

## 3GPP Turbo Convolutional Code Encoder

### Features

- Fully compliant with 3GPP (UMTS, WCDMA) specification. Supports data block length 40 to 5114.
- Available for Xilinx FPGA or ASIC implementation
- Compact design, uses 561 CLB slices, 4 Block Rams, and 3 Multipliers in Virtex II, or 568 CLB slices, 4 Block Rams, and 3 multipliers in Spartan III
- Clock speed reaches 125 MHz in Virtex II or 105 MHz Spartan III
- Fully synchronous one clock design

### Functional Description

The block diagram of the 3GPP Turbo Convolutional Code encoder is shown in Figure 1. The encoder is composed of two parallel concatenated convolutional encoders as constituent encoders and an internal interleaver. The data input is fed into the first constituent encoder in natural order

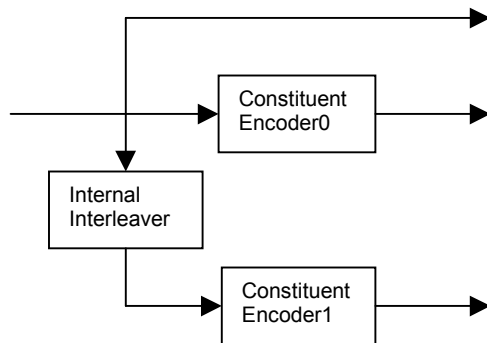


Figure 1. Block diagram of the Turbo Convolutional Code Encoder for 3GPP.

and at the same time the second one in an interleaved order. The output includes the data sequence in natural order and the two parity sequences generated by the two constituent encoders.

Because of the existence of the interleaver, Turbo convolutional code is a block code.

When started, the encoder will first load the whole data sequence into an input buffer and then generate an interleaved data sequence. The parity sequences from the two constituent encoders and the original data sequence will be stored in an output buffer and wait for the signals to read them out.

### Pin Out

Figure 2 is the schematic symbol of the Turbo convolutional code encoder for 3GPP. The pin out is described as follows.

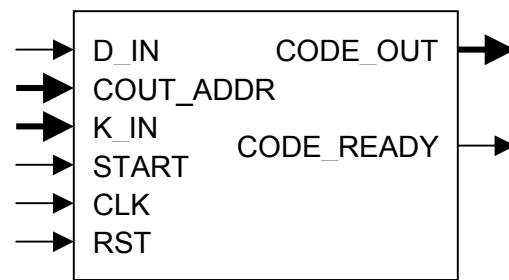


Figure 2. Schematic symbol of the Turbo convolutional code encoder

#### 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.

#### 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. The length of START should be one clock cycle.

#### K\_IN

Thirteen bit input, the length of the data sequence to be encoded. K\_IN is in the range of 40 to 5114.

**COUT\_ADDR**

Thirteen bit input, the address to read out the code sequence from the output buffer. The code sequence can be read out at any rate. The speed is controlled by COUT\_ADDR. In addition to the K\_IN code symbols, there are six additional tail symbols. The output code symbol is one clock after the COUT\_ADDR.

**D\_IN**

One bit input, the data sequence to be encoded. The first bit of the data sequence should be one clock after the START pulse. The encoder reads in one bit every clock.

**CODE\_OUT**

Three bit output, the output code symbol. The MSB of the code symbol is the data bit. The LSB of the code symbol is the parity bit from the second constituent encoder. The middle bit is the parity bit from the first constituent encoder. The LSB of the first three tail symbols and the middle bit of the last three tail symbols are dummy bits. Additional logic will be needed to output the code sequence in the order defined in the 3GPP specification.

**CODE\_READY**

One bit output. When CODE\_READY is high the code sequence is ready for output. CODE\_RAEDY will stay high from the moment encoding is done until the start of the next encoding process.

**Timing Diagrams**

The 3GPP Turbo Convolutional Code 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 bit of the data sequence.

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

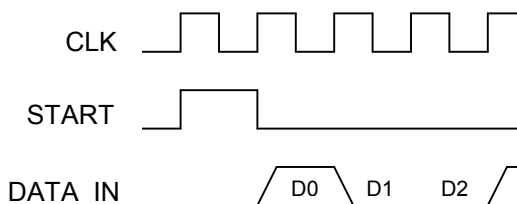


Figure 3. The timing diagram at the starting point of the encoding process

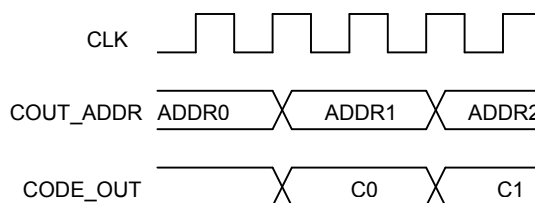


Figure 4. The timing diagram at the starting point of the output

COUT\_ADDR can change at any rate.

**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 contact our local distributors or use the following information to contact us directly:

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](mailto:sales@highlandcomm.com)

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