read-Write | Bits[383:352] of WDATA. |
| 0x40 | WDATA | Read-Write | Bits[415:384] of WDATA. |

*Table 3:* **SiberBridge Registers *(Continued)***

| Offset | Register Name | Read/Write | Description |
|---|---|---|---|
| 0x44 | WDATA | Read-Write | Bits[447:416] of WDATA. |
| 0x48 | WDATA | Read-Write | Bits[479:448] of WDATA. |
| 0x4C | WDATA | Read-Write | Bits[511:480] of WDATA. |
| 0x50 | WDATA | Read-Write | Bits[543:512] of WDATA. |
| 0x54 | WDATA | Read-Write | Bits[575:544] of WDATA. |
| 0x58 | RDATA | Read-Only | Bits[31:0] of RDATA. RDATA is loaded during read operations on the SiberCAM interface. The RDATA registers are loaded in the 72-bit quantities based on the settings of the GO register. |
| 0x5C | RDATA | Read-Only | Bits[63:32] of RDATA. |
| 0x60 | RDATA | Read-Only | Bits[95:64] of RDATA. |
| 0x64 | RDATA | Read-Only | Bits[127:96] of RDATA. |
| 0x68 | RDATA | Read-Only | Bits[159:128] of RDATA. |
| 0x6C | RDATA | Read-Only | Bits[191:160] of RDATA. |
| 0x70 | RDATA | Read-Only | Bits[223:192] of RDATA. |
| 0x74 | RDATA | Read-Only | Bits[255:224] of RDATA. |
| 0x78 | RDATA | Read-Only | Bits[287:256] of RDATA. |
| 0x7C | SEMAPHORE | Read-Write | Bits[31:0] semaphore<br>The contents of the semaphore register can only be updated with a non-zero value, if and only if, the value of the semaphore is a value of zero. |
| 0x80 | RESULT0 | Read-Only | Bitbit[31] – valid result flag<br>Bit[29] – multiple hit (MULTN inverted)<br>Bit[28] – hit (HITN inverted)<br>Bits[27:24] – THDO[3:0]<br>Bits[21:18] – SA[25:22]<br>Bits[17:0] – SA[17:0]<br>This register contains the results of a search initiated with the context bits of the GO register set to a value of "0." |
| 0x84 | RESULT1 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "1." |
| 0x88 | RESULT2 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "2." |
| 0x8C | RESULT3 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "3." |
| 0x90 | RESULT4 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "4." |
| 0x94 | RESULT5 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "5." |
| 0x98 | RESULT6 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "6." |
| 0x9C | RESULT7 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "7." |

*Table 3:* **SiberBridge Registers *(Continued)***

| Offset | Register Name | Read/Write | Description |
|---|---|---|---|
| 0xA0 | RESULT8 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "8." |
| 0xA4 | RESULT9 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "9." |
| 0xA8 | RESULT10 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "10." |
| 0xAC | RESULT11 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "11." |
| 0xB0 | RESULT12 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "12." |
| 0xB4 | RESULT13 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "13." |
| 0xB8 | RESULT14 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "14." |
| 0xBC | RESULT15 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "15." |
| 0xC0 | RESULT16 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "16." |
| 0xC4 | RESULT17 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "17." |
| 0xC8 | RESULT18 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "18." |
| 0xCC | RESULT19 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "19." |
| 0xD0 | RESULT20 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "20." |
| 0xD4 | RESULT21 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "21." |
| 0xD8 | RESULT22 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "22." |
| 0xDC | RESULT23 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "23." |
| 0xE0 | RESULT24 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "24." |
| 0xE4 | RESULT25 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "25." |
| 0xE8 | RESULT26 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "26." |
| 0xEC | RESULT27 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "27." |
| 0xF0 | RESULT28 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "28." |

*Table 3:* **SiberBridge Registers *(Continued)***

| Offset | Register Name | Read/Write | Description |
|---|---|---|---|
| 0xF4 | RESULT29 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "29." |
| 0xF8 | RESULT30 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "30." |
| 0xFC | RESULT31 | Read-Only | This register contains the results of a search initiated with the context bits of the GO register set to a value of "31." |

