INT_DEINT Interleaver Deinterleaver

September 11, 2000

Product Specification

CSELT S.p.A

Via G. Reiss Romoli, 274
I-10148 Torino, Italy
Phone: +39 011 228 5259
Fax: +39 011 228 5695
E-mail: viplibrary@cselt.it
URL: www.cselt.it

Features

- Interleaver/Deinterleaver functionality
- Customizable RTL source code allowing generation of different netlist versions
- The high flexibility of the core allows the generation of netlists compliant with international standards such as UMTS (3GPP), GSM (ETS 300 909) and DVB (EN 300 744)
- Customized testbench for pre- and post-synthesis verification supplied with the module
- Core customization:
  - Programmable Interleaving or Deinterleaving functionality
  - Different architectures are provided to guarantee the best implementation in terms of area and poerformance for a wide range of re-ordering methods.
  - Programmable number of (de)interleaving modes simultaneously supported to use the same implementation of the core to perform interleaving (or deinterleaving) for up to four different data flows, such as GSM TCH/FS and GSM SACCH.
  - Programmable interleaving depth and input/output data block size
  - Programmable data and memory word size

AllianceCORE™ Facts

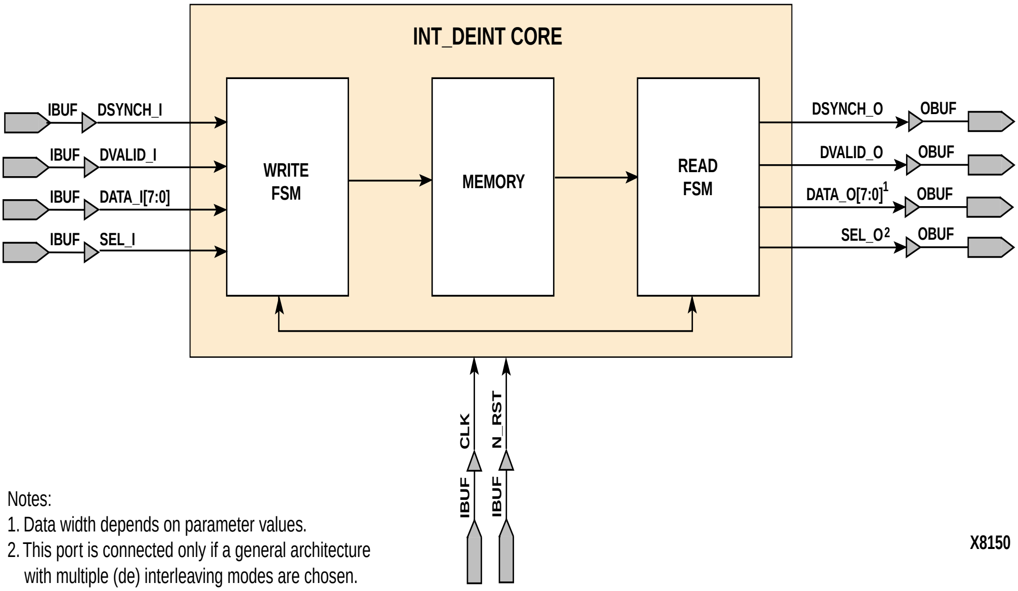
Core Specifics[1]		
Supported Family	Spartan	Virtex
Device Tested	S40-3	V50-6
CLBs[2]	612	160
Clock IOBs	1	1
IOBs[3]	21	21
Performance (MHz)	12	73
Xilinx Tools	M3.1i	M3.1i
Special Features	SelectRAM	BlockRAM
Provided with Core		
Documentation	User Manual	
Design File Formats	EDIF netlist, XNF netlist	
Constraints File	NCF constraint file	
Verification	VHDL testbench	
Instantiation Templates	VHDL, Verilog	
Reference Designs & Application Notes	None	
Additional Items	None	
Simulation Tool Used		
Synopsys VSS		
Support		
Design and customization support provided by CSELT		

Notes:
1. Data refer to the following customization:
   Convolutional Interleaver;
   Interleaving depth equal to 8;
   M is equal to 12;
   Data size is equal to 8;
2. Utilization numbers for Virtex are in CLB slices
3. Assuming all core I/Os are routed off-chip

Applications

The interleaving/deinterleaving approach is widely used in digital telecommunication systems, in the channel coding part. This function is performed by changing the order of data before the transmission on the channel (interleaving) and then reordering them at the receiver stage (deinterleaving).

