SigmaRAM™ Echo Clocks

Introduction

High speed, high throughput cell processing applications require fast access to data. As clock rates increase, the amount of time available to access and register data downstream decreases. The usefulness of output data at a downstream device is directly related to its relationship to that device's clock domain. At clock rates of 250 MHz and above, clock skew and output data parameter degradation due to temperature and voltage variations must be taken into consideration. A clock tree providing clocks to several chips across a circuit board may exhibit clock edge skew as much as several hundred picoseconds. Temperature and voltage variations can also result in clock and data skews of similar magnitude. These clock and data skews, combined with the effects of impedance mismatch, can lead to misaligned data and loss of data integrity. The advent of source-synchronous clocks in the SigmaRAM architecture is a time-tested approach to addressing the problem of high-speed data alignment and data integrity.

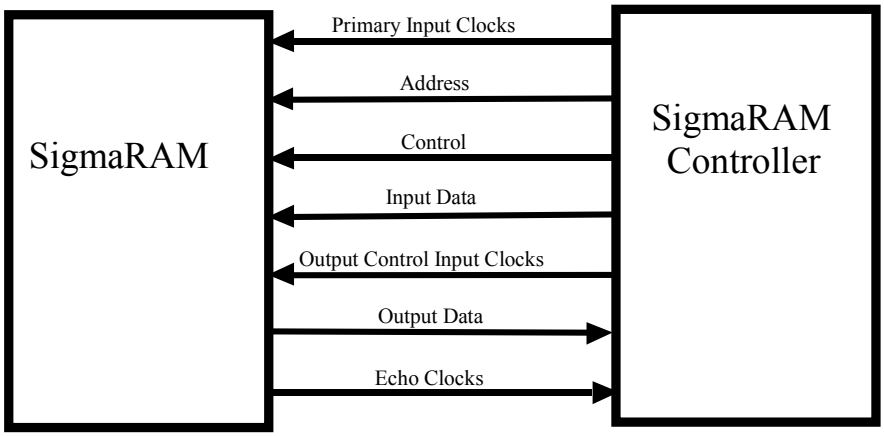

Figure 1: Simplified SigmaRAM Interface

A basic SigmaRAM interface is shown in **Figure 1**. The SigmaRAM controller can be any processor, ASIC, or FPGA designed to create and drive memory control signals.

Primary input clocks refer to input clocks driven to the SRAM that are used to latch address, control and input data during read and write operations. By definition, all synchronous SRAMs, including SigmaRAMs, utilize at least one primary input clock.

Output control input clocks refer to input clocks driven to the SRAM that are used to clock a pipelined SRAM's output data register during read operations. When implemented, they are used primarily to adjust the position of the output data valid window with respect to the primary input clocks. However, they have very little impact on the size of the output data valid window, and are, therefore, discounted in the remainder of this document.

Echo clocks refer to output clocks driven from the SRAM. They are phase-delayed versions of the primary input clocks, and are typically the same frequency as the primary input clocks. They are also registered in a similar fashion as output data. That is, both echo clock registers and output data registers are fired from the same internal signal, resulting in output clock edges very tightly aligned with output data edges. **Figure 2** shows the echo clock output data timing relationship in a 250 MHz SDR SigmaRAM.

