# Turbo Decoder

**User's Guide**

## Introduction

Lattice's Turbo Decoder core provides an ideal solution that meets the needs of turbo decoding applications. The core provides a customizable solution allowing turbo decoding of data in many system designs. This core allows designers to focus on the application rather than the Turbo Decoder, resulting in a faster time to market.

Turbo coding is an advanced error correction technique widely used in the communications industry. The Turbo Decoder IP Core from Lattice is compliant with three different standards: 3GPP, 3GPP2 and CCSDS. Lattice's Turbo Decoder core was developed in conjunction with Lattice's Turbo Encoder core to provide a complete solution.

This User's Guide explains the functionality of the Turbo Decoder core and how it can be implemented to provide decoding. The Turbo Decoder core comes with the documentation and files listed below:

- Lattice gate level netlist

- RTL simulation model

- Core instantiation template

## Features

- Fully compatible with Third Generation Partnership Project (3GPP) standard:
  – 3GPP TS 25.212 version 4.2.0

- Fully compatible with CDMA2000/3GPP2
  – 3GPP2 C.S002-C, May 2002

- Fully compatible with Consultative Committee for Space Data Systems standard:
  – CCSDS 101.0-B-5

- Throughput of 2Mbps for 3GPP at 30MHz, 7 iterations, 6-bit input symbol width

- Two's complement data/parity input

- Depuncturing supported

- Variable soft-widths for input symbols

- User-defined number of states

- Variable block sizes during runtime

- Programmable number of iterations (1-15)

- Optional hard decision storage

- Selectable Max-Log-Map or Log-Map algorithm

- Optional external memory with programmable pipeline stages

- Optional double buffering

- Bit Error Rate of $10^{-6}$ (at 1.5 dB Eb/No SNR)

## General Description

Turbo coding is an advanced error correction technique widely used in the communications industry. Turbo encoders and decoders are key elements in today's communication systems to achieve the best possible data reception with fewest possible errors. The basis of turbo coding is to introduce redundancy in the data to be transmitted through a channel. The redundant data helps to recover original data from the received data. In data transmission, turbo coding helps achieve near Shannon limit performance.

Lattice provides a Turbo Decoder IP core that is both flexible and compliant with three different standards: 3GPP, 3GPP2 and CCSDS. 3GPP is widely used in WCDMA and MC-CDMA applications while CCSDS is most commonly found in telemetry and space communications. Figure 1 shows the top-level block diagram of this core.

Lattice also supplies a Turbo Encoder core that provides a complete state-of-the-art error correction solution.

*Figure 1. Turbo Decoder I/O Block Diagram*
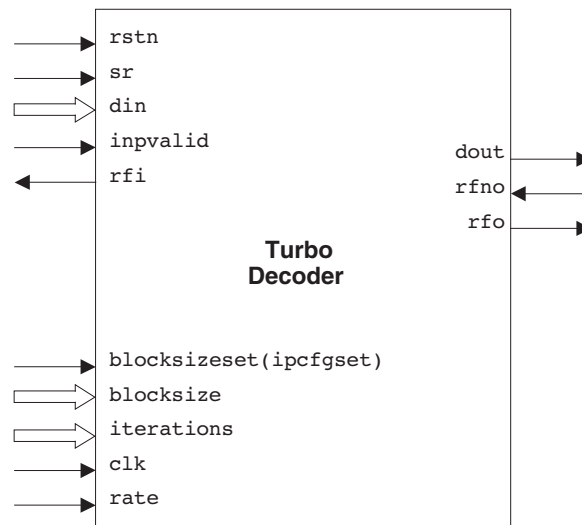


Note: Additional I/O signals are required if either an external memory or double buffer is selected. Please refer to the Additional Signals for External Memory section of this document for further information.

## MAP Algorithm

Turbo decoding is based on the principle of comparing the probability of a received soft input data being a '1' and '0'. The Lattice Turbo Decoder uses a decoding scheme called the MAP - Maximum Aposteriori Probability algorithm. The algorithm determines the probability of whether each received data symbol is a '1' as well as '0'. This is done with the help of the data, parity symbols, and the decoder knowledge of the encoder trellis. A trellis is a form of a state transition table of the encoder input/output. Based on the data and parity information, the MAP decoder computes the probability of the encoder being in a particular state. Depending on the soft data, parity value and the weight from the previous state, the probability that the data is a '1' or '0' can be computed. The MAP decoder computes the weight for each data symbol in a given block for both the forward and reverse direction. This results in the computation of a forward and reverse metric. Using these two values, the probabilities are computed. After the probabilities are determined, they are compared and a decision is made. The Lattice Turbo Decoder IP core uses the logarithm of the probability to reduce computation; this is known as Log Likelihood ratio (LLR). The computation of the probabilities is done iteratively to obtain a reliable result. Once the result is considered reliable, one can make a final decision as to whether the data symbol is a '1' or a '0'. The Lattice Turbo Decoder can implement both the Log-Map and Max-Log-Map algorithm. The Log-Map algorithm gives a slightly better performance than the Max-Log-Map but utilizes more resources and runs at a slower frequency.

