

## JE301 and JE302 Stand Alone JPEG Encoder for Lattice EC and ECP FGAs

10. April 2005

Product Specification V0.5



Penz VHDL  
Frankenstraße 16  
D-55299 Nackenheim  
Germany

Phone: +49 (0)6135-950328  
Fax: +49 (0)6135-950329  
E-Mail: mail@penz-vhdl.de  
URL: www.penz-vhdl.de

### Features

- Optimized for Lattice EC and ECP FGAs
- Marker generation included
- Block building RAM included, no external RAM needed
- JPEG file output
- Compliant with Baseline ISO/IEC 10918-1
- Motion-JPEG capability
- Monochrome or Color (YCbCr 4:2:2)
- Up to 2048\* samples per row
- Baseline encoder
- 8-bit/pixel input
- Line by line pixel input
- 2 Quantization tables, reprogrammable
- 4 fixed Huffman tables (two DC and two AC)
- Fully synchronous design
- Fully stall able design
- Single clock cycle per pixel encoding
- No pause cycles between blocks

Core Facts	
Provided with core	
Documentation	Programmers Manual
Design File Formats	netlist
Verification	Testbench, Testvectors
Constraints File	Not needed
Instantiation Template	VHDL
Reference design & application notes	Demo Application
Additional Items	
Simulator Tool Used	
ModelTech ModelSim 5.8	
Support	
Penz VHDL	

### Applications

The JE301 and JE302 are designed to generate a baseline encoded (Motion) JPEG file from an image, in real-time and without using a CPU. It is suitable where an image must be compressed in real-time, like in Frame grabbers, intelligent cameras, surveillance systems and scanners.

The JE301 is optimized for EC devices.

The JE302 is optimized for ECP devices.

Table 1: Core Implementation

Supported Family	Device Tested	Samples per Row	SLICES	Block RAM	9x9 Multiplier	IOBs	Performance	Lattice Tools
EC	LFECE10-3	2048*	3400	17	--	75	50 MHz	ISPLever 4.2
ECP	LFCEP10-3	2048*	2700	17	24	75	60 MHz	ISPLever 4.2

\* Contact us for more samples per row.