Thanks to code flexibility the core can be used in a wide range of applications such as GSM TCH/FS, GSM TCH F9.6, GSM SACCH, GSM FACCH/H, UMTS 3GPP, DVB: terrestrial broadcasting (bit-wise interleaving).



Notes:
1. Data width depends on parameter values.
2. This port is connected only if a general architecture with multiple (de) interleaving modes are chosen.

Figure 1: Interleaver Deinterleaver Block Diagram

General Description

The INT_DEINT core implements both the Interleaver and Deinterleaver functionalities, changing the order of data, to spread burst errors introduced by the channel. The core can be configured to implement different ordering and re-ordering methods: The general (non block-based) interleaver, a convolutional (non block-based) interleaver, and as a block-based interleaver.

In the general interleaver, the input data can be mapped on more than one output block, so that each output block contains data from various input blocks. See figure 2.

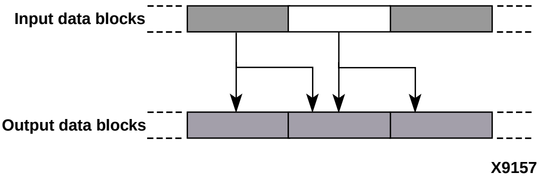


Figure 2: General non block-based Interleaving

To provide an easier parameter specification and to optimize the hardware implementation, two additional interleaver architectures are used: the convolutional and the block-based column-reading architectures.

The convolutional interleaver is a peculiar type of non block-based interleaver, widely used in applications together with concatenated coders/decoders. See Figure 3.

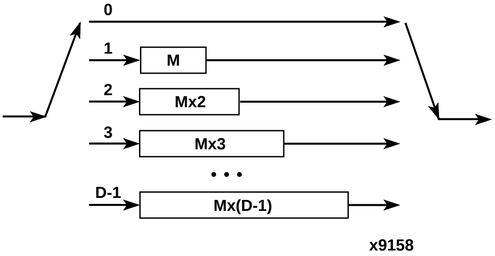


Figure 3: Convolutional Interleaving

The last architecture is a Block-based column reading interleaver where data are written in a fixed order (by row) and read by column. See Figure 4.

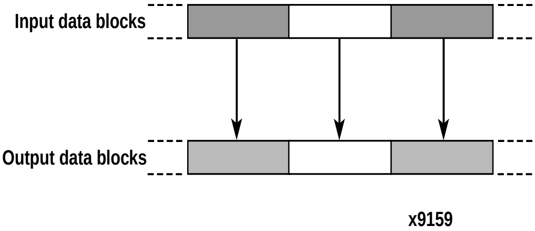


Figure 4: Block-based Interleaving

The general architecture (non block interleaver) supports multiple interleaving\deinterleaving modes. This function allows in using the same hardware to perform interleaving/deinterleaving for up to four different interleaving/deinterleaving ordering methods (i.e. GSM TCH/FS, GSM TCH F9.6, GSM SACCH, GSM FACCH/H).

The macro programmability includes: definition of the architecture, multiple (de)interleaving modes, (de)interleaving depth, input and output block size, data size, memory word size.

The Absolute Maximum ratings, Operating Conditions, DC Electrical Specifications and Capacitances depend on the Xilinx device selected for implementation and can be retrieved in the corresponding Xilinx datasheet.

Functional Description

The internal architecture of the INT_DEINT core is shown in Figure 1. A brief description of the operation of each module follows.

Write FSM

This module is responsible of writing the incoming data in the memory. This module performs no reordering.

Memory

The memory module is used to store the data in the order received.

Read FSM

This module reads data from the memory in the correct order.

Pinout

The pinout of this core has not been fixed to a specific FPGA I/O allowing flexibility with a user's application. Signal names are shown in the block diagram in Figure 1 and described in Table 1.

Table 1: Core Signal Pinout

Signal	Signal Direction	Description
DSYNCH_I	Input	Data synchronism input
DVALID_I	Input	Data valid input
DATA_I[7:0] (1)	Input	Data input
SEL_I	Input	(De)interleaving mode selection input
CLK	Input	Master clock
N_RST	Input	Asynchronous reset
DSYNCH_O	Output	Data synchronism output
DVALID_O	Output	Data valid output
DATA_O[7:0]	Output	Data output
SEL_O (2)	Output	(De)interleaving mode selected output

Notes:
1. Port size is equal to DSIZE (a generic value)
2. This port is connected only if a general architecture with multiple (de) interleaving modes is chosen (depends on IDNUM and IDTYPE_1 generic values).