The Log Likelihood ratio is the probability that the received data bit is a '0' divided by the probability that the received data bit is a '1'.

$$L(D) = \log \frac{P\,(D=0)}{P\,(D=1)}$$

The value of L(D) is positive if $P(D=1) \leq P(D=)$ and negative otherwise. The output data value is '1' if L(D) is positive and '0' if L(D) is negative. For one complete cycle of iteration, one needs to compute the LLR using parity for non-interleaved as well as interleaved data.

# Block Diagram

Figure 2 shows a block diagram of the Turbo Decoder detailing the key components and the data paths between these blocks.

*Figure 2. Turbo Decoder Functional Block Diagram*
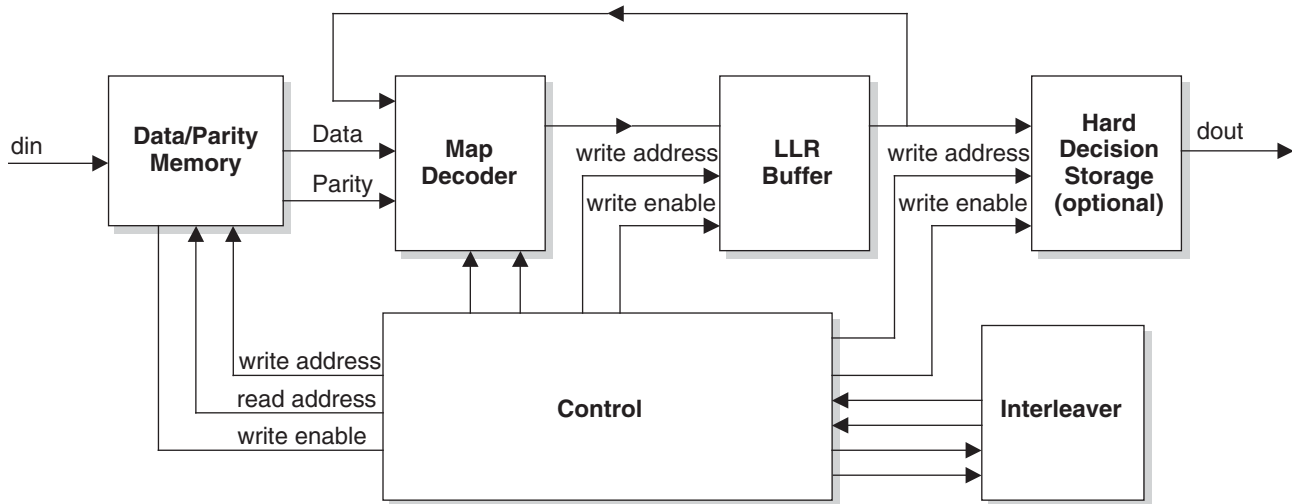


# Functional Description

The Turbo Decoder consists of four main components: control module, decoder, interleaver and memory buffers.

## Control Module

The control module takes care of the interface, pipelining and handshake communication between various blocks and I/O pins. Data and parity are read serially into the memory and it is assumed that the data is received in the same order as it was transmitted from the encoder. Signal `blocksizeset(ipcfgset)` initializes the `blocksize` by specifying the size of the block to be input to the decoder. Input data can be given only when `rfi` is asserted. Input data has to be qualified with `inpvalid` to be accepted by the core.

## Decoder and Interleaver

Once the data is entered into the decoder, the decoder starts computing the LLR of each data symbol. The LLR is computed for the block sequence twice, once using the non-interleaved data and the corresponding parity and then using the interleaved data and the corresponding parity. One round of this computation is called an iteration. Each iteration is divided into two sections, an ODD window and an EVEN window. The LLR for systematic parity is computed during the ODD window and the LLR for interleaved data is computed during the EVEN window. When both ODD and EVEN window computations are done, one iteration is complete. The user can set the number of iterations for each block on the iterations pin. During the second half of the iteration, EVEN window, the LLR computed in the first half is improved upon by using previous computations. Every window makes use of LLR information computed in the previous window and tries to improve on the estimate of LLR. The interleaver is used in the second half of iteration to generate the interleaved address. This address is used to address the data and parity memory to read the interleaved data for the second half of the iteration. It is also used to address the LLR memory unit and get the previously computed LLR information. At the end of one iteration, the decoder has a set of LLR for each input data. The second iteration starts with again using the non-interleaved parity bits and data and the previously computed LLR to get a new estimate of the LLR for the data. Once the decoder completes the number of iterations required to be done, the LLR memory buffer has the final LLR values. The sign of the LLR values determines whether the data is a '1' or a '0'. A positive sign means the data value is a '1', otherwise it is a '0'.

## Output Data Handshaking

When the decoder is ready to output data, signal `rfo` is asserted high after the decoder has completed the specified number of iterations. The user can then assert signal `rfno` to read the decoded data, which then allows data to be output on `dout`.

