High Speed - Low Latency
OTU-1 RS Decoder Core

Revision History

Version	Date	Details
Version 0.1	12.08.03	First Draft
Version 0.2	08.11.04	Updated pinout
Version 0.3	13.11.04	Updated resource utilisation
Version 0.4	17.11.2004	Added Virtex4 info

Contents

1 Features

The Aliathon OTU-1 RS Decoder Core provides a very flexible, resource efficient FPGA based Core for G.709 Forward Error Correction (FEC) applications. The core:

- Is fully compatible with other supporting Aliathon Cores.
- Implements RS Decoding using Berlekamp-Massey algorithm for low latency.
- Implements G.709 RS (255,239) OTU-1 Reed Solomon Decoding.
- Counts all corrected bits, symbols and indicates uncorrectable Code Words.
- Is fully synchronous and runs at very high clock speeds.
- Is available for Altera and Xilinx FPGAs.

2 Functional Description

Figure 1 illustrates the major functional blocks within the Decoder Core.

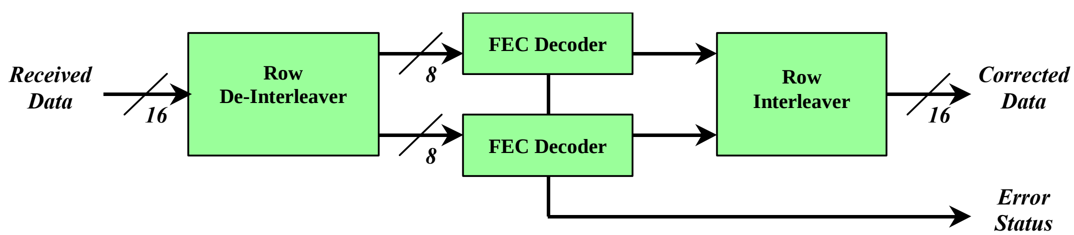

Figure 1

2.1 Row De-Interleaver

The Row De-Interleaver simply takes the OTU-1 Rows and extracts the 16 Interleaved RS (255,239) Codewords. This then allows the Codewords to be presented sequentially to the FEC Decoders.

2.2 FEC Decoder

Figure 2 illustrates the major functional blocks within the FEC Decoder.

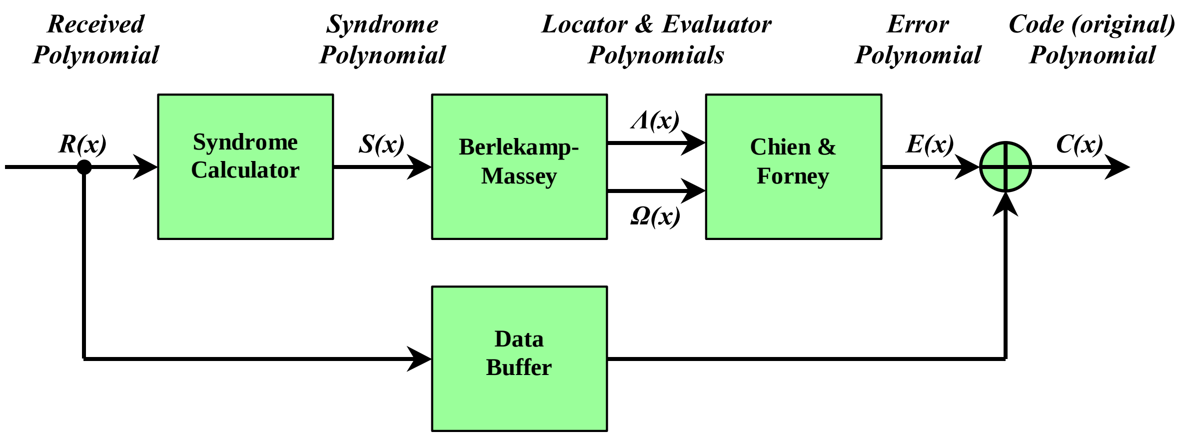

Figure 2

The OTU-1 Data is processed 16-bits wide, so we need two FEC Decoders. The FEC Decoder corrects the received Codewords and sends them to the Row Interleaver.

2.2.1 Syndrome Calculator

The Syndrome calculator block calculates the Syndrome ***S(x)*** values from the received Codeword ***R(x)***. These Syndromes are the used to solve the 'Key Equation'. This equation relates the Error ***Locator*** Polynomial ***Λ(x)*** and the Error ***Evaluator*** Polynomial **Ω*(x)*** as follows:

$$\Lambda(x)\, S(x) \equiv \Omega(x) \bmod x^{2t}$$

Where **t** is the number of errors that can be detected using the Code.

2.2.2 Berlekamp-Massey

The Berlekamp-Massey block is a fast method of solving the 'Key-Equation' from the Syndrome values. This block produces both the ***Locator*** and ***Evaluator*** Polynomials, which together may be used to find the error positions and values.

2.2.3 Chien Search and Forney Calculation

The Chien Search and Forney Calculation block determines the positions and values of the errors from the ***Locator*** and ***Evaluator*** Polynomials. This block produces the ***Error*** Polynomial, the coefficients of which are simply the error positions and values. Finally, these values are the added to the received Codeword to reconstruct the original Transmitted Codeword.