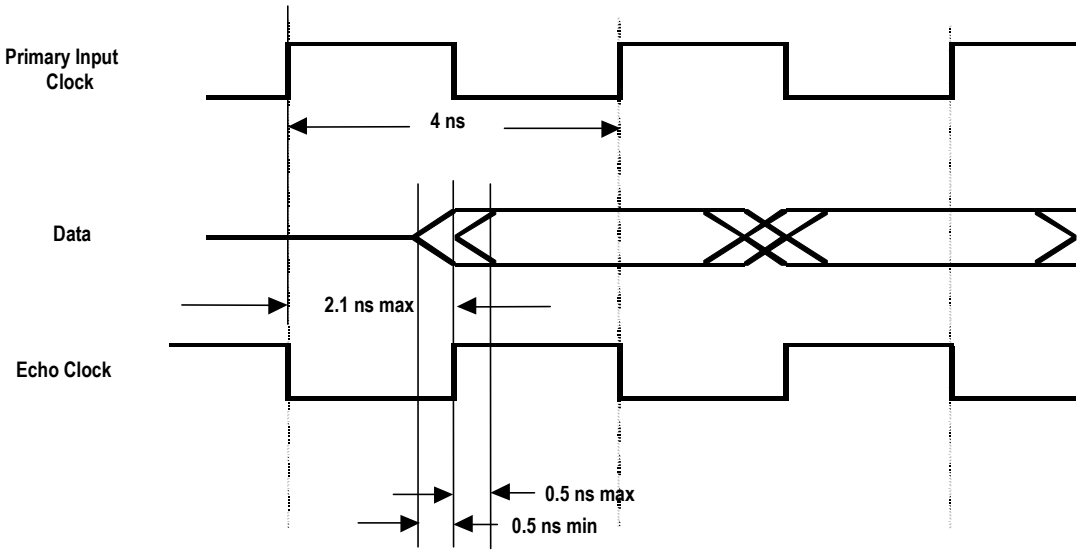

Figure 2: Clock Relationships

Purpose

Like all semiconductors, SRAMs are subject to process variations between chips. These process variations directly impact output timing—the better the process, the faster the output delay, and the worse the process, the slower the output delay. Without compensation, these variations can create difficult timing problems when using similar parts from different suppliers, or even when using different lots from the same supplier. At slower clock rates, these variations have a minimal impact, but at higher speeds they can be significant.

Semiconductors, like SRAMs, also exhibit a well known sensitivity to temperature and voltage. Temperature gradients across a circuit board may cause variations of up to several hundred picoseconds in output delay. This delay may be uneven across devices on a board or even across outputs on a single chip. Voltage is another source of uncertainty on a circuit board. In principle, the supply voltage at all parts on a board is the same at all times. In practice, supply voltages vary slightly as loads switch on and off. In addition, changes in supply voltage from chip to chip may also cause variation in output timing.

Process, temperature, and voltage variations combine to affect output delay rather severely. The delay is maximized at worst-case process, high temperature, and low voltage; and minimized at best-case process, low temperature, and high voltage. The operation of a given semiconductor is guaranteed to be within these two boundary conditions, as specified in the published timing parameters for the given device.

At DDR data rates, output delay variations of 0.5 ns can be critical. At 250 MHz, for example, the output data driven from a DDR SRAM during a read cycle can change approximately every 2 ns. Using an external clock to locate and strobe this high speed DDR data can be a problem.

SRAMs that provide echo clocks resolve the problem of data location by providing output data reference clocks that respond to variations in process, temperature, and voltage in virtually the exact same manner as output data, effectively negating the influence of the process, temperature, and voltage variations. In addition, because the clocks are source synchronous, using echo clocks automatically compensates for buffer delay variations and differences in physical placement of multiple SRAMs. In SigmaRAMs, this feature is taken one step further by offering a set of complimentary echo clocks on each side of the package. In this way the tightest alignment to data is achieved.

The key point in this discussion is how the size of the output data valid window of an SRAM varies depending on to which clock the output data is referenced. Ideally, the output data valid window referenced to any clock should be equal to a full clock period for SDR operation (i.e., 4 ns in a 250 MHz SDR SRAM) and half of a clock period for DDR operation (i.e., 2 ns in a 250 MHz DDR SRAM). The problem is that output data delay is subject to the process, temperature, and voltage variations previously discussed. Consequently, the ideal case is not achievable.