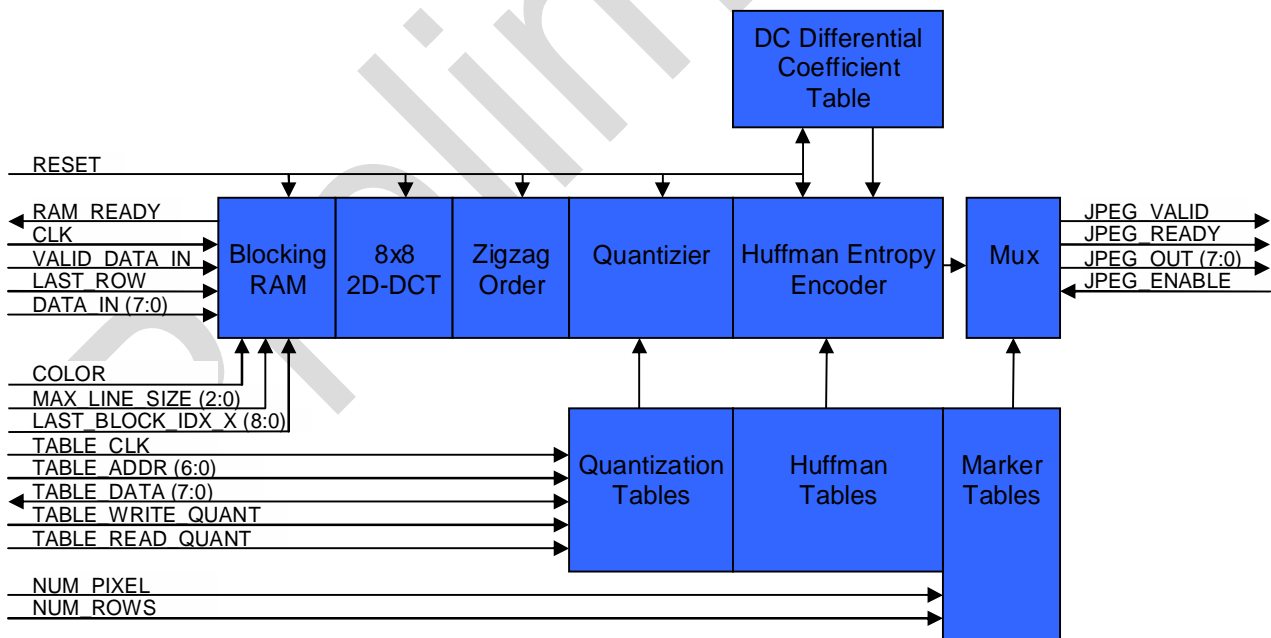
## General Description

The JE301 and JE302 encodes a monochrome or color (YCrCb 4:2:2) image into a JPEG standard conform bit stream. With build in RAM, to rearrange the lines into the 8x8 pixel blocks. Base-line encoding with a pixel size of 8 bit is used. All needed Markers are generated too. The complete JPEG file appears on the 8 bit output register. The core is stall able by the handshake signals on the pixel input and the file output interface. Standard DC- and AC-tables for quantization and Huffman encoding are predefined. The content of the Quantization tables can be reprogrammed.

## Functional Description

The incoming lines are stored into the internal RAM and read out in 8x8 pixel blocks. The block goes through the two-dimensional discrete cosine transformation (2D-DCT) and will be transformed into 64 coefficients into the frequency domain. Now, the coefficients will be read out in a Zigzag order and quantized by dividing through the selected Quantization table. The first (DC) quantized coefficient is differentially coded, using the most recently DC coefficient from the same component. Now all coefficients are run length coded to Run-Size symbols. Finally the symbols are Huffman coded, using the code from the selected Huffman table. The bit-stream of Huffman codes is divided into parts with 8 bit and embedded between all needed Markers to generate a valid JPEG file.

Figure 1: Encoder Block Diagram



## Core Modifications

Usually no core modifications are necessary, but when a special interface or more tables are needed, modifications can be applied. Please contact Penz-VHDL for more information.

## Verification Methods

The encoder has been validated as functional and timing simulation with several test patterns under ModelSim 5.8. A finally verification has been done in a demo-application with some color and BW images.

## Pinouts

Table 2: Core Signal Pinout

Signal Name	Direct.	Description
<b>Pixel Input Interface</b>		
CLK	In	Encoder clock
RESET	In	Reset encoder state machines
VALID_DATA_IN	In	Pixel is valid
DATA_IN (7 : 0)	In	Pixel input
LAST_ROW	In	Last row in this image
COLOR	In	Mode, mono chrome or color
MAX_LINE_SIZE (2:0)	In	Number of samples per row
LAST_BLOCK_IDX_X (8:0)	In	Index of the last 8 pixel-block in a row
NUM_PIXEL (11:0)	In	Number of pixel per row
NUM_ROWS (11:0)	In	Number of rows per image
RAM_READY	Out	Encoder is ready for new pixel
<b>Compressed Data Output Interface</b>		
JPEG_ENABLE	In	Data handshake
JPEG_VALID	Out	Output data is valid
JPEG_READY	Out	Last data for this field
JPEG_OUT (7:0)	Out	Output data
<b>Tables Programming Interface</b>		
TABLE_CLK	In	Clock for table reprogramming
TABLE_ADDR (6:0)	In	Table select and addressing
TABLE_DATA (7:0)	In/Out	Table read- or write-data
TABLE_WRITE_QUANT	In	Write data into Quantization table
TABLE_READ_QUANT	In	Read data from Quantization table

## Recommended Design Experience

Only basic VHDL knowledge is necessary, no manual placement is required. Advanced designers can reach a higher performance.

## Ordering Information

The JPEG Encoder core is provided under license by Penz-VHDL. The VHDL source code is also available. Please contact Penz-VHDL for information about pricing and conditions of sale.