

Summary

A read-modify-write technique permits the RAM facility in XC4000 LCA devices to operate faster than with conventional read/write operation. In addition, safe operation is guaranteed using a clock at the RAM-cycle rate. As a design example, the implementation of a shift register is described.

Specifications

Minimum Cycle Times (estimated for XC4000-5)	
16 x 8-bit RAM	25 ns (30 ns with 50% duty cycle)
64 x 8-bit RAM	35 ns
Byte-wide Shift Register	20 ns

LCA Family

XC4000

Demonstrates

Fast RAM operation

Introduction

The timing requirements of the XC4000 RAM primitive are that the address must be set up before the start of the Write Enable pulse (WE) and held for a short time after its end; data must be set up before the end of WE and held until its end. While such requirements are not unusual, they are not easily met in an LCA device.

Conventionally, the presence or absence of WE determines whether a RAM cycle is a read or write. Operating in this way, time is wasted generating the WE pulse each cycle. In larger RAMs, WE must also be gated to individual banks according to the address, wasting still more time. In addition, a clock rate twice the RAM-cycle rate is usually required to control the timing skew created in the generation and distribution WE.

With read-modify-write operation, the RAM is written every cycle. In write cycles, new data is written into the RAM; in read cycles, the data read from the RAM is rewritten, leaving the contents of the RAM unchanged. The unconditional write permits the clock signal to be used directly as WE, with WE asserted while the clock is High. The guaranteed low skew of the dedicated clock distribution nets simplifies the design, and increases its performance.

RAM Operation

A 16-word RAM is shown in Figure 1. The clock signal is used as the active-High WE, and no WE gating is provided. During the clock-Low period the address sets up on the RAM, data is read and registered on the rising edge of the clock. The flip-flops used for this register are available in every RAM-configured CLB and cannot be used independently since the DIN and H1 inputs are used in the RAM operation.

The Write signal determines whether old or new data is written during the clock-High period. The selected data must be set-up an appropriate time before the end of WE and held until the end of WE. This condition is easily satisfied if changes in the Input-Data and Write signals are triggered by the falling edge of Clock/WE.

The address hold time is satisfied by the output delay of the address register, provided that **both are driven directly from the same global clock net**. This can only be achieved using a BUFGS as the clock driver. The clock inverter is absorbed into the CLB to select the active clock edge, and does not create skew.

Several factors control the performance: The minimum clock-Low time is determined by the clock-to-set-up time from the address register to the read-data register, the minimum clock-High time is determined by the clock-to-set-up time from the read-data register to the trailing edge of WE in the RAM, and the total cycle time must permit both the Input Data and the Write signal to set up before the trailing edge of WE.

Figure 2 shows a 64-word RAM. Two banks of 32 words are used and both banks are written unconditionally. Write selection between the two banks is achieved using separate input data multiplexers. The Write controls is ANDed with bank select controls decoded from the address. Thus, data is only written to the appropriate bank. The output data from the appropriate bank is selected in a multiplexer to provide the read data output.

The pipeline stage at the multiplexer output is optional. If it is necessary to access the write data as it is being written, an additional connection may be made directly to the RAM outputs, and the data captured on the rising edge of the clock.



Figure 1. 16 x 8 RAM



Figure 2. 64 x 8 RAM

Shift Register

Large shift registers can be implemented in RAM, as shown in Figure 3. In this design, shift-register segments of 16 words or less are concatenated to any required length. Sixteen is chosen to minimize the RAM cycle time, and eliminate the need for bank switching or multiplexing.

In Figure 3, each segment has its own address counter. This is unnecessary in most cases, since address counters with the same modulus can be shared. A typical shift register might have one modulo-16 address counter shared among all segments except the last. The last segment has a separate shorter address counter to provide the desired length. The shift register can be tapped between any two segments, and separate address counters can provide arbitrary taps.