Output Data Valid Window Referenced to Primary Input Clocks

A typical datasheet specifies maximum primary input clock to output data delay as tKHQV max, and minimum primary input clock to output data delay (also referred to as output data hold time) as tKHQX min. For example, SigmaRAM datasheets (both SDR and DDR) specify tKHQV max as 2.1 ns and tKHQX min as 0.5 ns at 250 MHz operation. The range of these values represents variability in the certainty of the output data valid window and must, therefore, be subtracted from the ideal output data period. The result is a significantly smaller data valid window than the ideal case. Using the 250 MHz example above, the tKHQV range is 1.6 ns. Consequently, in a 250 MHz SDR SigmaRAM, the output data valid window is equal to 4 ns – 1.6 ns = 2.4 ns, or 60% of the theoretical maximum. And, in a 250 MHz DDR SigmaRAM, the output data valid window is equal to 2 ns – 1.6 ns = 0.4 ns, or just 20% of the theoretical maximum.

Output Data Valid Window Referenced to Echo Clocks

Clearly the main issue in this analysis is the variability of output data delay relative to the clock (whichever clock that may be) and its effect on the size of the output data valid window. Echo clocks, fired coincident with output data, and thereby tightly aligned with output data, help reduce this variability and, hence, maximize the size of the output data valid window. For example, SDR SigmaRAM data sheets specify that echo clocks be aligned within ±0.5 ns of output data at 250 MHz operation. Consequently, the resulting output data valid window is equal to 4 ns – (0.5 ns – (–0.5 ns)) = 3.0 ns, or 75% of the theoretical maximum. Similarly, DDR SigmaRAM datasheets specify that echo clocks be aligned within ±0.25 ns of output data at 250 MHz operation. Consequently, the resulting output data valid window is equal to 2ns – (0.25 ns – (–0.25 ns)) = 1.5 ns, or 75% of the theoretical maximum.

Figure 3 shows typical output data valid windows for SDR and DDR SRAMs, referenced to both a primary input clock and an echo clock.

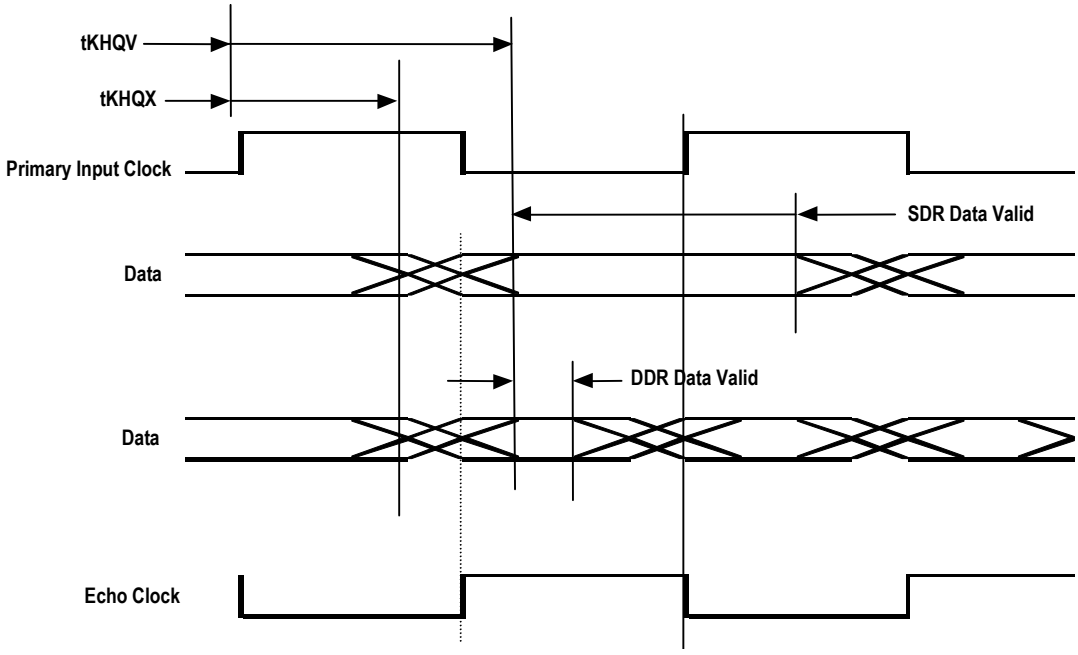

Figure 3: Data Valid Windows

Application

Scrutiny of the echo clock to output data relationship shows that the clock edge is coincident with the onset of data. Used downstream this would afford a maximum hold time but no setup time. How then should an echo clock be used?

