Turbo Encoder/Decoder MegaCore Function

Solution Brief 50 **September 2000, ver. 1.0**

Target Applications:
3G Wireless Systems, Satellite Communications

Family:
APEX™ 20K, APEX 20KE

Ordering Code:
PLSM-TURBO/ENC
PLSM-TURBO/DEC

Vendor:

ALTERA®

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

Features

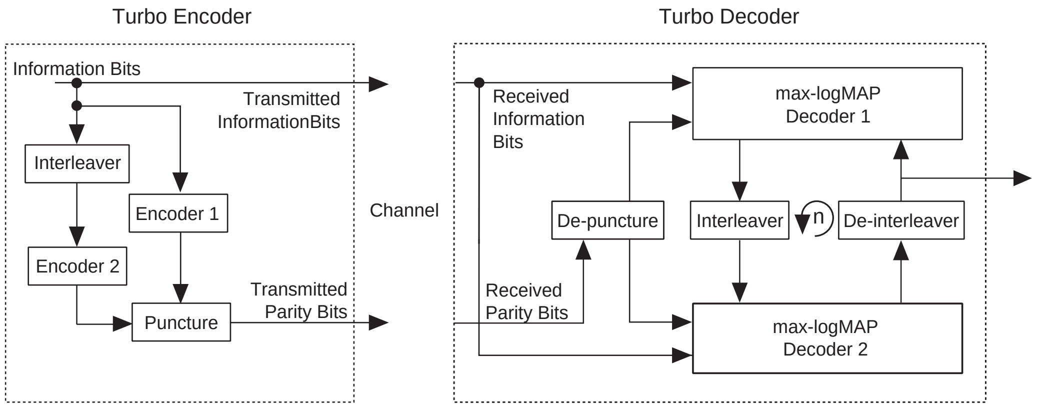
- Compliant with 3rd Generation Partnership Project (3GPP); Technical Specification Group Radio Access Network; Multiplexing and Channel Coding (FDD) (3G TS 25.212, version 3.1.0)
- High-performance max-logMAP (logarithmic 'maximum a posteriori') decoder for maximum error correction
- Data rates in excess of 2 megabits per second (Mbps)
- Includes 3GPP-compliant mother interleaver
- Interleaver block sizes from 40 to 5,114 bits
- Block size can change between each block
- Soft values (logarithmic likelihood) from 3 to 8 bits
- Optional two memory banks for maximum throughput
- Optimized for the Altera® APEX™ 20K and APEX 20KE architectures
- MegaWizard® Plug-In for easy parameterization

General Description

In wireless communication, data is sent through the air. During this process, the data signals are exposed to noise, and the data can be partly destroyed. The Altera Turbo Encoder efficiently adds check sums and parity bits, enabling the Turbo Decoder to find and correct errors, and reconstruct the destroyed data.

Figure 1 shows a basic block diagram of the turbo encoder/decoder function.

Figure 1. Turbo Encoder/Decoder Block Diagram



Turbo Encoder

The Altera Turbo Encoder MegaCore® function has two encoders that use a stream-driven implementation that feeds the incoming information bits through to the output. Encoder 1, a recursive convolutional encoder, encodes information bits, then feeds the information bits through a pseudo-random interleaver into encoder 2. These two encoders generate two completely different data sequences which are affected differently by errors. The different sequences allow the turbo decoder to find and correct data errors more accurately. To save bandwidth, encoded bit streams can be punctured, sending only every other parity bit.

Turbo Decoder

After depuncturing the received data stream, the information bits and parity 1 bits are fed into decoder 1. The decoder delivers probabilities of the received values, indicating that likelihood that a bit is correct. Decoder 1 then evaluates these probabilities and combines them with the parity 1 probabilities, refining the soft information so the confidence of the individual bit correctness is maximized. The refined probabilities are fed into decoder 2, again producing enhanced soft information. When data is exchanged between the two decoders, the soft values are reordered with the interleaver and de-interleaver to match the interleaving structure.

Interleavers

The turbo encoder interleaver is a 3-stage interleaver which makes encoder sequence 1 different from encoder sequence 2, so that errors during transmission effect the sequences differently. The input sequence is first written row-by-row into a matrix. The rows are algebraically interleaved based on sets of prime integers, then interleaved with a predefined pattern. The output sequence is generated by reading out the matrix column-by-column.

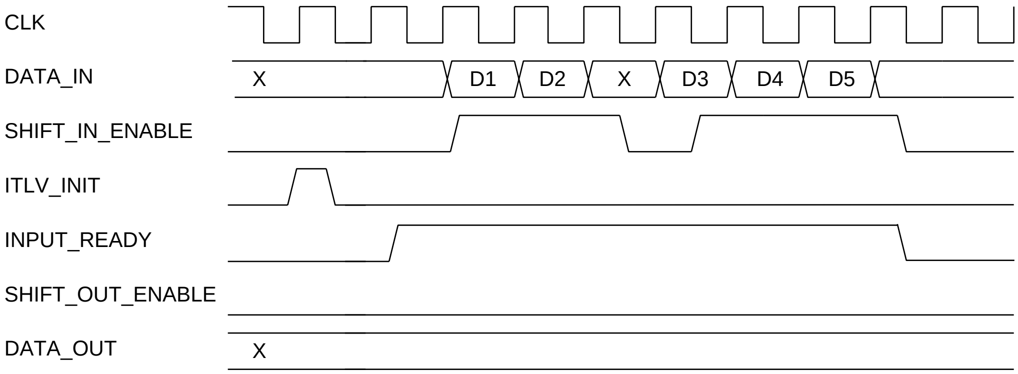
Functional Descriptions

The turbo encoder is a slave device that operates with two control input signals: `SHIFT_IN_ENABLE` and `SHIFT_OUT_ENABLE`. The process of encoding a block of data can be broken down into the following four phases: shift-in, active, shift-out, and finished. This section describes each phase.