A synchronous reset signal, `sr` is available to reinitialize the Turbo Decoder in the middle of a block processing. The current block being processed will be completely discarded during this reset. This can be done at any point of time during the operation

## Memory Buffer

The memory buffering for this IP splits into four sections. These sections are described in detail below.

### Input Data/Parity Memory

The Turbo Decoder core requires a large amount of memory to store the input data block. Since data memory requirements are large, an external memory is recommended so that on-chip memory can be used for other purposes. An external memory interface is provided in the IP. A single or double buffer memory mode may be selected depending on the available external memory at hand. Double buffer memory allows one block of data to be processed while another block is written and read. Double buffer memory delivers better performance than the single buffer selection by minimizing delay between the processing of each block.

### Internal Memory

Some internal memory is required to implement the interleaver and other necessary functions of the Turbo Decoder. Lattice's Turbo Decoder requires a small amount of memory for internal purposes. For example, the 3GPP configuration uses 4.6Kb spread over four memory blocks.

### LLR Memory

After the Turbo Decoder completes the required number of iterations, the LLR memory buffer stores the final LLR values. The size of the LLR memory buffer is dependant on configuration and block size.

### Hard Decision Storage Memory

The Turbo Decoder IP core offers optional hard decision storage. When LLR memory is used as an output buffer, the decoder cannot go onto process the next block of data until current LLR values of the previous block are completely read out. This results in an extra processing delay of B cycles (B = `blocksize`). To minimize delay, output data after hard decision can be stored in separate memory to allow the decoder to operate on a new data block if memory can be spared.

### Operational Data Flow

The following flow diagram describes the sequence for every block introduced into the Turbo Decoder core.

**Figure 3. Turbo Decoder Data Flow**
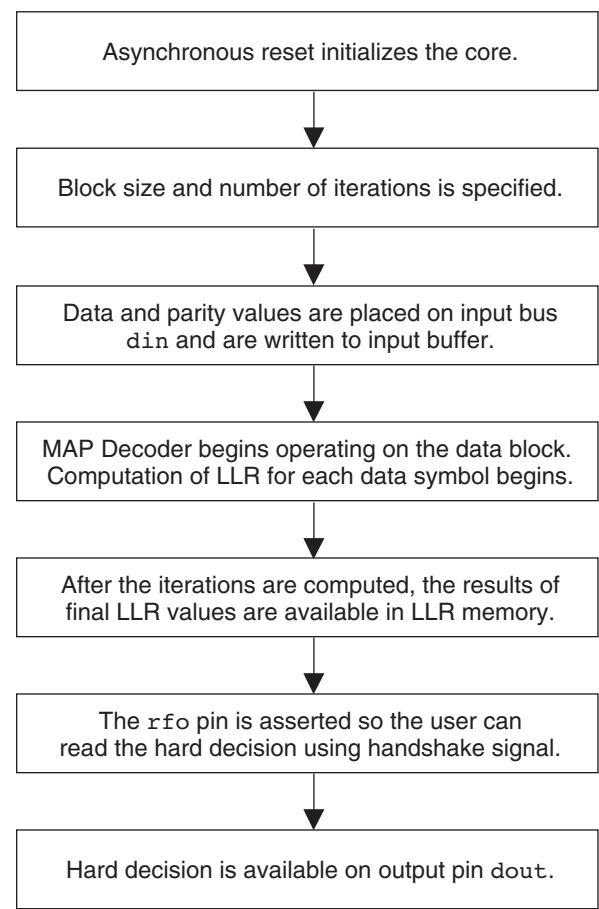


Figure 4 shows the loading of block size and iterations for the current input block with assertion of input signal `blocksizeset(ipcfgset)` after asynchronous reset `rstn` has been de-asserted. After the reset assertion, the Turbo Decoder asserts `rfi` to indicate that it can receive a data block. At the same time the decoder de-asserts `rfo` signal to show that there is no data available to give any hard decisions at the output.

**Figure 4. Initialization and Data Input to Decoder**
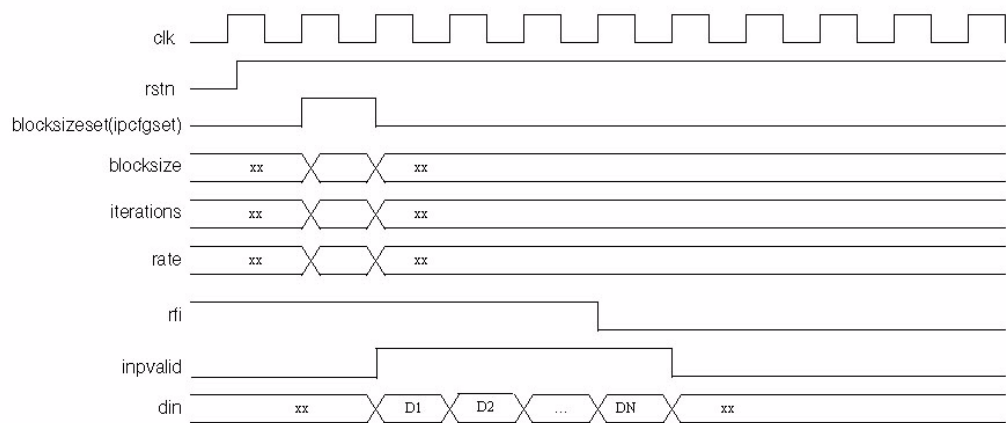
Note: Signals are transferred on the rising edge. N is the block size.