Typically the clock used to capture the SRAM's output data should be centered in the output data valid window. In SDR SigmaRAMs, output data only changes from the rising edge of the input clock during read operations. Therefore, the rising edge of Echo Clock Complement is ideally positioned to capture the SRAMs output data because it is already centered in the output data valid window. Echo Clock Complement can be used to capture data in the SRAM controller with minimal timing adjustment.

In DDR SigmaRAMs, output data changes from both the rising and falling edges of the input clock during read operations. Therefore, neither the rising edge of Echo Clock nor the rising edge of Echo Clock Complement are centered in the output data valid windows at the output pins of the SRAM. As a result Echo Clock must be shifted 90 degrees (a quarter of a cycle), to center its rising edge in the output data valid window of the first piece of data. Echo Clock Complement must be shifted 90 degrees (a quarter of a cycle), in to center its rising edge in the output data valid window of the second piece of data. Once shifted, the rising edges of Echo Clock and Echo Clock Complement can be used to capture data in the SRAM controller reliably.

Transferring Output Data to the SRAM Controller's Internal Clock Domain

When echo clocks are used to capture SRAM output data in the SRAM Controller, the data must be transferred to the controller's internal clock domain. There are different ways to accomplish this when using SDR and DDR SRAMs. One method is described below for each type of SRAM.

