# 2.5 Gbps GPON FEC Decoder

## Features

- **Fully compliant with the GPON standard (G.984.3), supports 8-bit, 16-bit or 32-bit input for different data rates, supports short last code word**
- **Available for FPGA or ASIC implementation**
- **High speed design, reaches 4.32 Gbps data rate in Virtex II, or 3.68 Gbps data rate in Spartan III, higher in ASIC**
- **Compact design, uses 3728 CLB slices in Virtex II, or 3766 CLB slices in Spartan III**
- **Fully synchronous one clock design**
- **718 clock cycle latency**

## Functional Description

The GPON standard uses the (255, 239) Reed-Solomon code as its FEC code. The decoder has three functional blocks and one memory block as shown in Figure 1. First the syndrome unit calculates the syndromes. Then the key equation solver solves the key equation for the error location polynomial. The correction unit calculates the error location and value and then adds the error sequence to the received code word to get the corrected code word. The memory unit is used to store the received code word while the decoder calculates the syndromes and solves the key equation. During the correction stage, the stored code word is read out from the memory and added to the error sequence to get the corrected code word.

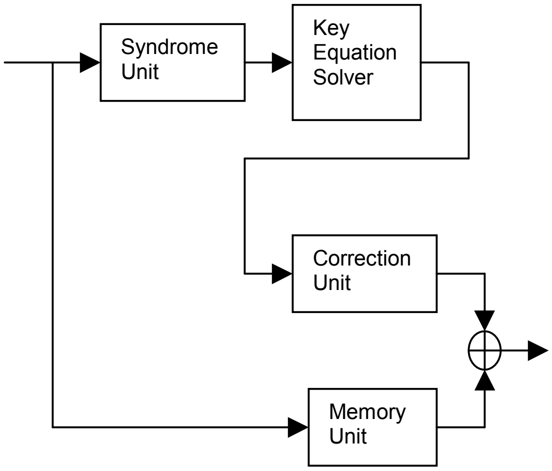

Figure 1. Block diagram of the Reed-Solomon decoder.

## Pin Out

Figure 2 is the schematic symbol of the decoder. The pins are explained as follows.

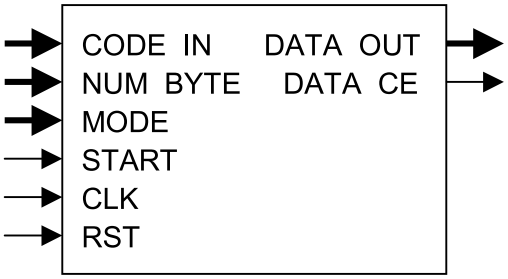

Figure 2. Schematic symbol of the decoder

### 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.

### 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.

### MODE

Two bit input. When mode is set to “00”, the 8 msb of CODE_IN is the input and the 8 msb of DATA_OUT is the output. When mode is set to “01”, the 16 msb of CODE_IN is the input and the 16 msb of DATA_OUT is the output. When mode is set to “10” or “11”, the full 32 bit CODE_IN is the input and the full 32 bit DATA_OUT is the output.

### NUM_BYTE

Sixteen bit input to specify the number of data bytes to be decoded. NUM_BYTE does not include the parity bytes. The minimum number of data bytes supported is 1 byte for 8 bit input, 2 bytes for 16 bit input, and 4 bytes for 32 bit input. The highest number of data bytes supported is that the total number of bytes (including data bytes, parity bytes, and the added zero bytes of the last short code word) is equal to 65535 in 8 bit input, 65534 in 16 bit input, and 65532 in 32 bit input. When specifying the number of data bytes you want to process, the number can be any value within the supported range for 8 bit input. For 16 bit input, the number can only be the even numbers within the supported range. For 32 bit input, the number can only be those which is a multiple of 4 within the supported range.

### CODE_IN

Thirty-two bit input, the code word stream to be decoded. When mode is set to “00”, only the 8 msb is used as input. When mode is set to “01”, only the 16 msb is used as input. When mode is set to “10” or “11”, the full 32 bit is used as input. The first data symbol (1-byte, 2-byte, or 4-byte) of the code word stream should be one clock after the START pulse. The decoder will read in one data symbol per clock until the last parity symbol of the last code word.

### DATA_OUT

Thirty-two bit output, the decoded data sequence. When mode is set to “00”, only the 8 msb is the output. When mode is set to “01”, only the 16 msb is the output. When mode is set to “10” or “11”, the full 32 bit is the output. DATA_OUT is not a continuous data stream. For every 239 data symbols there will be 16 clock cycles without data output. These clock cycles are for the parity bytes.

### DATA_CE

One bit output, the clock enable for outputting the data sequence. DATA_CE is used to enable the clock for the next processing module of the data sequence. Since DATA_OUT is not a continuous data sequence, DATA_CE is not always one. For every 255 clock cycles, DATA_CE will be one for the first 239 clock cycles and zero for the rest 16 clock cycles. For the last part of the data sequence, if there are less than 239 data symbols, DATD_CE will be one only for the number of clock cycles which is equal to the number of data symbols.