Figure 5 shows the timing for data output. Once the Turbo Decoder has completed processing the data for a given number of iterations it is ready to output the hard decision data. The decoder indicates this by asserting signal `rfo`. Once the user detects this signal, `rfno` can be asserted to indicate that the output data can be received on the output port `dout`. Thus for every `rfno` asserted the decoder will output the hard decision on the output port `dout`. Once the decoder has output all data it will lower the `rfo` signal.
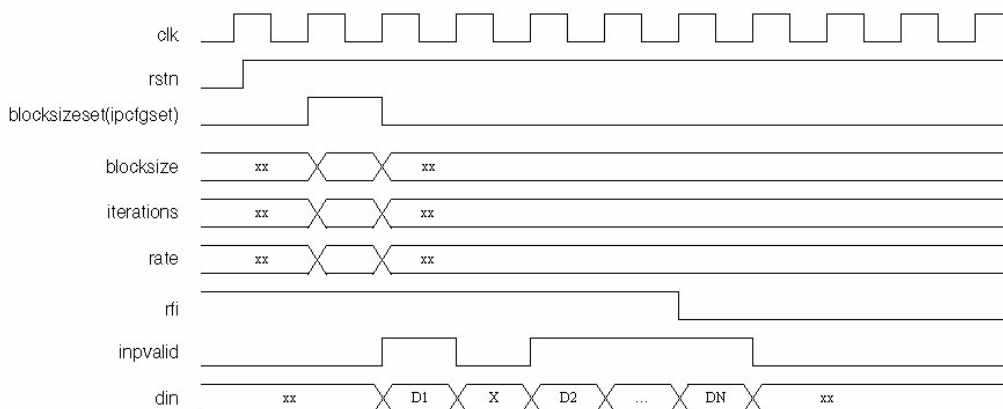
*Figure 5. Decoder Output When Processing is Done*
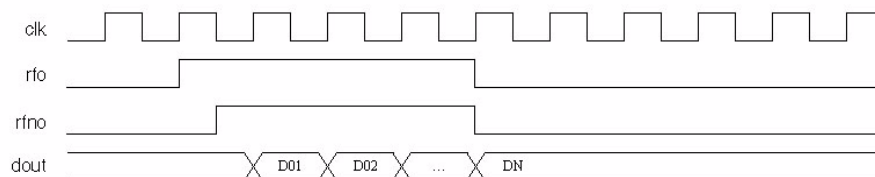


Note: Signals are transferred on the rising edge. N is the block size.

Figure 6 illustrates how `inpvalid` can be used to control the flow of data into the decoder. The input data is received in conjunction with input signal `inpvalid`. When `inpvalid` is asserted the input data on port `din` is taken as valid data and written into the input data memory buffer. When the input port `inpvalid` is de-asserted, the Turbo Decoder core ignores the value on `din`.

*Figure 6. Input Data Handshaking with inpvalid Going Low*



Note: Signals are transferred on the rising edge. N is the block size.

Figure 7 shows how `rfno` can be used to control the flow of data from the Turbo Decoder core. The output hand-shake signals are `rfo` and `rfno`. Once the decoder is ready to give out the hard decision data it asserts `rfo`. The receiving system will then assert the `rfno` signal when it is ready to receive the data. The decoder now puts the hard decision data value on the port `dout`. When the decoder has output the last data it lowers `rfo` to indicate that no more hard decision data is available in the decoder.

*Figure 7. Output Handshake with rfno Signal Going Low*
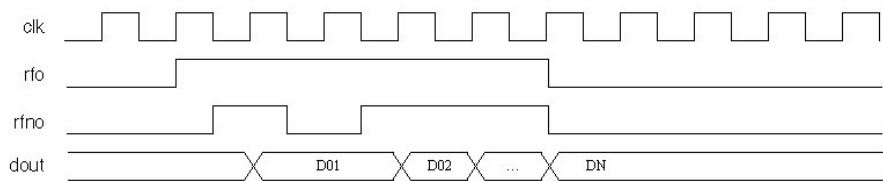


Note: Signals are transferred on the rising edge. N is the block size.

When the decoder is processing data it will lower `rfi` if it cannot receive another data block. This occurs in two different scenarios. If only a single buffer has been chosen then the decoder will de-assert the `rfi` signal as soon as an input data block is read in completely. In case of double buffer memory the decoder will lower `rfi` when it has two blocks in memory and one of them is still being processed. Once the decoder completes processing the block, it will assert both `rfo` and `rfi`. As shown in Figure 8, the average processing delay for the Turbo Decoder in the case of 3GPP is 2.13xBxI, where B is the block size and I is the number of iterations.