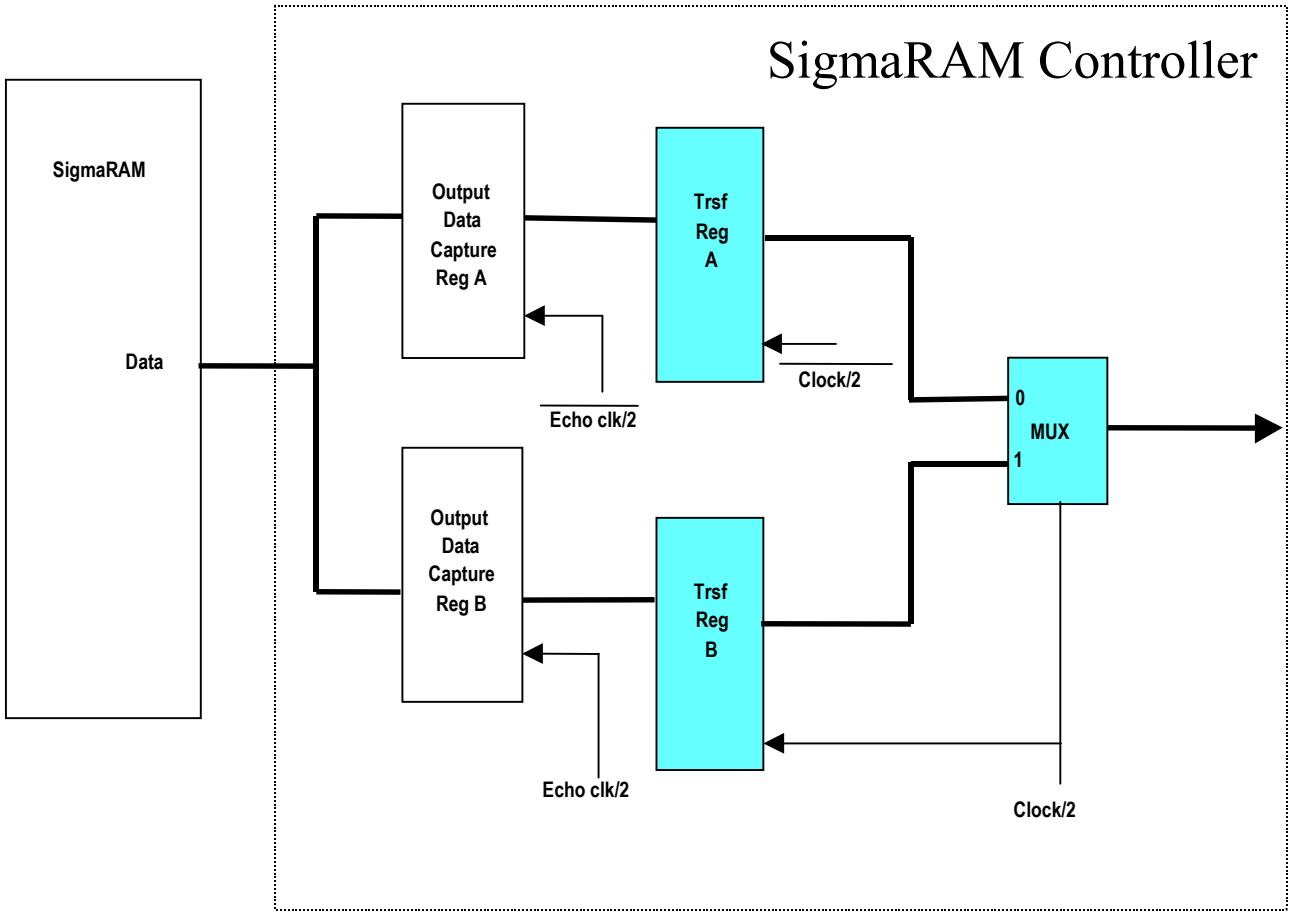

Figure 4: SigmaRAM SDR Controller

When using SDR SigmaRAMs:

1. Implement two Output Data Capture Registers (A and B) and two Data Transfer Registers (A and B).
2. Capture the output data driven during each read operation into either Output Data Capture Register (A or B) using the rising edges of Echo Clock/2 and Echo Clock /2 Complement. The output of each Output Data Capture Register will be valid for two consecutive cycles.
3. Capture the output of Output Data Capture Register A into Data Transfer Register A using the rising edge of the SRAM Controller's Clock/2. Capture the output of Output Data Capture Register B into the Data Transfer Register B using the rising edge of the SRAM Controller's Clock/2 Complement. At this point the data has been transferred to the controller's internal clock domain.