## Configuring the SiberCAM Device

The SiberCAM device is selected for 2-port operation by the static MODE pins. However, the size of the SiberCAM SD bus is selected by writing the appropriate pattern in to the SiberCAM configuration register. After hardware reset, the SD bus is configured for 18-bit operations. The WW bits in the SiberCAM configuration register must be programmed to a value of "10." The R bit must be programmed with a value of "1." This is done using following process:

The WDATA register is loaded with the following information:

- Bits [17:0] contain bits 17:0 of the load configuration opcode as described in the SiberCAM data sheet.
- Bits [143:18] are unused.
- Bits [161:144] contain bits 35:18 of the load configuration opcode as described in the SiberCAM data sheet.
- Bits [287:162] are unused.
- Bits [305:288] contain bits 17:0 of the load configuration data as described in the SiberCAM data sheet. WW bits must be programmed with a value of "10." The R bit must be programmed with a value of "1."
- Bits [431:306] are unused.
- Bits [449:432] contain bits 35:18 of the load configuration data as described in the SiberCAM data sheet. WW bits must be programmed with a value of "10." The R bit must be programmed with a value of "1."
- Bits [575:450] are unused.

The GO register is written with the following information:

- The burst length bit field must be programmed with a value of "11."
- The read/write bit must be programmed with a value of "0."
- The OP bit field must be programmed with the appropriate operation code as per the SiberCAM data sheet.

When the GO register is written, the SiberBridge performs the write transfer on the SiberCAM interface. The operation is complete after the busy bit in the STATUS register is a value of "0." Thereafter, all transfers between the SiberBridge and the SiberCAM device will be 72-bit DDR.

## State Machines and Diagrams

### MAIN_FSM Module

This module initiates the bursting of “write data” and generates the “busy” signal.

Figure 7 is the state machine for MAIN_FSM.