*Figure 8. Decoder State During Data Processing*
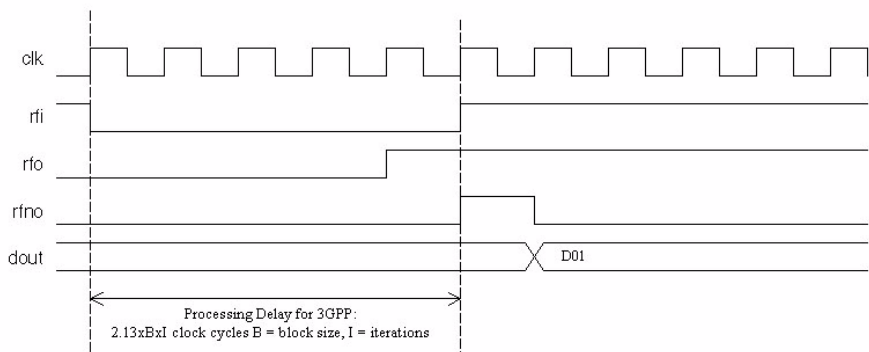


Note: Signals are transferred on the rising edge.

# Parameter Descriptions

User-configurable parameters for each standard are shown in Table 1.

*Table 1. User-Configurable Parameters*

| Parameter Name | 3GPP | 3GPP2[1] | CCSDS |
|---|---|---|---|
| Decoder type | 3GPP | 3GPP2 | CCSDS |
| Number of states | 8 | 8 | 16 |
| Maximum block size | Default: 5114 | Default: 20730 | Default: 8920 |
| Input data width | Range: 3-6<br>Default: 6 | Range: 3-6<br>Default: 6 | Range: 3-6<br>Default: 6 |
| Window size | Range: 16 or 32<br>Default: 16 | Range: 16 or 32<br>Default: 16 | Range: 16 or 32<br>Default: 16 |
| External memory | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes |
| Double buffer | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes |
| Decoder algorithm | Max Log Map (Default)<br>Log Map | Max Log Map (Default)<br>Log Map | Max Log Map (Default)<br>Log Map |
| External memory pipeline | Range: 1-6<br>Default: 1 | Range: 1-6<br>Default: 1 | Range: 1-6<br>Default: 1 |
| Hard decision storage | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes | Range: Yes or No<br>Default: Yes |

1. 3GPP2 type is currently available only for LatticeEC™ and LatticeECP™ device families.

# Signal Descriptions

Table 2 shows the signal descriptions for the input and output ports of the Turbo Decoder IP core.

*Table 2. Turbo Decoder Signal Definitions*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| `clk` | Input | 1 | System Clock |
| `rstn` | Input | 1 | Active Low Asynchronous Reset |
| `sr` | Input | 1 | Synchronous Reset |
| `din` | Input | 3-6 | Data Input (soft encoded data from demodulator) |
| `dout` | Output | 1 | Data Output |
| `iterations` | Input | 4 | Input bus to set the number of decoding iterations. The valid values are from 1 to 15. |
| `blocksiz-eset(ipcfgset)` | Input | 1 | Interleaver initialization. `blocksize` on input pins can be changed and accepted when this signal is asserted. |
| `inpvalid` | Input | 1 | Enables the decoder to read the data at `din` when asserted. |
| `rfi` | Output | 1 | Active High signal. Indicates that decoder is available to accept data. This signal is de-asserted one data symbol before buffer is full. |
| `blocksize` | Input | 13-15 | Block size up to 20730 bits can be set depending on the configuration selected. Block size ranges:<br>3GPP: 40-5114<br>3GPP2: 378-20730<br>CCSDS: 1784, 3568, 7136, 8920 |
| `rfno` | Input | 1 | Asserted to indicate successful reading of encoded data from `dout`. |
| `rfo` | Output | 1 | When asserted encoded data is ready and available at `dout`. |
| `rate` | Input | 1 or 2 | Determines the rate of the Turbo Decoder<br>For 3GPP, size 1 bit, 0-rate 1/3 and 1- rate 1/2<br>For 3GPP2, size 2 bits, 0-rate 1/2, 1- rate 1/3, 2- rate 1/4 and 3- rate 1/5<br>For CCSDS, size 2 bits, 0-rate 1/2, 1- rate 1/3, 2- rate 1/4 and 3- rate 1/6 |

## Additional Signals for External Memory

When external memory is used with this core additional signals are provided to form the interface to the external memory. These are detailed below.

### 3GPP

In the case where external memory is selected, the I/O pins in Table 3 will be added to the block for exchanging data with the memory in the case of 3GPP. It is assumed that data and parity are stored in different memory buffers. Non-interleaved and interleaved parity are stored in different buffers.