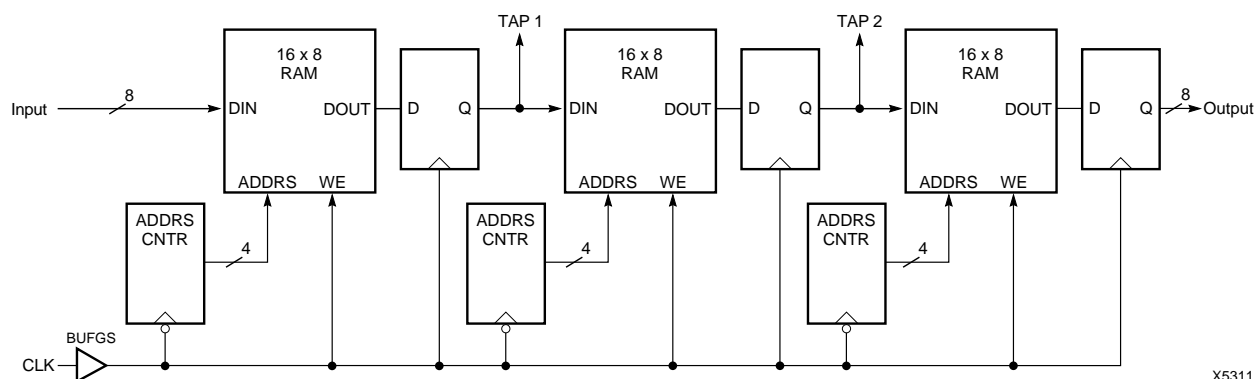
The address counters need not be conventional binary counters. Provided the address sequence cycles repeatedly, the order is irrelevant. A 4-bit counter with any desired sequence requires only two CLBs; all four flip-flop are fed back to the four function generators, which determine the next state of their respective flip-flops.

The shift register in Figure 3 does not have a shift-enable control. A multiplexer to re-write existing data into the RAM is, therefore, unnecessary since a write is performed every cycle.

Shift-Enable control can be added, using the clock enable scheme shown in Figure 4. New data is written into the RAM on every clock cycle, even when not shifting. The RAM address, however, is not updated while shift is disabled. Instead, it is held constant, and the corresponding RAM location is over-written repeatedly. The only data retained in the RAM is the data entered as the shift is re-enabled permitting the address counter to advance.

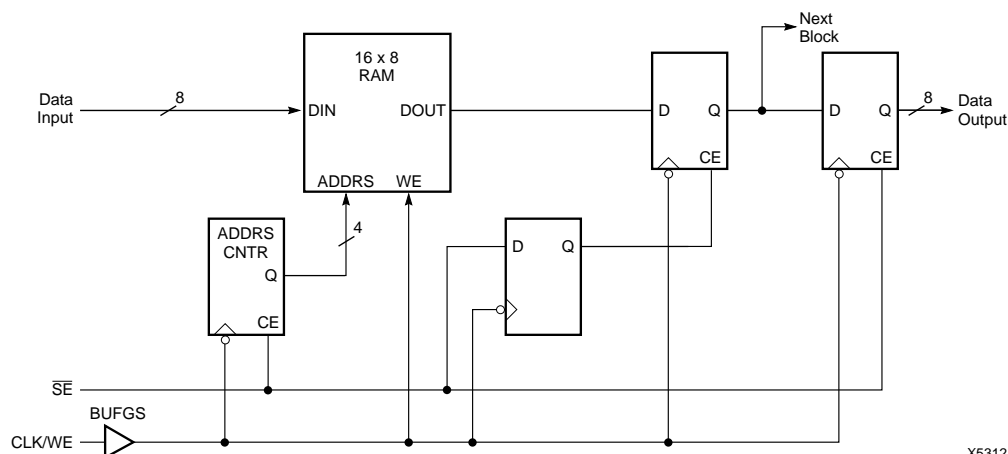
When shifting is disabled, the data that is stored in the register at the RAM output remains there until shifting resumes. At this time, it is passed to the final output register or to the next RAM. The final output register prevents the output from changing when shift is first disabled. It does not alter the length of the shift register; with a 16-word RAM, data appears at the output as a result of the falling clock edge exactly 16 clocks after the falling edge on which the data was clocked in. Without the register, data appears one-half clock earlier, and when shift is disabled, would already be present.

No additional register is required between shift-register blocks. However, if there is a shift-register tap between blocks, the tap must be provided with an output register. The Shift-Enable control must be valid to enable or disable registers and counters clocked on the falling clock edge.



X5311

Figure 3. Byte-wide RAM-based Shift Register



X5312

Figure 4. RAM-Based Shift Register with Shift Enable