Core Modifications

The INT_DEINT core is parameterizable. Parameters are implemented as a set of generics in the synthesisable VHDL source code of the core. Parameters allow the user to specify some architectural and functional features of the synthesized core netlist, so as to adapt it to a specific design or application. Core parameters are shown in Table 2.

Table 2: Core Parameters (VHDL Generics)

Parameter	Description
IDSEL	Selects interleaving or deinterleaving mode
IDNUM	Number of (de)interleaving modes supported
IDTYPE_x (x=0,1,2,3)	Selects (de)interleaving type for the x-th mode
COEFF_B1_x, COEFF_B2_x, COEFF_B3_x, COEFF_B4_x, COEFF_B5_x, COEFF_B6_x (x=0,1,2,3)v	Coefficients for the first reordering formula for the x-th mode
COEFF_J1_x, COEFF_J2_x, COEFF_J3_x, COEFF_J4_x, COEFF_J5_x, COEFF_J6_x, COEFF_J7_x, COEFF_J8_x (x=0,1,2,3)	Coefficients for the second reordering formula for the x-th mode
DEPTH_x (x=0,1,2,3)	Interleaving depth of the x-th mode
BISIZE	Block In size
BOSIZE	Block Out size
DSIZE	Data size

Verification Methods

Extensive functional (pre-synthesis) and timing (post-synthesis) simulation has been performed for different values of the core parameters, using the Synopsys VSS simulator. Simulation scenarios (including data and command files) and parametric testbench for design verification are provided with the core.

The parametric testbench is composed of a convolutional encoder, a noisy transmission channel, and a depuncturing emulator. The data input flow, channel noise level, and puncturing parameters are easily customizable by editing some text files. The INT_DEINT_GENERATOR block provides all the signals needed to test the INT_DEINT macro, while the DATA_READ block captures the data produced by the macro. The test environment shown in Figure 5.

The stimuli generator block can be programmed using two input files provided :

program.txt: used to set the control signals values;
data_in.txt: used to set the input data values.

The DATA_READ block produces a number of log files equal to the number of (de)interleaving modes supported by the selected configuration.

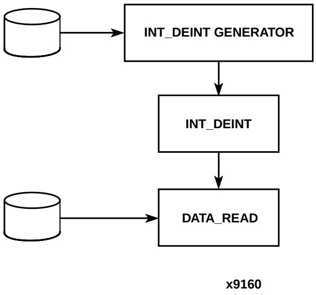


Figure 5: Test Environment

Recommended Design Experience

Experience with the Xilinx design flow is recommended to the users of the netlist version of the core. For the source code version, users should also be familiar with synthesis tools (such as VHDL Compiler, FPGA Compiler) and simulator (such as VSS, Modelsim).

Ordering Information

The INT_DEINT core is provided under license by CSELT S.p.A. for use in Xilinx programmable logic devices. Please contact CSELT S.p.A. for information about pricing, terms and conditions of sale.

CSELT S.p.A. reserves the right to change any specification detailed in this document at any time without notice, and assumes no responsibility for any error in this document.

All trademarks, registered trademarks, or servicemarks are property of their respective owners.

Related Information

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

For general Xilinx literature, contact:

Phone: +1 800-231-3386 (inside the US)
+1 408-879-5017 (outside the US)
E-mail: literature@xilinx.com

Implementation Request Form

To: CSELT S.p.A.

FAX: +39 011 228 5695

E-mail: viplibrary@cselt.it

CSELT configures and ships Xilinx netlist versions of the INT_DEINT core customized to your specification. Please fill out and fax this form so CSELT can respond with an appropriate quotation that includes performance and density metrics for the target Xilinx FPGA.

From: ____

Company:______________________________________

Address:_______________________________________

City,State,Zip:__________________________________

Country:_______________________________________

Phone:____

FAX:____

E-mail:____

Implementation Issues

1. Interleaver / Deinterleaver mode ________________

2. Input data bus width ________________

3. Interleaver mode_________________

4. Interleaver depth____________

5. Input/Output data block size________________

6. Memory word size__________________________

Business Issues

1. Indicate timescales of requirement:
______ date for decision
______ date for placing order
______ date of delivery

2. Indicate your area of responsibility:
______ decision maker
______ budget holder
______ recommender

3. Has a budget been allocated for the purchase?
Yes ______ No ______

4. What volume do you expect to ship of the product that will use this core? ______

5. What major factors will influence your decision?
______ cost
______ customization
______ testing
______ implementation size

6. Are you considering any other solutions? _______