3GPP Turbo Convolutional Code Decoder

Features

- **Fully compliant with 3GPP (UMTS, WCDMA) specification. Support data block length 40 to 5114**
- **Available for Xilinx FPGA or ASIC**
- **Compact design, uses 1836 CLB slices, 18 Block Rams, and 3 multipliers in Virtex II, or 1865 CLB slices, 18 Block Rams, and 3 multipliers in Spartan III, among the smallest on the market**
- **Clock speed reaches 75 MHz in Virtex II, supports 2 Mbps data rate at 8 iterations or 4 Mbps data rate at 4 iterations. Clock speed reaches 60 MHz in Spartan III, supports 2 Mbps data rate at 6 iterations or 4 Mbps data rate at 3 iterations;**
- **No external noise power estimation**
- **4 bit soft input for data and parity**
- **Early termination for power efficiency**
- **Power efficient interleaving operation**
- **Fully synchronous one clock design**

Functional Description

The block diagram of the 3GPP Turbo Convolutional Code decoder is shown in Figure 1. The constituent decoders use the Max Log MAP algorithm to decode the constituent code sequences. The extrinsic information from one constituent decoder is used as the a priori information to the other decoder. The two constituent decoders work iteratively to improve the performance of the decoder from iteration to iteration until the optimal performance is achieved or the total number of iterations is reached.

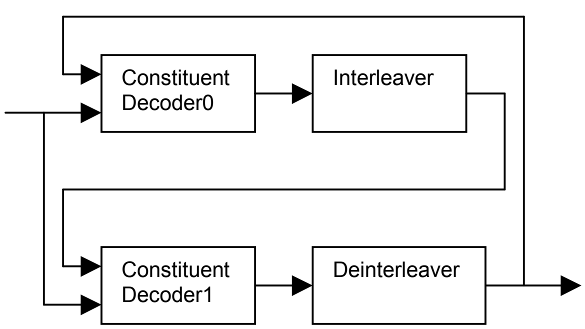

Figure 1. Block diagram of the 3GPP Turbo Convolutional Code decoder.

Pin Out

Figure 2 is the schematic symbol of the 3GPP Turbo Convolutional Code decoder. The pin out is described as follows.

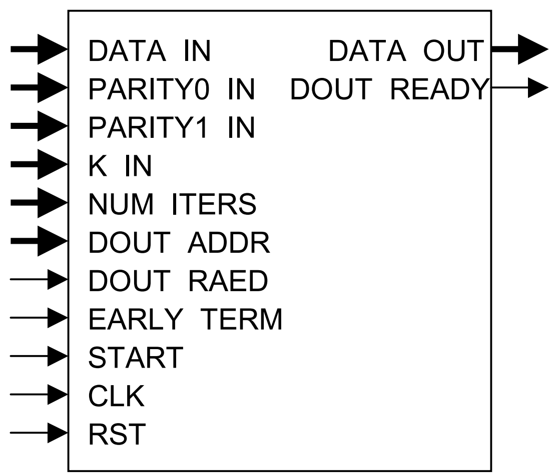

Figure 2. Schematic symbol of the decoder

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 decoding process. At the rising edge of CLK, if START is high while RST is low, the internal state machine will start the decoding

Copyright © 2004 Highland Communications Technologies

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

EARLY_TERM
One bit input, the signal to enable the early termination mechanism. If EARLY_TERM is high, the decoding process will stop when the optimal performance is achieved or the total number of iterations is reached. If EARLY_TERM is low the decoding process will stop when the total number of iterations is reached.

DOUT_READ
One bit input, the signal to enable the readout of the decoded data. When DOUT_READ is high, DATA_OUT is the decoded data.

DOUT_ADDR
Thirteen bit input, the address of the output data. The output data bit is available one clock after DOUT_ADDR. DOUT_ADDR can change at any rate.

NUM_ITERS
Four bit input, the signal to specify the number of iterations. NUM_ITERS specifies the total number of HALF iterations. It can take any value from 1 to 15. For a 2 Mbps data rate, the highest number of iterations supported is 6. To specify 6 iterations, NUM_ITERS needs to be set to 12. The early termination mechanism will stop the decoding process at half iterations if the optimal performance is reached at half iterations.

K_IN
Thirteen bit input, the length of the data sequence to be decoded. K_IN is in the range of 40 to 5114.

DATA_IN
Four bit input, the received data sequence. The first symbol should be one clock after the START pulse. The decoder reads one symbol every clock.

PARITY0_IN
Four bit input, the first received parity sequence. The first symbol should be one clock after the START pulse. The decoder reads one symbol every clock.

PARITY1_IN
Four bit input, the second received parity sequence. The symbol should be one clock after the START pulse. The decoder reads one symbol every clock.

DATA_OUT
One bit output, the decoded data sequence.

DOUT_READY
One bit output. After the decoding process is stopped, DOUT_READY is set high. This signal that the decoded data sequence is ready for readout. The START signal will bring DOUT_RAEDY to low.

Timing Diagrams

The 3GPP Turbo Convolutional Code decoder is very easy to be integrated into a larger design. The following timing diagrams help to clarify some of the synchronization issues.

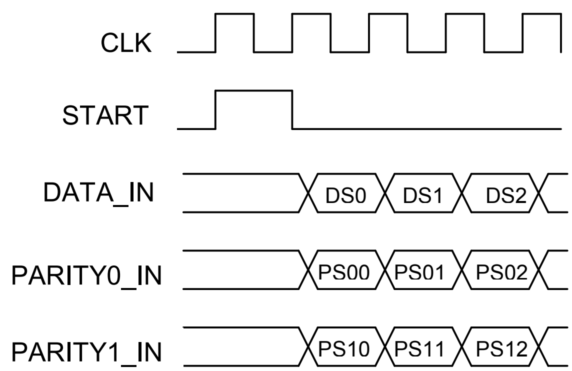

Figure 3. The timing diagram at the starting point of the encoding process

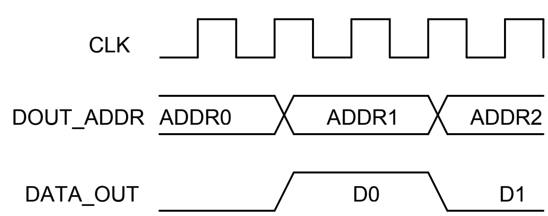

Figure 4. The timing diagram at the starting point of the output

Copyright © 2004 Highland Communications Technologies

Figure 3 shows the timing diagram at the starting point of the decoding process, where DS0 is the first symbol of the received data sequence, PS00 is the first symbol of the first received parity sequence, and PS10 is the first symbol of the second received parity sequence.

Figure 4 shows the timing diagram at the starting point of the output, where D0 is the first bit of the decoded data sequence. DOUT_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 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/

Copyright © 2004 Highland Communications Technologies