OTU2 I.7 FEC IP Core (IP-OTU2EFECI7Z) Data Sheet

Revision	0.02
Release Date	2015-02-24
Document number	TD0382

Table of Contents

1 Introduction

The Altera G.975.1 I.7, IP-OTU2EFECI7Z IP core implements the two orthogonally concatenated BCH super FEC code from the appendix I.7 of the ITU-T G.975.1 standard, [1].

The main features are:

- Net electrical coding gain (NECG) of ~8.0 dB
- 7% overhead
- Latency of 156 µsec (includes both encoder + decoder latency)
- 64 bit data path width
- Error statistics monitoring:
    - Counts of corrected zeros
    - Counts of corrected ones
    - Counts of uncorrectable bits

2 Architecture

Figure 1 illustrates the system architecture of the IP-OTU2EFECI7Z IP core.

The FEC encoder receives OTU2 data including frame alignment signal (FAS) and multi-frame alignment signal (MFAS) from the device core. The OTU2 data is encoded with redundant FEC data. The resulting FEC encoded OTU2 must subsequently be scrambled before it is transmitted across the OTN network

In the other direction the frame start of the OTU2 received from the OTN network must be recovered in order to forward the descrambled OTU2 data to the FEC decoder. The redundant FEC data is decoded and identified errors are corrected before the data is forwarded to the device core.

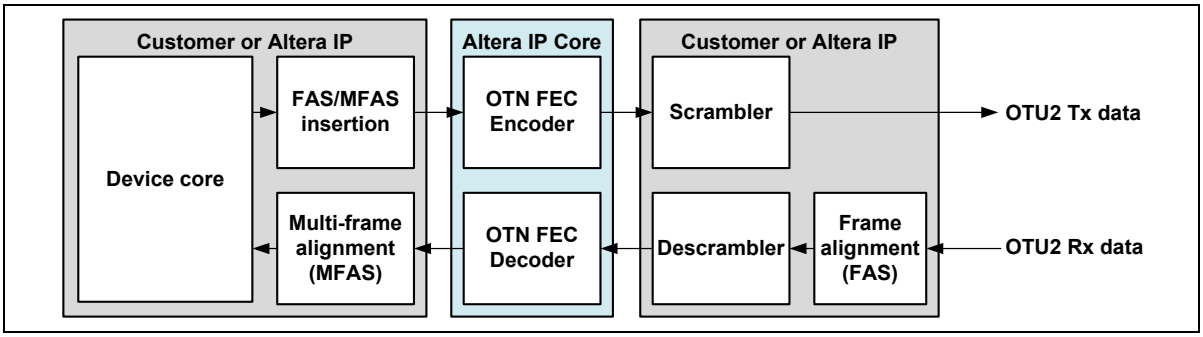

Figure 1: G.975.1 I.7 EFEC System Architecture

3 Performance and Resource Utilization

Arria V devices use combinational adaptive logic modules (ALM) and logic registers. Table 1 shows the typical performance and resource usage for the 10 Gbit/s G.975.1 I.7 EFEC on a Arria V GX I3 device, industrial temperature range with speed grade 3, as reported by the Quartus® II software. The resource figures will vary slightly depending on the Quartus version used, synthesis seed numbers etc. The latencies through the encoder and decoder do not vary from frame to frame; they are a fixed number of clock cycles.

Table 1: Performance - 10 Gbit/s G.975.1 I.7 EFEC on Arria V GX

Block	ALMs	Logic Registers	Memory (M10Ks)	fMax (MHz)	Latency (µs @175 MHz)
Encoder	6,125	6,624	20	>200	~7
Decoder	25,350	38,704	434	>200	~149

4 Functional Overview

4.1 Encoder

The encoder expects a complete standard OTU2 frame including the FEC field at its input. The FEC field will be overwritten, and the input value is thus ignored. At its output the encoder generates a complete I.7 OTU2 frame with the parity information. The module uses one clock, clk_i which is at least OTU2/64.

The data to be encoded shall be delivered on data_i one complete frame at a time, in 2040 consecutive cycles with valid data words. A valid data word has en_i asserted (high). During the first of these cycles fst_i shall be asserted (high). If the clock for the block is faster than OTU2/64, the difference is handled by inserting data words with en_i cleared.

The output of the encoder uses the same format, and is generated a fixed number of cycles after the input. This latency does not vary from frame to frame.

Table 2 lists the encoder input and output ports for connecting to the OTU2 G.975.1 I.7 EFEC IP core.