*Table 3. Additional I/Os Due to External Memory for 3GPP*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_dat_buf1 | Input | 3-6 | Information data port 1 |
| g2_dat_buf1 | Input | 3-6 | Information data port 2 |
| g1_par_odd1 | Input | 3-6 | Parity 1 (systematic) data port 1 |
| g2_par_odd1 | Input | 3-6 | Parity 1 (systematic) data port 2 |
| g1_par_even1 | Input | 3-6 | Parity 2 (interleaved) data port 1 |
| g2_par_even1 | Input | 3-6 | Parity 2 (interleaved) data port 2 |
| data_to_mem | Output | 3-6 | Information/parity data to memory |
| data_waddr | Output | 3-6 | Information/parity Write address |
| wren_dat_buf1 | Output | 1 | Write enable for Information data |
| wren_par1_buf1 | Output | 1 | Write enable for parity 1 (systematic) |
| wren_par2_buf1 | Output | 1 | Write enable for parity 2 (interleaved) |
| g1_rden | Output | 1 | Information/parity read enable port 1 |
| g2_rden | Output | 1 | Information/parity read enable port 2 |
| g1_dat_raddr | Output | 11-15 | Information read address port 1 |
| g2_dat_raddr | Output | 11-15 | Information read address port 2 |
| g1_par_raddr | Output | 11-15 | Parity read address port 1 |
| g2_par_raddr | Output | 11-15 | Parity read address port 2 |

In the case where double buffering is selected along with the external memory the I/O pins in Table 4 will also be added to the core for exchanging data with the second buffer in the case of 3GPP.

*Table 4. Additional I/Os Due to Double Buffering for 3GPP*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_dat_buf2 | Input | 3-6 | Information data port 1 |
| g2_dat_buf2 | Input | 3-6 | Information data port 2 |
| g1_par_odd2 | Input | 3-6 | Parity 1 (systematic) data port 1 |
| g2_par_odd2 | Input | 3-6 | Parity 1 (systematic) data port 2 |
| g1_par_even2 | Input | 3-6 | Parity 2 (interleaved) data port 1 |
| g2_par_even2 | Input | 3-6 | Parity 2 (interleaved) data port 2 |
| wren_dat_buf2 | Output | 1 | Write enable for Information data |
| wren_par1_buf2 | Output | 1 | Write enable for parity 1 (systematic) |
| wren_par2_buf2 | Output | 1 | Write enable for parity 2 (interleaved) |

## CCSDS

For CCSDS decoder type, in the case that an external memory is selected; the following additional pins are required. It is assumed that data and parity are stored in different memory buffers. Non-interleaved and interleaved parity are stored in different buffers.

*Table 5. Additional I/Os Due to External Memory for CCSDS*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_par2_odd1 | Input | 3-6 | Parity 3 (systematic) data port 1 |
| g2_par2_odd1 | Input | 3-6 | Parity 3 (systematic) data port 2 |
| g1_par3_odd1 | Input | 3-6 | Parity 4 (systematic) data port 1 |
| g2_par3_odd1 | Input | 3-6 | Parity 4 (systematic) data port 2 |
| g1_par3_even1 | Input | 3-6 | Parity 5 (interleaved) data port 1 |
| g2_par3_even1 | Input | 3-6 | Parity 5 (interleaved) data port 2 |
| wren_par3_buf1 | Output | 1 | Write enable for parity 3 (systematic) |
| wren_par4_buf1 | Output | 1 | Write enable for parity 4 (systematic) |
| wren_par5_buf1 | Output | 1 | Write enable for parity 5 (interleaved) |

Note: These ports are in addition to the ports listed for 3GPP in Table 3.

In the case where double buffer is selected along with the external memory the I/O pins in Table 6 will also be added to the core for exchanging data with the second buffer in the case of CCSDS.

*Table 6. Additional I/Os Due to Double Buffering for CCSDS*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_par2_odd2 | Input | 3-6 | Parity 3 (systematic) data port 1 |
| g2_par2_odd2 | Input | 3-6 | Parity 3 (systematic) data port 2 |
| g1_par3_odd2 | Input | 3-6 | Parity 4 (systematic) data port 1 |
| g2_par3_odd2 | Input | 3-6 | Parity 4 (systematic) data port 2 |
| g1_par3_even2 | Input | 3-6 | Parity 5 (interleaved) data port 1 |
| g2_par3_even2 | Input | 3-6 | Parity 5 (interleaved) data port 2 |
| wren_par3_buf2 | Output | 1 | Write enable for parity 3 (systematic) |
| wren_par4_buf2 | Output | 1 | Write enable for parity 4 (systematic) |
| wren_par5_buf2 | Output | 1 | Write enable for parity 5 (interleaved) |

Note: These ports are in addition to the ports listed for 3GPP in Table 4.

**3GPP2**

For the 3GPP2 decoder type, in the case that an external memory is selected; the following additional pins are required. It is assumed that data and parity are stored in different memory buffers. Non-interleaved and interleaved parity are stored in different buffers.

