White Paper

Using Parity to Detect Errors

Introduction

Altera® Stratix® II and Stratix devices feature the TriMatrix™ memory architecture, which is composed of three different-sized embedded RAM blocks. TriMatrix memory includes the 512-bit M512 blocks, the 4-Kbit M4K blocks, and the 512-Kbit M-RAM™ blocks.

TriMatrix memory blocks support a parity bit for each storage byte. The parity bit, along with internal logic resources, can implement parity checking for error detection to ensure data integrity. The parity generation and control logic is implemented outside the TriMatrix memory blocks, in the logic elements.

This white paper describes how parity can be used to detect errors and also illustrates one of the many different ways to do so: by using the Parity Detection Circuit design example.

Using Parity Methods for Error Detection

During data transmission from a transmitter to a receiver, electrical noise can corrupt the transmission signal. If the noise is substantial, it can alter the logic level of the signal, introducing errors in the transmitted signal. A parity bit is an extra 0 or 1 bit that is attached to the original signal and used to detect errors.

There are two parity methods, even and odd. In the even parity method, the value of the bit is chosen so that the total number of 1s in the transmitted signal, including the parity bit, is even. Similarly, with odd parity, the value of the bit is chosen so that the total number of 1s is odd. For example, for the following byte 11010000, the even parity bit would be 1, making the total number of 1s in the signal even, and the odd parity bit would be 0, making the total number of 1s in the signal odd.

You can determine if an error occurred during transmission by calculating the parity of the received bytes and comparing the generated parity with the transmitted parity. Parity can only detect an odd number of errors. If an even number of errors occurs, the computed parity will match the transmitted parity. Additionally, the parity method allows for error detection only; it cannot be used to correct errors because it does not provide a means of determining which bit is in error.

Parity Detection Circuit Design Example

The Parity Detection Circuit design example demonstrates how parity bits can be used to detect errors. The design example consists of a parity generator, a parity comparator, an error value generator, the memory block, and the data and control signal pipelines. Although the design example is for a single byte, you can easily expand it to work with multiple bytes by duplicating the control logic components. Even parity is used, where the ninth bit, `data[8]`, represents the transmitted parity value.

Parity detection is performed as follows:

- Incoming data is registered.
- Registered data is passed to the parity generator to calculate the parity of the byte.
- Calculated parity and transmitted parity are compared.
- Data is pipelined through a register to compensate for the combinatorial delays through the generator and comparator.
- Control signals `WREN` and `ADDRESS` are pipelined to match the data pipeline.
- If the parity values are the same, then the original value is sent to the memory block. If the parity values are different, an error value `H'1FF` is sent to the memory block to denote a transmission error.

Figure 1 shows the schematic of the parity detector circuit.