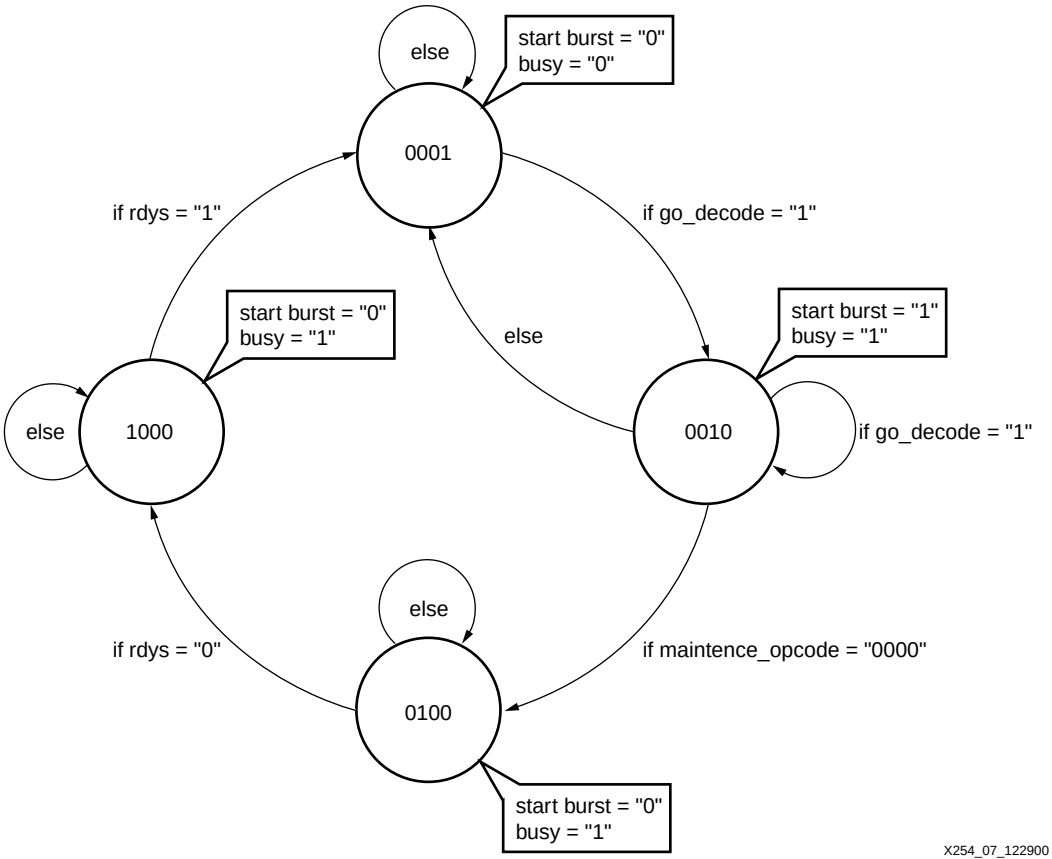


*Figure 7:* **Main_FSM State Machine**

## Packer Module

The Packer module generates select signals to identify which portion of the 288-bit read register will be written to during a read from the SiberCAM device. Figure 8 is the state machine for the Packer module.

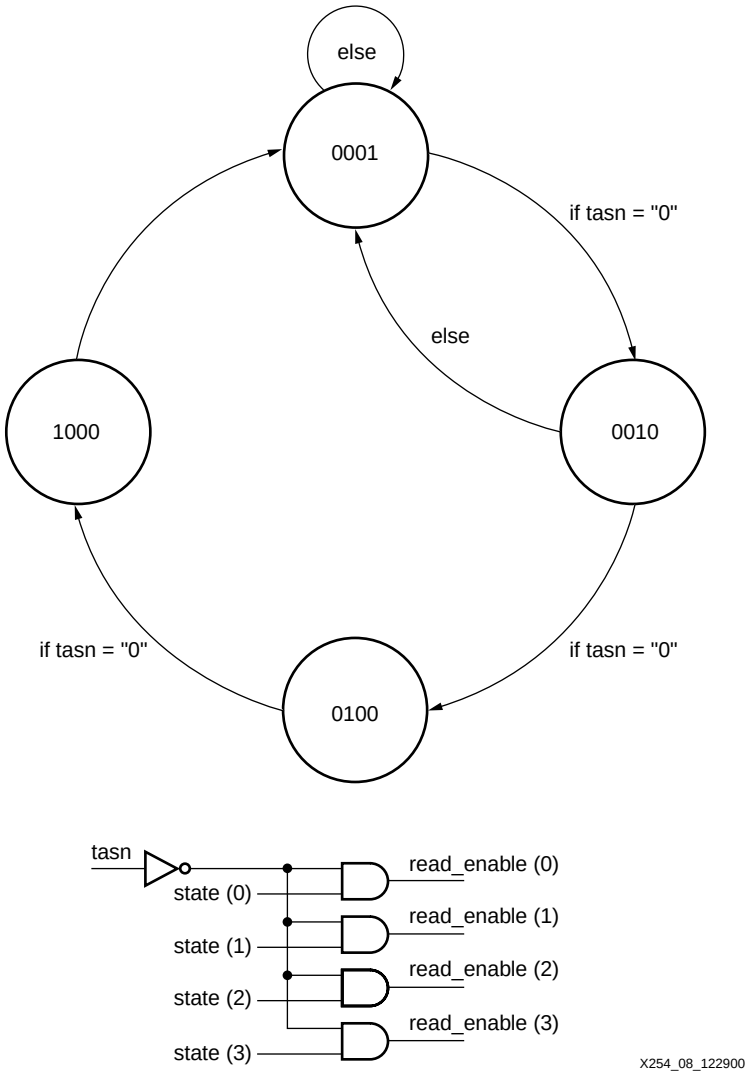


*Figure 8:* **Packer State Machine**

## Burster Module

The Burster module bursts data onto the SiberCAM device using DDR registers. Table 4 shows the Burst activity. Msel[2:0] selects the 72-bit block from the WDATA register to be bursted out. Msel changes its value on positive edge of kclk. With this, a total of 144 bits of WDATA are stored in nDDRin and pDDRin registers. The usage of special Double Data Rate registers for bursting out WDATA on SD bus is shown in Figure 9.