*Table 7. Additional I/Os Due to External Memory for 3GPP2*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_par2_odd1 | Input | 3-6 | Parity 3 (systematic) data port 1 |
| g2_par2_odd1 | Input | 3-6 | Parity 3 (systematic) data port 2 |
| g1_par2_even1 | Input | 3-6 | Parity 4 (interleaved) data port 1 |
| g2_par2_even1 | Input | 3-6 | Parity 4 (interleaved) data port 2 |
| wren_par3_buf1 | Output | 1 | Write enable for parity 3 (systematic) |
| wren_par4_buf1 | Output | 1 | Write enable for parity 4 (interleaved) |

Note: These ports are in addition to the ports listed for 3GPP in Table 3.

In the case where a double buffer is selected along with the external memory, the I/O pins in Table 8 will also be added to the core for exchanging data with the second buffer in the case of 3GPP2.

*Table 8. Additional I/Os Due to Double Buffering for 3GPP2*

| Port Name | I/O Type | Width | Signal Description |
|---|---|---|---|
| g1_par2_odd2 | Input | 3-6 | Parity 3 (systematic) data port 1 |
| g2_par2_odd2 | Input | 3-6 | Parity 3 (systematic) data port 2 |
| g1_par2_even2 | Input | 3-6 | Parity 4 (interleaved) data port 1 |
| g2_par2_even2 | Input | 3-6 | Parity 4 (interleaved) data port 2 |
| wren_par3_buf2 | Output | 1 | Write enable for parity 3 (systematic) |
| wren_par4_buf2 | Output | 1 | Write enable for parity 4 (interleaved) |

Note: These ports are in addition to the ports listed for 3GPP in Table 4.

# IPexpress™ User-Configurable Core

The Turbo Decoder core is an IPexpress User-Configurable IP core, which allows designers to configure the IP and generate netlists as well as simulation files for use in designs. The IPexpress flow also supports a hardware evaluation capability, making it possible to create versions of the IP core that operate in hardware for a limited period of time without requiring the purchase of an IP license.

To download a full evaluation version of this IP core, please go to the Lattice IP Server tab in the ispLEVER® IPexpress GUI window. All ispLeverCORE™ IP cores available for download are visible on this tab.

# References

The Lattice Turbo Decoder IP core is compliant with two standards: 3GPP and CCSDS. More information about each standard can be referenced at the following locations.

- The 3rd Generation Partnership Project (www.3gpp.org) provides specifications to 3GPP TS 25.212 v4.2.0 (2001-09) standards.

- The Consultative Committee for Space Data Systems (www.ccsds.org) provides specifications to CCSDS 101.0-B-5 standards.

## Technical Support Assistance

Hotline:   1-800-LATTICE (North America)

          +1-503-268-8001 (Outside North America)

e-mail:    techsupport@latticesemi.com

Internet:  www.latticesemi.com

## Revision History

| Date | Version | Change Summary |
|---|---|---|
| — | — | Previous Lattice releases. |
| December 2006 | 04.1 | Added IPexpress User-Configurable Core section. |
| | | Updated LatticeECP/EC appendix. |
| | | Added support for LatticeECP2, LatticeXP and LatticeSC. |
| December 2006 | 04.2 | Updated appendices. Added support for LatticeECP2M device family. |
| July 2007 | 04.3 | Updated to support LatticeXP2 devices with ispLEVER 7.0. |
| November 2008 | 04.4 | Updated signal names. |
| | | Updated appendices. |

# Appendix for ORCA® Series 4 FPGAs

*Table 9. Performance and Utilization[1]*

| Parameter File | Mode | Parameters | ORCA 4 PFUs | LUTs | Registers | PIO | EBR | f_MAX (MHz) |
|---|---|---|---|---|---|---|---|---|
| turbo_deco_o4_1_001.lpc | 3GPP | See Table 1 | 1235 | 3750 | 3569 | 184 | 10 | 46 |
| turbo_deco_o4_1_003.lpc | CCSDS | See Table 1 | 1708 | 3109 | 4859 | 269 | 16 | 36 |

1. Performance and utilization characteristics are generated targeting an OR4E06-2BA352 in ispLEVER 3.0 software.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting ORCA Series 4 devices is TURBO-DECO-O4-N1. Table 1 lists the netlists that are available in Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for ispXPGA® FPGAs

*Table 10. Performance and Utilization[1]*

| Parameter File | Mode | Parameters | ispXPGA PFUs | LUTs | Registers | PIO | EBR | f<sub>MAX</sub> (MHz) |
|---|---|---|---|---|---|---|---|---|
| turbo_deco_xp_1_001.lpc | 3GPP | See Table 1 | 2167 | 4586 | 3628 | 184 | 32 | 52 |
| turbo_deco_xp_1_003.lpc | CCSDS | See Table 1 | 3486 | 5834 | 4966 | 269 | 49 | 46 |

1. Performance and utilization characteristics are generated targeting an LFX1200B-04FE680C in ispLEVER 3.x software. The evaluation version of this IP core only works on this specific device density, package, and speed grade

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting ispXPGA devices is TURBO-DECO-XP-N1. Table 1 lists the netlists available as Evaluation Packages for the ispXPGA device, which can be downloaded from the Lattice web site at www.latticesemi.com. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeECP/EC FPGAs