Figure 1. Parity Detector Schematic

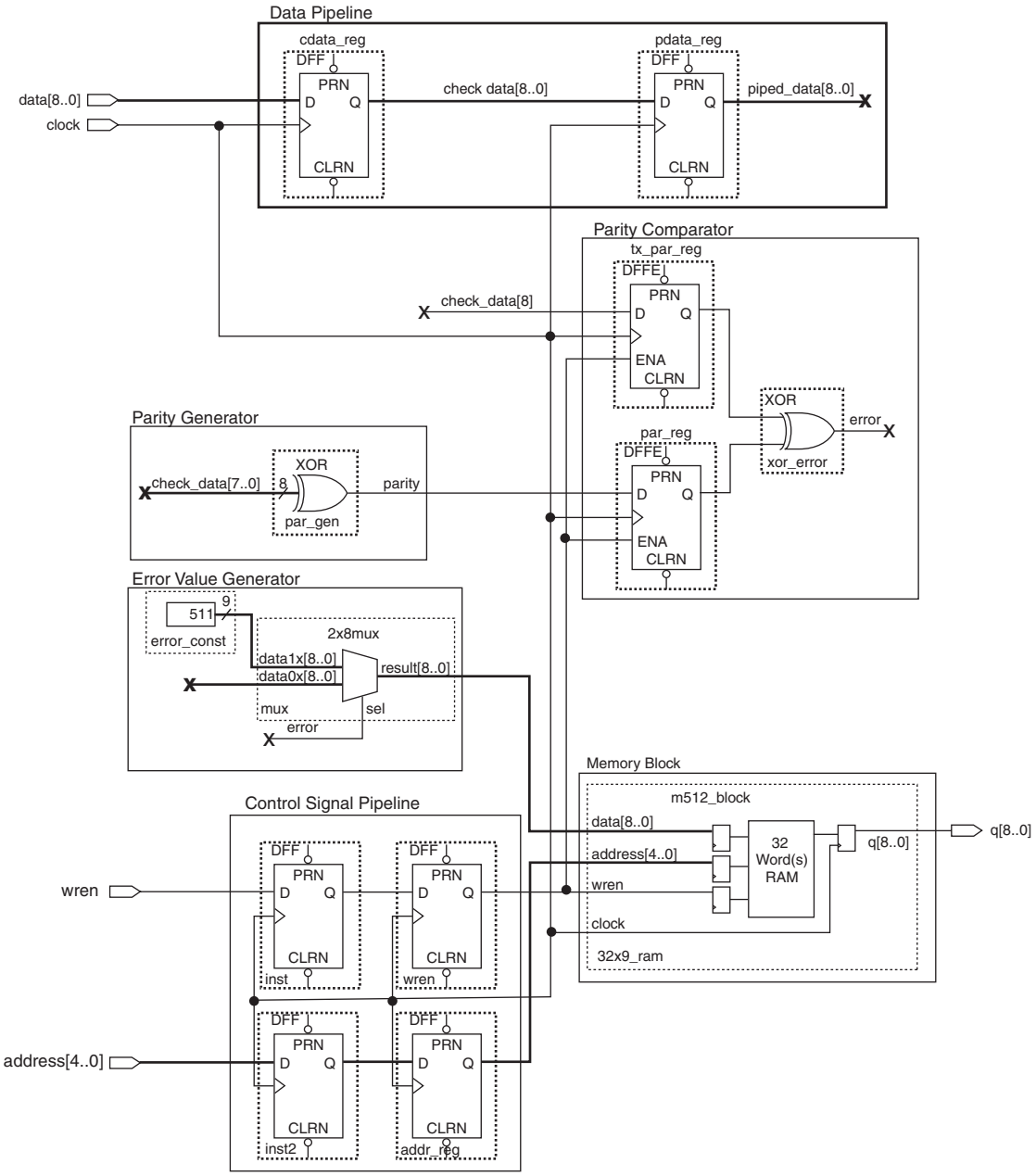


Table 1 describes the functionality of the inputs, outputs, and registers in the design example and simulation.

Table 1. Node Definitions

Node Name	Node Type	Description
clock	Input	System clock
wren	Input	Write enable to RAM block
address	Input	Address to RAM block
Data	Input	Data to RAM Block
Q	Output	Output Data from RAM block
wren_reg	Register	Pipelined wren
Addr_reg	Register	Pipelined address
pdata_reg	Register	Pipelined data
par_reg	Register	Calculated parity
tx_par_reg	Register	Pipelined transmitted parity (data[8])
xor_error	Combinatorial	Output of the parity comparator

Figure 2 illustrates the detection of a transmission error during a write operation to the memory block. The error value H'1FF is stored instead of the incorrect value H'114 so that the data is not used later in the data path. You can also create logic that requests the corrupted data to be resent so the memory block can be updated with the correct value.

Figure 2. Transmission Error Detection Example

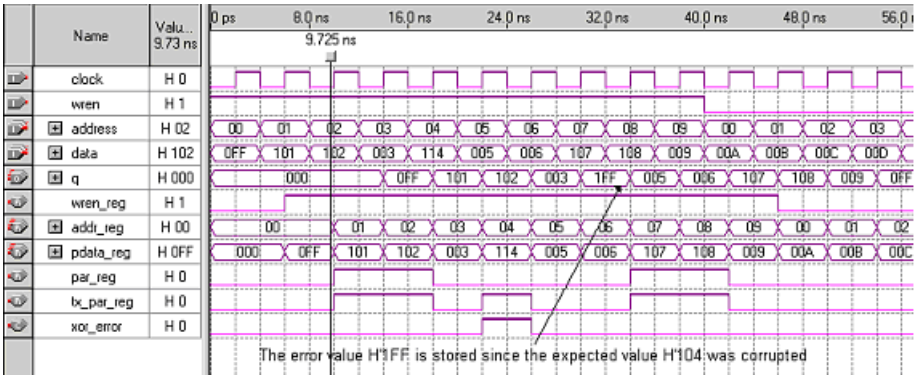
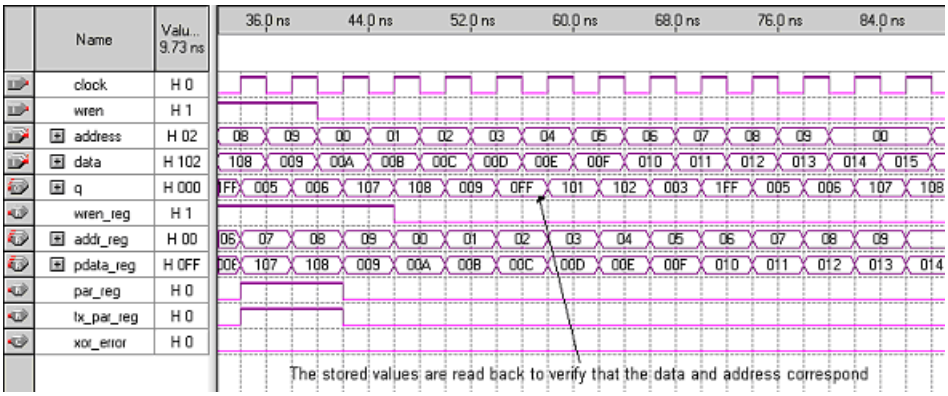


Figure 3 shows the stored data reading out from the memory block. The data and address correspond because the control signals have the same 2-stage pipeline as the data. The data is available after a three-clock cycle delay, from the two pipeline registers and the input register into the memory block.

Figure 3. Stored Data Reading Out Example



Conclusion

All three TriMatrix memory blocks feature parity-bit support, which can be used to detect errors when interfacing to external devices that support parity. The Parity Detection Circuit design example provides an easy way to perform error detection in Stratix II and Stratix devices.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com