The Shift-In Phase

Shifting-in can only commence when `INPUT_READY` is high. The data at `DATA_IN` must be valid when `SHIFT_IN_ENABLE` is high because it is registered on the next rising clock edge. Figure 2 shows the shift-in phase timing diagram.

Figure 2. The Shift-In Phase Timing Diagram



The Active Phase

The encoder will automatically go into the active phase once it detects that the required number of bits have been shifted-in. During this phase, the encoder interleaves and encodes the information, generates the tail bits, and prepares itself to produce punctured output data. When the active phase is over, the encoder will automatically enter the shift-out phase.

The Shift-Out Phase

During the shift-out phase, OUTPUT_READY goes from low to high, and output is registered. SHIFT_OUT_ENABLE can be asserted low at any time to create a pause in the output stream. When all of the punctured data has been read from the encoder, the encoder enters the finished phase.

The Finished Phase

In the finished phase, the encoder prepares itself for the next block of data by clearing all registers. One clock cycle later, the encoder will return to the shift-in phase, ready to accept the next block of data.

Performance

The max-logMAP decoder requires two clock cycles to decode each bit, plus a few cycles to fill the pipeline at the start of each decoding block. The max-logMAP decoder operates twice for each iteration of the turbo decoder and once for each set of parity bits. Therefore, each iteration of the turbo decoder requires four clock cycles per information bit. The maximum clock frequency of the turbo decoder is about 50 MHz, depending on the parameters selected. The performance of the turbo decoder is determined by the number of iterations required. For example, with five iterations, 20 clock cycles will be required per sample. Allowing an overhead of 5 cycles per sample gives 25 cycles per bit. At a clock rate of 50 MHz, this provides a bit rate of 2 Mbps. To achieve higher throughput, use several turbo decoders in parallel. The decoding of each block is totally independent of all other blocks.

Size and Typical Configurations

The amount of logic needed for the turbo decoder MegaCore function is about 5,000 to 6,000 logic elements (LEs), which fit on an APEX EP20K200 device. A typical configuration uses 5 bits to represent the soft decision values. The parity memory is on-chip, and the alpha matrix memory is off-chip. The information likelihood and apriori memories are 5,000 × 5, and occupy 13 embedded system blocks (ESBs) each. The parity memory occupies 26 ESBs. The total on-chip memory requirement is 58 ESBs, and an APEX EP20K300E is a suitable device. If the parity memory is implemented off-chip, the total ESB count is 32 , and the decoder would fit in an APEX EP20K200E device. Table 1 shows other example configurations.

Table 1. Configuration Examples

SOFTBITS	BANKSWAP	Alpha Memory	Parity Memory	ESB Count	Suitable Device
3	0	On-chip	On-chip	75	EP20K400
3	1	On-chip	On-chip	97	EP20K400
4	0	Off-chip	On-chip	44	EP20K200
4	1	Off-chip	On-chip	74	EP20K400
5	0	Off-chip	Off-chip	30	EP20K200
5	1	Off-chip	Off-chip	42	EP20K200
5	0	Off-chip	On-chip	55	EP20K300E
5	1	Off-chip	On-chip	92	EP20K400
5	0	On-chip	On-chip	135	EP20K600E
8	0	Off-chip	Off-chip	44	EP20K200
8	1	Off-chip	On-chip	64	EP20K300E
8	0	Off-chip	On-chip	84	EP20K400
8	1	Off-chip	Off-chip	144	EP20K600E

Generating a Custom MegaCore Function

Altera provides a MegaWizard Plug-In Manager with the RS Compiler MegaCore function. You can use the MegaWizard Plug-In Manager within the MAX+PLUS® II or Quartus™ software, or as a standalone application to create and integrate custom megafunctions without changing your design's source code. You can then simulate your design to verify compatibility and instantiate the custom megafunction in your design file.

Start the MegaWizard Plug-In Manager by choosing the **MegaWizard Plug-In Manager** command in any MAX+PLUS II or Quartus application, or type the command `megawiz` at a command or UNIX prompt. Specify that you wish to create a new custom megafunction, and select **Turbo Codec** from the **Communications** folder. The first page of the wizard enables you to choose to create either a decoder or an encoder. If you select decoder, you must select your required parameters, including the number of bits for soft information, and number of clock cycles for alpha and parity memory access.

Altera provides VHDL models that you can use to simulate the functionality of the Turbo Encoder/Decoder MegaCore function in your system. Users who have licensed the MegaCore function can also generate VHDL Output Files (**.vho**) or Verilog Output Files (.**vo**) for simulation in third-party simulators.

To learn more about the Altera Turbo Encoder /Decoder MegaCore function, refer to the *Turbo Encoder/Decoder MegaCore Function User Guide.*

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