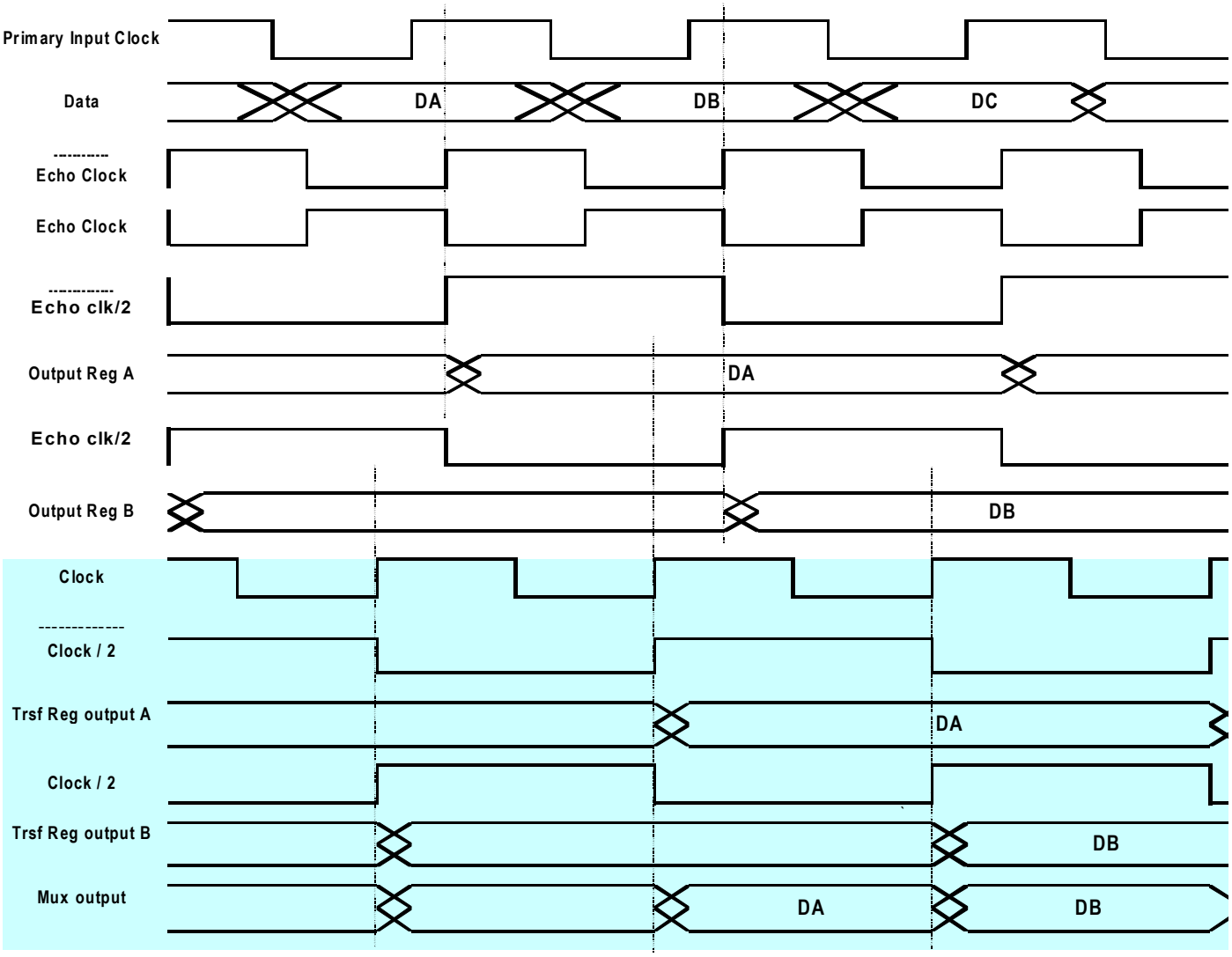

Figure 5: SDR Timing

When using DDR SigmaRAMs:

1. Implement one Output Data Hold Register, two Output Data Capture Registers (A and B), and two Data Transfer Registers (A and B). The two Output Data Capture Registers and the Data Transfer Registers should be twice as wide as the Output Data Hold Register.
2. Capture the first piece of output data driven during each read operation into the Output Data Hold register using the rising edge of Delayed Echo Clock.
3. Capture the second piece of output data driven during each read operation into the Output Data Capture Registers (A or B) using the rising edge of Delayed Echo Clock/2 and Delayed Echo Clock/2 Complement. At the same time, capture the output of the Output Data Hold Register into the other half of the same Output Data Capture Register (A or B). The output of each Output Data Capture Register will be valid for two consecutive cycles.
4. Capture the output of the Output Data Capture Register A into the Data Transfer Register A using the rising edge of the SRAM Controller's Clock/2 Complement. Capture the output of the Output Data Capture Register B into the Data Transfer Register B using the rising edge of the SRAM Controller's Clock/2. At this point the output has been transferred to the controller's internal clock domain.