*Table 4:* **Burst Activity**

| msel[2:0] | nDDRin[71:0] | pDDRin[71:0] |
|---|---|---|
| 001 | wdata[71:0] | wdata[143:72] |

*Table 4:* **Burst Activity *(Continued)***

| msel[2:0] | nDDRin[71:0] | pDDRin[71:0] |
|---|---|---|
| 010 | wdata[215:144] | wdata[287:216] |
| 011 | wdata[359:288] | wdata[431:360] |
| 100 | wdata[503:432] | wdata[575:504] |

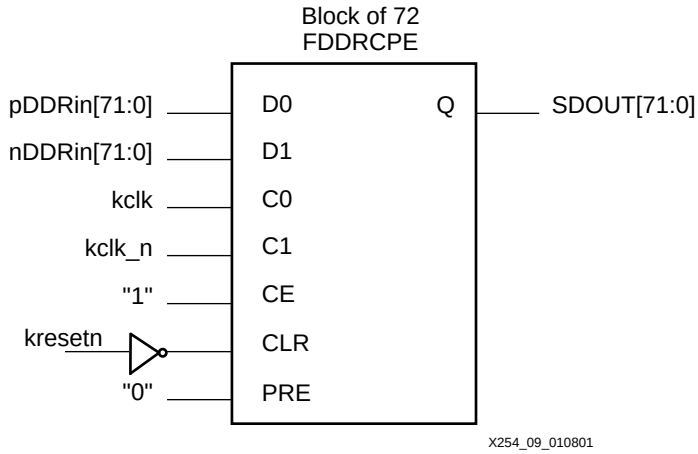


*Figure 9:* **BURSTER_MUX and BURSTER_DDR**

## Performing Maintenance Write Operations

A write operation is performed on the SiberCAM interface by writing data to the WDATA registers and then writing to the GO register. The number of 72-bit write operations performed on the SiberCAM interface are determined by the burst length bits of the GO register. Table 5 describes the burst activity based on the burst length bits:

*Table 5:* **Burst Activity**

| | |
|---|---|
| 00 | Two 72-bit (DDR) transfers |
| 01 | Four 72-bit (DDR) transfers |
| 10 | Six 72-bit (DDR) transfers |
| 11 | Eight 72-bit (DDR) transfers |

The bus usage for 72-bit DDR 2-port mode maintenance is described in the SiberCAM data sheet (www.sibercore.com). The WDATA bits are loaded according to the descriptions in the datasheet. For example, a 288-bit write entry operation is performed as follows:

The WDATA register is loaded with the following information:

- Bits [35:0] contain the opcode as described in the SiberCAM data sheet.
- Bits [143:36] are unused.
- Bits [179:144] contain the address as described in the SiberCAM data sheet.
- Bits [287:180] are unused.
- Bits [359:288] contain bits 71:0 of the write data.
- Bits [431:360] contain bits 143:72 of the write data.
- Bits [503:432] contain bits 215:144 of the write data.
- Bits [575:504] contain bits 287:216 of the write data.

The GO register is written with the following information:

- The burst length bit field must be programmed with a value of "11."

- The read/write bit must be programmed with a value of "0."
- The OP bit field must be programmed with the appropriate operation code as per the SiberCAM data sheet.

When the GO register is written, the SiberBridge performs the write transfer on the SiberCAM interface. The WDATA bits [71:0] are written on the SD inputs of the SiberCAM device to be read on the rising edge of the clock. WDATA bits [143:72] are written on the SD inputs of the SiberCAM device to be read on the next falling edge of the clock. WDATA bits [215:144] are written on the SD inputs of the SiberCAM device to be read on the next rising edge of the clock, etc. The operation is complete once the busy bit in the STATUS register is a value of "0."

## Performing Maintenance Read Operations

A read operation, like a write operation, is performed on the SiberCAM interface by writing data to the WDATA registers and then writing to the GO register. The number of 72-bit write operations performed on the SiberCAM interface are determined by the burst length bits of the GO register.