Table 2: Encoder I/O Port Listing

I/O port	Name	Port Width (Bits)	Description
in	clk_i	1	Clock port. Frequency should be at least OTU2 rate/64.
in	rst_i	1	Async reset. Deassert synchronized to clk_i.
in	en_i	1	Set when data_i is valid.
in	fst_i	1	Pulsed with the first word of the data input, i.e. the word containing the frame start.
in	data_i	64	Data words to be encoded. Valid when en_i is asserted.
out	fst_o	1	Pulsed with the first word of the encoded data output, i.e. the word containing the frame start.
out	en_o	1	Set when data_o is valid.
out	data_o	64	Encoded output data words. Valid when en_o is asserted.

4.2 Decoder

The decoder expects a complete I.7 OTU2 frame on its input, and from this it generates 4 OTU2 rows on its output. The output sequence mimics the way data is typically output by a GFEC decoder. E.g. 4 rows each with 3824 bytes of ODU data and 256 FEC bytes to discard. The block uses one clk_x1_i which must be at least OTU2/64.

The data to be decoded shall be delivered on otn_frame_i one complete frame at a time, in 2040 consecutive cycles with valid data words. A valid data word has otn_frame_valid_i asserted (high). If the clock for the block is faster than OTU2/64, the difference shall be handled by inserting data words with otn_frame_valid_i cleared.

The decoder output is delivered a fixed number of cycles after the input. This latency does not vary from frame to frame.

Table 3 lists the decoder input and output ports for connecting to the OTU2 G.975.1 I.7 EFEC IP core.

Table 3: Decoder I/O Port Listing

I/O port	Name	Port Width (Bits)	Description
in	clk_x1_i	1	Clock port. Frequency should be at least OTU2 rate/64.
in	rst_x1_i	1	Async reset. Deassert synchronized to clk_i.
in	otn_frame_i	DW	Received data words. Valid when otn_frame_valid_i is asserted. Input in 2040 valid data words, i.e. a complete frame. If clk_i is faster than the nominal OTU2 clock (166.33MHz) the difference is absorbed by inserting data words with otn_frame_valid_i cleared.
in	otn_frame_sof_i	1	Pulsed with first word of the OTN frame input, i.e. the word containing the start of frame.
in	otn_frame_valid_i	1	Set when otn_frame_i is valid.
out	otn_frame_o	DW	Corrected payload bytes. Valid when otn_frame_valid_o is asserted. Output in 4 times 478 ODU words with 32 FEC words between each run of 478 words (to emulate G.709 RS FEC output sequence).
out	otn_frame_sof_o	1	Pulsed with first word of the corrected OTN frame output, i.e. the word containing the start of frame.
out	otn_frame_valid_o	1	Set when otn_frame_o is valid.
out	otn_frame_sf_o	1	Signal fail. Output fifo underrun.
Output status registers			
out	rx_dec_in_sync_o	1	Set when the decoder found MSYNC synchronization (rx_dec_sync_state_o == IN_SYNC).
out	rx_dec_sync_state_upd_o	1	Pulsed when rx_dec_sync_state_o changed.
out	rx_dec_sync_state_o	2	State of the decoder MSYNC synchronization: --MSYNC_HUNT_1 = 2'd0; --MSYNC_HUNT_0 = 2'd1; --WAIT_FOR_EOPAR = 2'd2; --IN_SYNC = 2'd3;
out	rx_dec_msync_upd_o	1	Pulsed when rx_dec_msync_o changed.
out	rx_dec_msync_o	32	Last received MSYNC value.
Counting of corrected zeros and ones			
out	counts_valid_o	1	Set when the rx_dec_*_count_o counters are valid.
out	rx_dec_0_corrected_0s_count_o	32	Number of bits received as 0 and corrected to 1 by decoding step 0.
out	rx_dec_1_corrected_0s_count_o	32	Number of bits received as 0 and corrected to 1 by decoding step 1.
out	rx_dec_2_corrected_0s_count_o	32	Number of bits received as 0 and corrected to 1 by decoding step 2.
out	rx_dec_3_corrected_0s_count_o	32	Number of bits received as 0 and corrected to 1 by decoding step 3.
out	rx_dec_4_corrected_0s_count_o	32	Number of bits received as 0 and corrected to 1 by decoding step 4.
out	rx_dec_0_corrected_1s_count_o	32	Number of bits received as 1 and corrected to 0 by decoding step 0.