2.2.4 Data Buffer

The Data Buffer block simply stores the received Codeword for the period of time taken to calculate the errors. After this calculation, the buffered Codeword is combined with the ***Error*** Polynomial to reconstruct the original Transmitted Codeword.

2.3 Row Interleaver

The Row Interleaver takes the 16 corrected RS (255,239) Codewords and combines them to reconstruct the corrected OTU-1 Rows.

2.4 Error Status

The FEC Decoders continuously indicates the number and values of the errors. This may be used to count the error rate. In addition, each Decoder will indicate if it has been unable to correct a particular received Codeword. This may occur if there are more than **8** byte errors in the Codeword. Please remember that the nature of RS Codes mean that it is possible to have Code words which look correctable, but which are not.

2.5 Performance

The performance of the OTU-1 Decoder is determined by the latency of the FEC Decoder and the Interleaving functions. The Interleaving and De-Interleaving functions add 1 Row of latency each with the FEC Decoder function adding an extra 2.6µS. This brings the total latency to approximately **27µS**.

3 Signal Description

The input and output signals are grouped by function into the following interfaces. Note that the <***ufi_data***>, <***dfo_data***> and <***dfo_err***> signals are 2 symbols (16-bits) wide…

3.1 Uncorrected Data Input Interface

This interface provides the input clock and data to the OTU-1 Decoder core.

Name	Type	Description
rst	**I**	Asynchronous reset input.
clk	**I**	**clk** is typically supplied by the external LIU. All inputs and outputs from the core are synchronous to this clock unless otherwise noted.
ufi_vld	**I**	When asserted this indicates that there is valid data on **ufi_data**. This input is useful for driving the core with data from another clock domain, via an asynchronous FIFO, for example.
ufi_data	**I**	**ufi_data** is the uncorrected data input. When **ufi_vld** is asserted it must carry valid data. Note that MSB is the first received.
ufi_sof	**I**	When asserted this indicates that the data on **ufi_data** is the first byte in the received OTU-1 Frame. This input is only read when **ufi_vld** asserted.

3.2 Corrected Data Output Interface

This interface provides the Corrected Output data from the OTU-1 Decoder.

Name	Type	Description
dfo_vld	**O**	When asserted this indicates that the following outputs are valid.
dfo_data	**O**	**dfo_data** is the corrected data output. When **dfo_vld** is asserted it contains valid data. Note that MSB is the first received.
dfo_sof	**O**	When asserted this indicates that the data on **dfo_data** is the first byte in the received OTU-1 Frame. This input is only read when **dfo_vld** asserted.

3.3 Config / Status Interface

This interface provides status on the current error correction process.

Name	Type	Description
fec_off	**I**	When asserted the core does not correct errors.
dfo_err	**O**	This single-cycle output indicates any detected errors. A '1' in any bit of this output indicates an errored bit in the corresponding received data.
dfo_uncorr	**O**	This output indicates that one of the received Codewords was uncorrectable. The signal is active for one clock cycle aligned to the last byte of the received Codeword that was uncorrectable.

4 Implementation Details

4.1 Resource Utilisation

The following figures are calculated assuming that all core IOs are routed off-chip. This results in a worst-case resource utilisation figure, and for any given application the resource utilisation is likely to be lower. The example parts are the mid-speed-grade, and the core exceeds performance requirements (170MHz for OTN) in these devices.

The core can be targeted for devices from other families and manufactures. Contact Aliathon for further details.

4.1.1 Altera

	Stratix Family E.g.: EP1S10F484C6			Stratix Family E.g.: EP1SGX10CF672C6		
	Used by Core	In example Part	% Utilised	Used by Core	In example Part	% Utilised
Logic Elements (LEs)	2531	10570	24%	2531	10570	24%
M512 RAM Blocks	6	94	7%	6	94	7%
M4k RAM Blocks	36	60	60%	36	60	60%
M-RAM Blocks	0	1	0%	0	1	0%
Fmax	>170MHz			>170MHz		

	Stratix2 Family E.g.: EP2S15F484C5		
	Used by Core	In example Part	% Utilised
ALMs	1230	6240	20%
M512 RAM Blocks	6	104	6%
M4k RAM Blocks	36	78	47%
M-RAM Blocks	0	1	0%
Fmax	>170MHz		

4.1.2 Xilinx

	Virtex-II Family E.g.: XC2V1000-5FG456			Virtex-II Pro Family E.g.: XC2VP7-5FG456		
	Used by Core	In example Part	% Utilised	Used by Core	In example Part	% Utilised
Slices	1389	5120	28%	1389	4928	29%
BlockRAMs	17	40	43%	17	44	39%
Fmax	>170MHz			>170MHz		

	Virtex-IV Family E.g.: XC4VLX15FF668-10		
	Used by Core	In example Part	% Utilised
Slices	1455	6140	24%
BlockRAMs	17	40	43%
Fmax	>170MHz		

5 Ordering Information

For technical enquiries and ordering please contact Aliathon Ltd at:

Aliathon Ltd
Evans Business Centre
Pitreavie Court, Dunfermline
United Kingdom
KY11 5UU

Phone: **+44 (0)1383 737 736**
Fax: **+44 (0)1383 749 501**
Email: info@aliathon.com
Web: www.aliathon.com