(28, 24) Reed-Solomon Encoder

Features

- **Industry standard (28, 24) Reed-Solomon Encoder**
- **Available for FPGA or ASIC implementation**
- **High speed design, reaches 2.4 Gbps data rate in Virtex IV,, higher in ASIC**
- **Compact design, uses 75 CLB slices in Virtex IV, among the smallest on the market**
- **Can work continuously with no gap between code blocks**
- **Fully synchronous one clock design**
- **One clock cycle latency**

Functional Description

The block diagram of the encoder is shown in Figure 1. The major functional block of the encoder is the parity unit. While the encoder outputs the data symbol, the parity unit calculates the parity symbol at the same time. The parity symbols are appended at the end of the data block to form the whole code word.

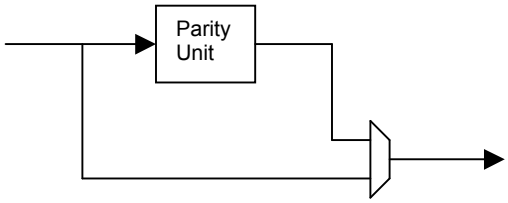

Figure 1. Block diagram of the Reed-Solomon encoder.

Pin Out

Figure 2 is the schematic symbol of the Reed-Solomon encoder.

RST
One bit input, the asynchronous reset. When RST is set high, all the internal flip-flops are asynchronously initialized. The core will stay in this state until RST is set low.

CLK
One bit input, the global clock. All sequential logic acts on the rising edge of CLK.

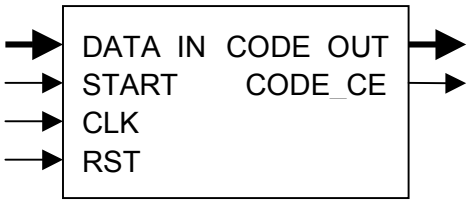

Figure 2. Schematic symbol of the encoder

START
One bit input, the signal to start the encoding process. At the rising edge of CLK, if START is high while RST is low, the internal state machine will start the encoding process. For continuous operation, the START signal for the nest code block must line up with the last parity byte of the previous code block. Otherwise, the START signal for the next code block must be at least five clocks after the last parity byte of the previous code block for the decoder to correctly decode the second code word. The length of START should be one clock cycle.

DATA_IN
Eight bit input, the data sequence to be encoded. The first byte of the data sequence should be one clock after the START pulse. The encoder reads in one byte every clock. Each data sequence must have 239 bytes and the encoder will append 16 parity check bytes at the end of the data sequence. For continuous operation, the first byte of the second data sequence must line up with the last parity byte of the first code word. Otherwise, the gap between the last parity byte of the first code word and the first byte of the following data sequence must be at least six clock cycles.

CODE_OUT
Eight bit output, the output code word. The first 239 bytes are the data sequence. The following 16 bytes are the parity bytes.

CODE_CE
One bit output, the clock enable for outputting the code word. The length of CODE_CE is 255 clock cycles.

Timing Diagrams

The Reed-Solomon encoder is very easy to be integrated into a larger design. The following timing diagrams help to clarify some of the synchronization issues.

Figure 3 shows the timing diagram at the starting point of the encoding process, where D0 is the first byte of the data sequence.

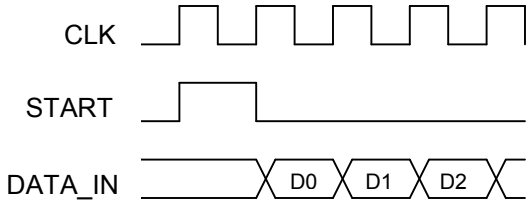

Figure 3. The timing diagram at the starting point of the encoding process

Figure 4 shows the timing diagram at the starting point of the output, where C0 is the first byte of the code word.

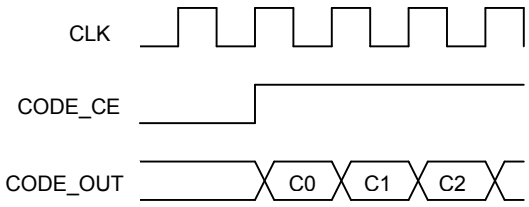

Figure 4. The timing diagram at the starting point of the output

Figure 5 shows the timing diagram of the continuous operation mode, where DA254 is the last parity byte of the first code word and DB0 is the first byte of the following data sequence word.

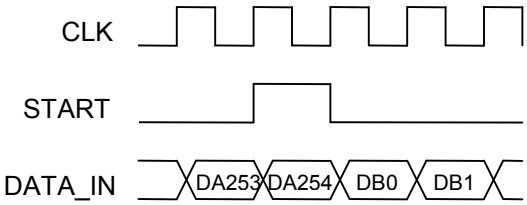

Figure 5. The timing diagram of the continuous operation mode

Deliverables

Deliverables include the encoder/decoder and the test bench. For Xilinx FPGA implementation, both source code and netlist are available. For ASIC implementation, only source code will be delivered. Source code can be in VHDL or Verilog.

Ordering Information

We have flexible licensing structures. Please use the following information to contact us:

Highland Communications Technologies
928 Concession Road, Fort Erie
Ontario, Canada L2A 6B8

Tel: 1-905-658-0989
Fax: 1-905-248-5188
Email: sales@highlandcomm.com

Web site:
http://www.highlandcomm.com/