## Timing Diagram

The GPON FEC decoder is very easy to be integrated into a larger design. Figure 3 helps to clarify some of the synchronization issues, where CODE_CE labels when the code word stream should be fed into the decoder, and n = 1 when mode = “00”, n = 2 when mode = “01”, and n = 4 when mode = “10” or “11”.

Following the START pulse, the code word stream is fed into the decoder continuously, one data symbol (1 byte, 2 byte or 4 byte) per clock. There is no gap between the data bytes and the parity bytes of the last short code word.

The output from the decoder is the decoded data sequence. For every 255 clock cycles, there will only be 239 data symbols output. For the last part of the data sequence, the number of data symbols will be less than or equal to 239, depending on the number of left over data symbols. DATA_CE is used to enable the clock for the next processing module of the code word.

The latency of the decoder is 718 clock cycles.

## 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:

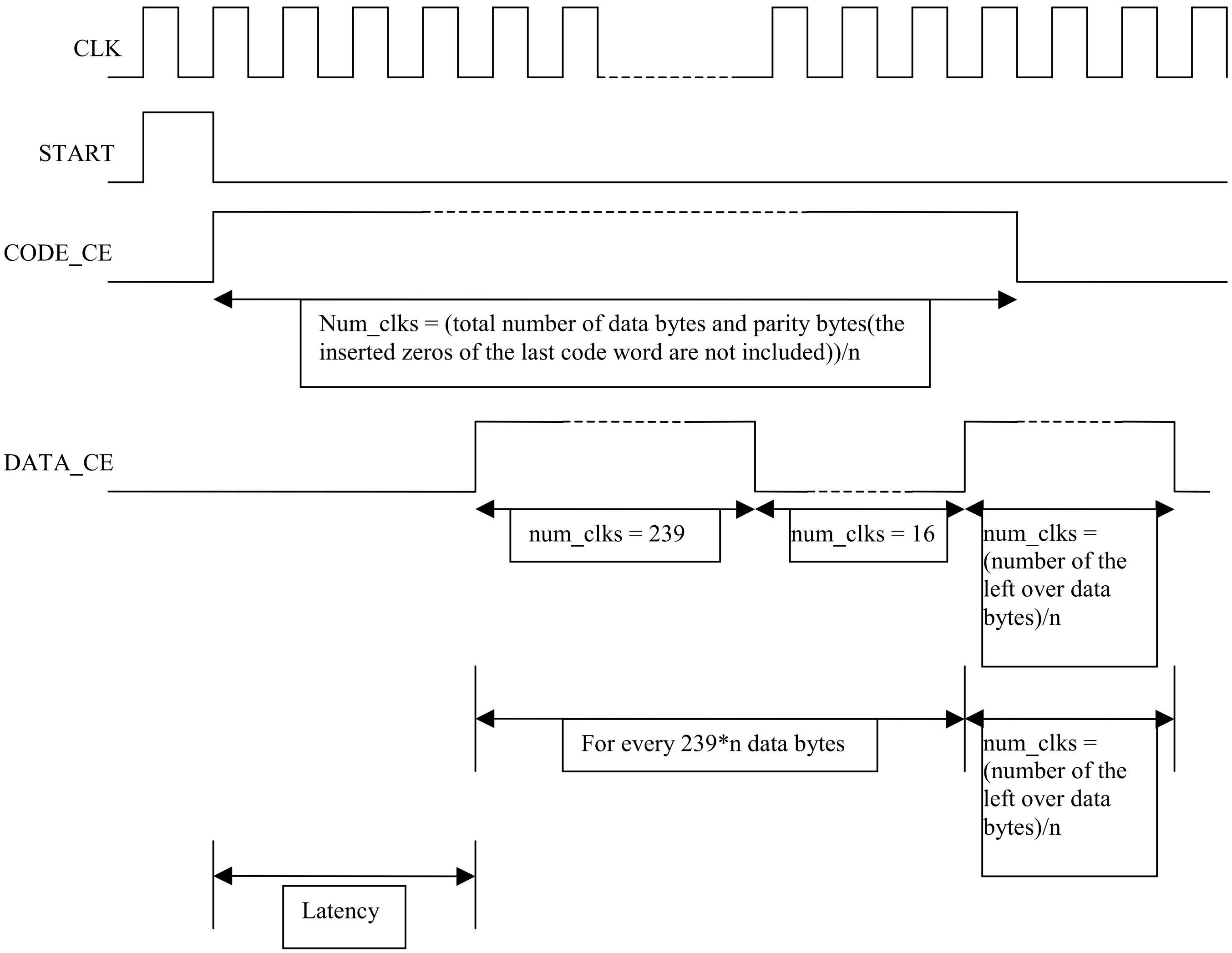
CLK
START
CODE_CE
Num_clks = (total number of data bytes and parity bytes(the inserted zeros of the last code word are not included))/n
DATA_CE
num_clks = 239
num_clks = 16
num_clks = (number of the left over data bytes)/n
For every 239*n data bytes
num_clks = (number of the left over data bytes)/n
Latency

Figure 3. Timing Diagram of the GPON FEC decoder

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/