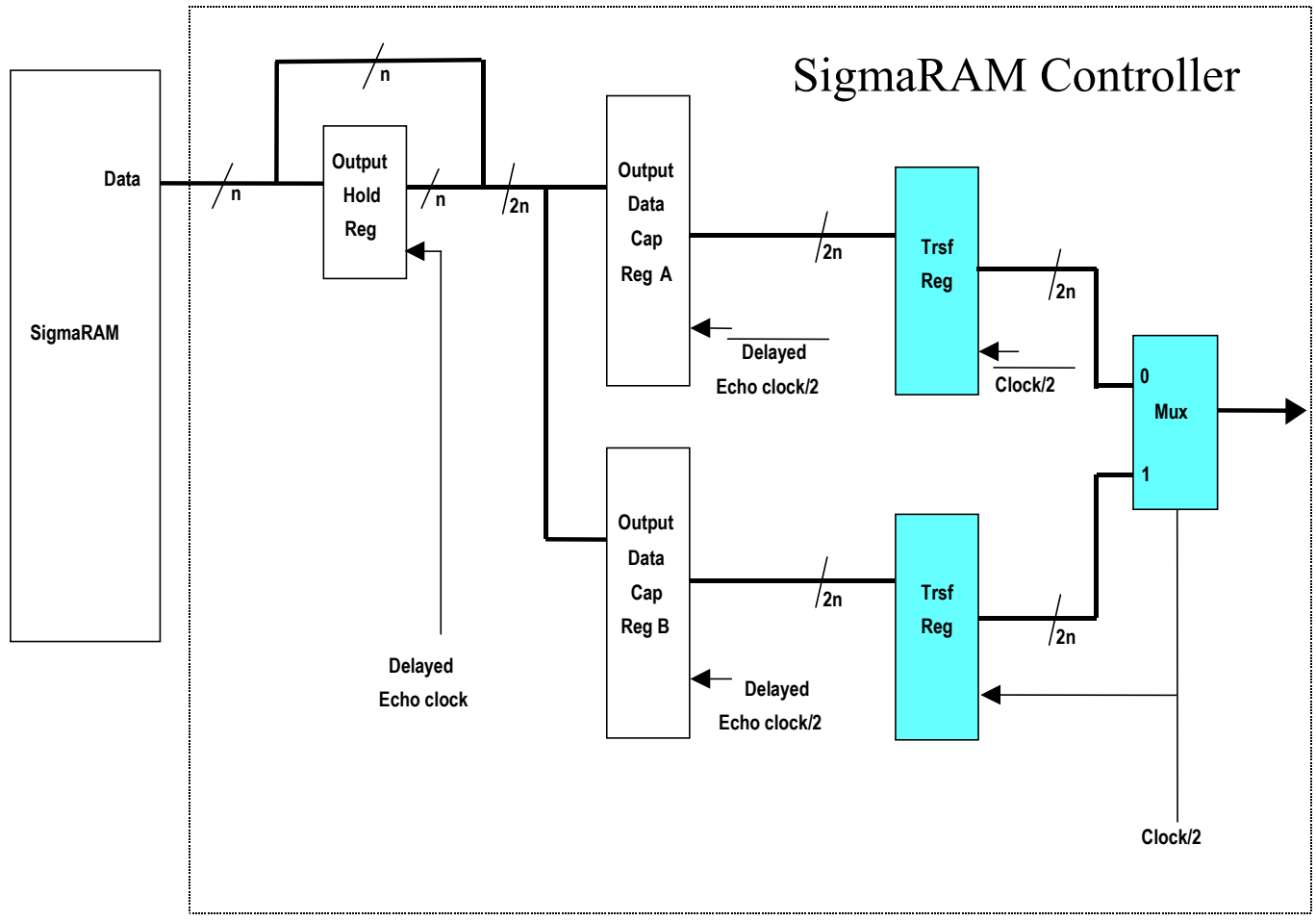

Figure 6: SigmaRAM DDR Controller

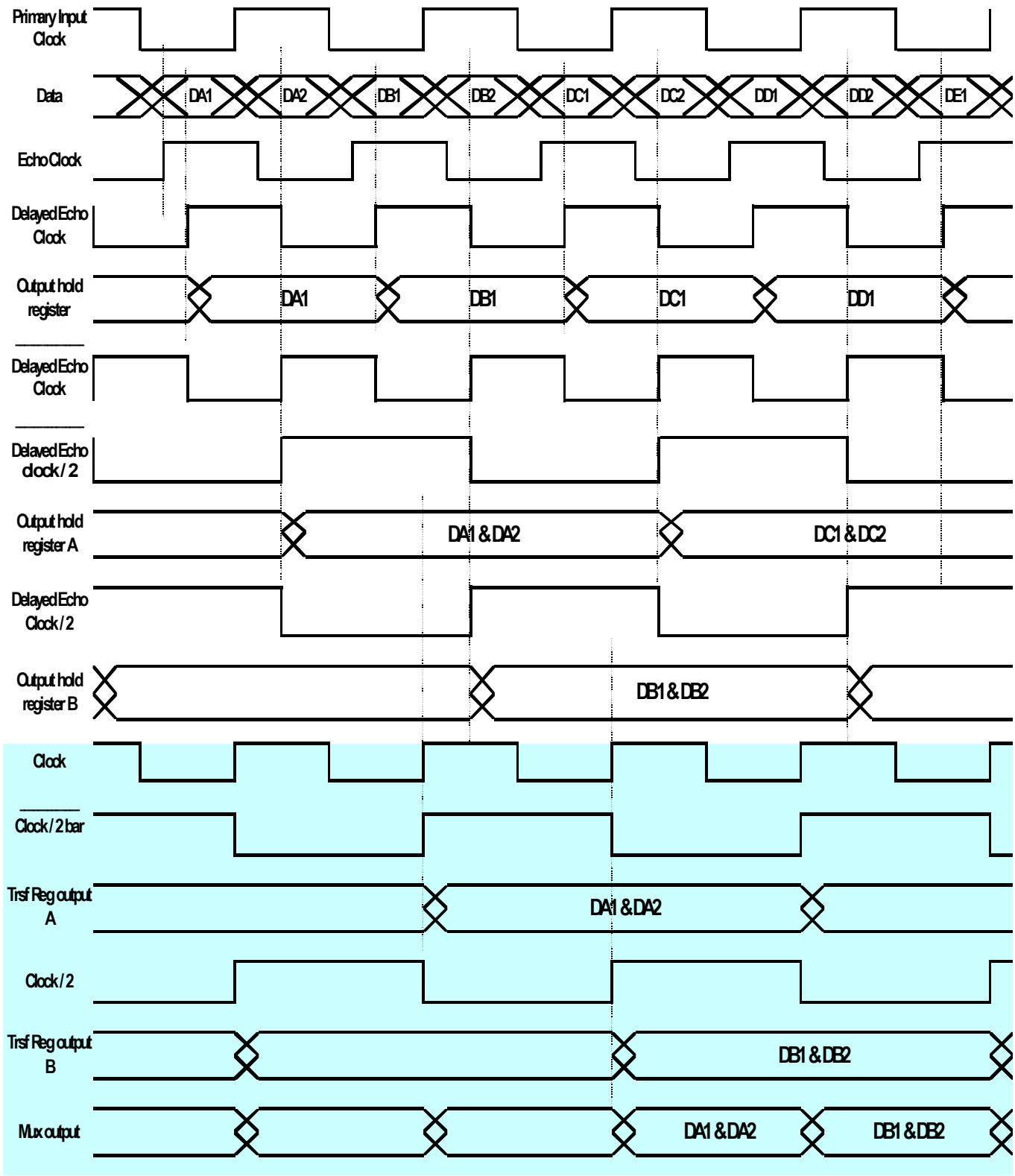

Figure 7: DDR Timing

The use of two Output Data Capture Registers and two Data Transfer registers affords considerable latitude in the position of the controller clock edge. By making the echo clock domain output valid for two clock periods, the controller clock can be positioned at any point within its two-clock window that allows sufficient setup and hold time. This design enables the SRAM Controller to tolerate significant skew between SRAMs in a given memory subsystem. Additional Output Data Capture Registers may be implemented to increase the amount of tolerable output skew even further.

For more information on implementation of the SigmaRAM DDR interface, see the application note "SigmaRAM Interface for Virtex-II Devices" at www.sigmaram.com.