The bus usage for 72-bit DDR 2-port mode maintenance is described in the SiberCAM data sheet. The WDATA bits are loaded according to these descriptions. For example, a 144-bit read entry operation is performed as follows:

The WDATA register is loaded with the following information:

- Bits [35:0] contain the opcode as described in the SiberCAM data sheet
- Bits [143:36] are unused
- Bits [179:144] contain the address as described in the SiberCAM data sheet
- Bits [575:180] are unused

The GO register is written with the following information:

- The burst length bit field must be programmed with a value of "01."
- The read/write bit must be programmed with a value of "1."
- The read size "01."
- The OP bit field must be programmed with the appropriate operation code as per the SiberCAM data sheet.

When the GO register is written, the SiberBridge performs the write transfer on the SiberCAM interface. After bursting the opcode and address, the SiberBridge waits for and latches read data from the SiberCAM device. The 144 bits of read data will be copied into bits 143:0 of RDATA. The operation is complete once the busy bit in the STATUS register is a value of "0."

## Performing Search Operations

Search operations can be initiated on the SiberCAM interface by writing data to the WDATA registers and then writing to the GO register. The number of 72-bit write operations performed on the SiberCAM interface are determined by the burst length bits of the GO register. Table 6 describes the burst activity based on the burst length bits:

*Table 6:* **Burst Activity**

| | |
|---|---|
| 00 | 36-bit, 72-bit, 144-bit searches |
| 01 | 288-bit searches |
| 10 | Do not use this setting for searches |
| 11 | Do not use this setting for searches |

The WDATA registers for a 36-bit search are loaded as follows:

- Bits [35:0] of WDATA are loaded with bits 35:0 of the search data
- Bits [575:36] are unused

The WDATA registers for a 72-bit search are loaded as follows:

- Bits [71:0] of WDATA are loaded with bits 71:0 of the search data.
- Bits [575:72] are unused.

The WDATA registers for a 144-bit search are loaded as follows:

- Bits [143:0] of WDATA are loaded with bits 143:0 of the search data.
- Bits [575:144] are unused.

The WDATA registers for a 288-bit search are loaded as follows:

- Bits [287:0] of WDATA are loaded with bits 287:0 of the search data.
- Bits [575:288] are unused.

To initiate the search, the GO register is written with the following information:

- The burst length bit field must be programmed as per the list above
- The read/write bit must be programmed with a value of "0."
- The OP bit field must be programmed with the appropriate operation code as per the SiberCAM data sheet.

The operation is complete once the busy bit in the STATUS register is a value of "0."

## Results

Table 7 lists a summary of the device utilization and the resulting performance. The device targeted in the reference design is the 2V1000-5FG456.

*Table 7:* **Device Utilization Summary**

| Devices | Utilization | Performance |
|---|---|---|
| Number of External IOBs | 168 out of 324 | 51% |
| Number of RAMB16s | 1 out of 40 | 2% |
| Number of SLICEs | 838 out of 5120 | 16% |
| Number of BUFGMUXs | 3 out of 16 | 18% |
| Number of DCMs | 2 out of 8 | 25% |
| Number of TBUFs | 1024 out of 2560 | 40% |

### Reference Design

The Xilinx SiberBridge reference design is available on the web (**xapp254.zip**). It includes the Verilog source files. Please contact **SiberCore Technologies at info@sibercore.com** to obtain information on the SiberCAM product line.

## Conclusion

SiberBridge is a reference design for interfacing a 32-bit host to a SiberCAM device. It uses the special features of the Virtex-II DCM for clock deskewing. Using the dedicated block RAM for saving context results achieves better results. The DDR registers in the Virtex-II IOB are used for double-data-rate bursting of data on the SiberCAM device. The design is fully synthesizable and tested at 100 MHz. It is beyond the scope of this application note to describe each and every block of the SiberBridge design. The Verilog reference design is modular and easy to understand. A "user constraint file" is also provided for instantiation by the place and route tools.

## Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---|---|---|
| 01/12/00 | 1.0 | Initial Xilinx release. |