***Table 11. Performance and Utilization[1]***

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2676 | 5230 | 2838 | 184 | 17 | 82 |
| 3GPP2 | 2834 | 5510 | 3020 | 249 | 49 | 84 |
| CCSDS | 4124 | 8093 | 4294 | 269 | 25 | 68 |

1. Performance and utilization characteristics are generated using LFECP33E-5F672C, with Lattice's ispLEVER v.7.1 SP1 software. When using this IP core in a different density, speed, or grade within the LatticeECP/EC family, performance and utilization may vary.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeECP devices is TURBO-DECO-E2-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeECP2™ FPGAs

*Table 12. Performance and Utilization[1]*

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2732 | 5365 | 2831 | 184 | 13 | 126 |
| 3GPP2 | 2801 | 5490 | 3015 | 249 | 27 | 124 |
| CCSDS | 4274 | 8433 | 4295 | 269 | 16 | 115 |

1. Performance and utilization characteristics are generated using LFE2-70E-7F672C, with Lattice's ispLEVER v.7.1 SP1 software. When using this IP core in a different density, speed, or grade within the LatticeECP2 family, performance and utilization may vary.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeECP2 devices is TURBO-ENCO-P2-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeECP2M™ FPGAs

*Table 13. Performance and Utilization[1]*

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2747 | 5403 | 3085 | 184 | 13 | 110 |
| 3GPP2 | 2712 | 5311 | 3312 | 249 | 27 | 111 |
| CCSDS | 4238 | 8350 | 4540 | 269 | 16 | 100 |

1. Performance and utilization characteristics are generated using LFE2M-35E-7F672C, with Lattice's ispLEVER 7.1 SP 1 software. When using this IP core in a different density, speed, or grade within the LatticeECP2M family, performance and utilization may vary.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeECP2M devices is TURBO-DECO-PM-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeXP™ FPGAs

*Table 14. Performance and Utilization[1, 2]*

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2744 | 5372 | 2838 | 184 | 17 | 76 |
| CCSDS | 4162 | 8268 | 4316 | 269 | 25 | 65 |

1. Performance and utilization characteristics are generated using LFXP20E-5F484C, with Lattice's ispLEVER v.7.1 SP1 software. When using this IP core in a different density, speed, or grade within the LatticeXP family, performance and utilization may vary.
2. Due to limited EBR availability, 3GPP2 is not supported in LatticeXP and not shown in this table.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeXP devices is TURBO-DECO-XM-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeXP2™ FPGAs

*Table 15. Performance and Utilization[1, 2]*

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2801 | 5490 | 3077 | 184 | 13 | 98 |

1. Performance and utilization characteristics are generated using LFXP2-17E-7F484C, with Lattice's ispLEVER v.7.1 SP1 software. When using this IP core in a different density, speed, or grade within the LatticeXP family, performance and utilization may vary.
2. Due to limited EBR availability, 3GPP2 and CCSDS are not supported in LatticeXP2 and are not shown in this table.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeXP2 devices is TURBO-DECO-X2-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Appendix for LatticeSC™ FPGAs

***Table 16. Performance and Utilization[1]***

| IPexpress User-Configurable Mode | SLICEs | LUTs | Registers | I/Os | sysMEM™ EBRs | $f_{MAX}$ (MHz) |
|---|---|---|---|---|---|---|
| 3GPP | 2766 | 5356 | 2827 | 184 | 13 | 172 |
| 3GPP2 | 2747 | 5334 | 3066 | 249 | 27 | 197 |
| CCSDS | 4008 | 7791 | 4326 | 269 | 16 | 174 |

1. Performance and utilization characteristics are generated using LFSC3GA25E-7F900C, with Lattice's ispLEVER 7.1 SP1 software. When using this IP core in a different density, speed, or grade within the LatticeSC family, performance and utilization may vary.

## Ordering Part Number

The Ordering Part Number (OPN) for all configurations of the Turbo Decoder core targeting LatticeSC devices is TURBO-DECO-SC-U3. Table 1 lists the parameter settings that are available for the Turbo Decoder. Use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

# Mouser Electronics

Authorized Distributor


Click to View Pricing, Inventory, Delivery & Lifecycle Information:


[Lattice](#):

[TURBO-DECO-O4-N1](#)  [TURBO-DECO-XP-N1](#)  [TURBO-DECO-X2-U3](#)  [TURBO-DECO-E2-UT3](#)  [TURBO-DECO-P2-UT3](#)  [TURBO-DECO-PM-UT3](#)  [TURBO-DECO-SC-UT3](#)  [TURBO-DECO-X2-UT3](#)  [TURBO-DECO-XM-UT3](#)  [TURBO-DECO-E2-U3](#)  [TURBO-DECO-P2-U3](#)  [TURBO-DECO-PM-U3](#)  [TURBO-DECO-SC-U3](#)  [TURBO-DECO-XM-U3](#)