I/O port	Name	Port Width (Bits)	Description
out	rx_dec_1_corrected_1s_count_o	32	Number of bits received as 1 and corrected to 0 by decoding step 1.
out	rx_dec_2_corrected_1s_count_o	32	Number of bits received as 1 and corrected to 0 by decoding step 2.
out	rx_dec_3_corrected_1s_count_o	32	Number of bits received as 1 and corrected to 0 by decoding step 3.
out	rx_dec_4_corrected_1s_count_o	32	Number of bits received as 1 and corrected to 0 by decoding step 4.
out	rx_dec_0_uncorrected_bits_count_o	32	Number of uncorrected rows after decoding step 0.
out	rx_dec_2_uncorrected_bits_count_o	32	Number of uncorrected bits after decoding step 2.
out	rx_dec_4_uncorrected_bits_count_o	32	Number of uncorrected bits after decoding step 4. Only use rx_dec_4_uncorrected _bits_count_o for statistics (uncorrected bits after the last stage).
Alarms and errors			
out	soft_err_o	1	Set when after a start of frame (sof) is received, the next sof does not arrive at the expected position. Assertion of soft_err_o will automatically reset the decoder.
out	msync_err_valid_o	1	The msync_* errors can be sample when this is asserted
out	msync_crc_err_o	1	CRC error in either MSYNC field was detected.
out	msync_both_crc_err_o	1	CRC error in both MSYNC fields was detected.
out	msync_msb_err_o	1	The MSYNC MSB bit does not follow the 0,1,0,1... sequence
out	msync_ix_phi_err_o	1	After synchronization was recovered an MSYNC was received with incorrect IX/PHI
out	msync_df_pf_err_o	1	After synchronization was recovered an MSYNC was received with incorrect DFILL/PFILL

5 Synthesis constraints

5.1 Encoder

The encoder top level name is: `i7enc_top_wrapper` and the corresponding ZIP file is named `fecenc_otn_i7.zip`.

The `fecenc_otn_i7.zip` contains a complete set of encrypted RTL files for the encoder as well as the license file needed for Quartus decryption.

The following sections describe the settings needed to be used in a high level synthesis flow when the encoder is to be used in a larger design.

5.1.1 Parameters

Not applicable: the toplevel entity for the encoder does not have generics (parameters in Quartus terms).

5.1.2 QSF settings

The following qsf settings are required in a higher chip level synthesis when the decoder is instantiated

```
set_global_assignment -name "REMOVE_DUPLICATE_LOGIC"           "ON"
set_global_assignment -name "REMOVE_DUPLICATE_REGISTERS"       "OFF"
set_global_assignment -name "AUTO_ROM_RECOGNITION"             "ON";
set_global_assignment -name "AUTO_RAM_RECOGNITION"             "ON";
set_global_assignment -name "AUTO_SHIFT_REGISTER_RECOGNITION" "OFF";
```

If this conflicts with chip level preferences then these setting must be applied to the entity level instead.

5.1.3 SDC settings

The encoder entity has one clock, `clk_i`, which must be driven from a PLL.

All remaining inputs and outputs to the encoder are synchronous with respect to `clk_i` and they are all single cycle timed.

5.2 Decoder

The decoder top level name is: `ufec_dec_top_wrapper` and the corresponding ZIP file is named `fecdec_otn_i7.zip`.

The `fecdec_otn_i7.zip` contains a complete set of encrypted RTL files for the decoder as well as the license file needed for Quartus decryption.

The following sections describe the settings needed to be used in a high level synthesis flow when the decoder is to be used in a larger design.

5.2.1 Parameters

The toplevel entity has 1 generic (parameter in Quartus terms): `DW`. `DW` should not be modified and must be left at its default value.

5.2.2 QSF settings

The following qsf settings are required in a higher chip level synthesis when the decoder is instantiated

```
set_global_assignment -name "REMOVE_DUPLICATE_LOGIC"           "ON"
```

```
set_global_assignment -name "REMOVE_DUPLICATE_REGISTERS"      "OFF"
set_global_assignment -name "AUTO_ROM_RECOGNITION"            "ON";
set_global_assignment -name "AUTO_RAM_RECOGNITION"            "ON";
set_global_assignment -name "AUTO_SHIFT_REGISTER_RECOGNITION" "OFF";
```

If this conflicts with chip level preferences then these setting must be applied to the entity level instead.

5.2.3 SDC settings

The decoder entity has one clock, `clk_x1_i, which` must be driven from a PLL.

All remaining inputs and outputs to the decoder are synchronous with respect to `clk_x1_i` and they are all single cycle timed.

6 References

[1] ITU-T Recommendation G.975.1, *Forward error correction for high bit-rate DWDM submarine systems*, ITU-T, Geneva, February 2004.

7 Revision history

Date	Revision	Description of changes
2015-02-23	0.01	• Initial version.
2015-02-24	0.02	• Review comment resolution. • Latency numbers.