HDLC IP v1.0

Key Features

- Synthesizable, technology independent VHDL IP Core
- HDLC compatible serial interface controller
- Address filtering, multicast and broadcast
- Receive input FIFO with configurable depth
- Transmit output FIFO with configurable depth
- Supports all standard data rates from 9600 to 921600 baud
- Fully custom data rates also supported – limited only by system
- Receive and transmit interrupt flags
- Receive and transmit FIFO full flags
- NRZ, NRZI with configurable level and Manchester Data Encoding
- Bit Stuffing and Bit Stripping
- Configurable CRC: 16-bit and 32-bit sequence
- Full Dublex
- Optional Avalon MM Interface
- Optional All Programmable parameters enable run-time reconfigurability

Supported FPGA

- Any

Supported Simulators

- Active-HDL v9.1 or higher,
- Riviera Pro v2014.02 or higher

Supported Synthesizers

- Mentor Precision r2013b.15 or later,
- Xilinx ISE 14.7/Vivado 2013.4 or later,
- Quartus II 14.1 or later,
- Synopsys Synplify 2015.03 or later (Lattice, Microsemi and Achronix versions)

Verification and Validation Flow

- Self-checking test bench
- Static timing analysis
- Code coverage
- Linting

Application

- Communication Systems
- Serial Interfaces

1 OVERVIEW

This datasheet describes the CES HDLC Core, a completely parameterized VHDL IP core able to send and receive serial data in HDLC format full duplex mode with configurable data rates, crc and data encoding. This IP core is designed to interface with both AXI4 Lite and Avalon protocol, or as a standalone core.

The CES_IO_HDLC IP supports all baud rates from 9600 to 921600 baud, although higher and lower rates may be supported depending on the system clock frequency. Both the receiver and transmitter circuits have a configurable FIFO which may be used to buffer the parallel input and output data, if required. In addition, this module includes two port used as input control register and output status register.

This document provides the theoretical basis of the implementing choice, details of the cores and the design services that CES can support to tailor the HDLC IP Core to your application.

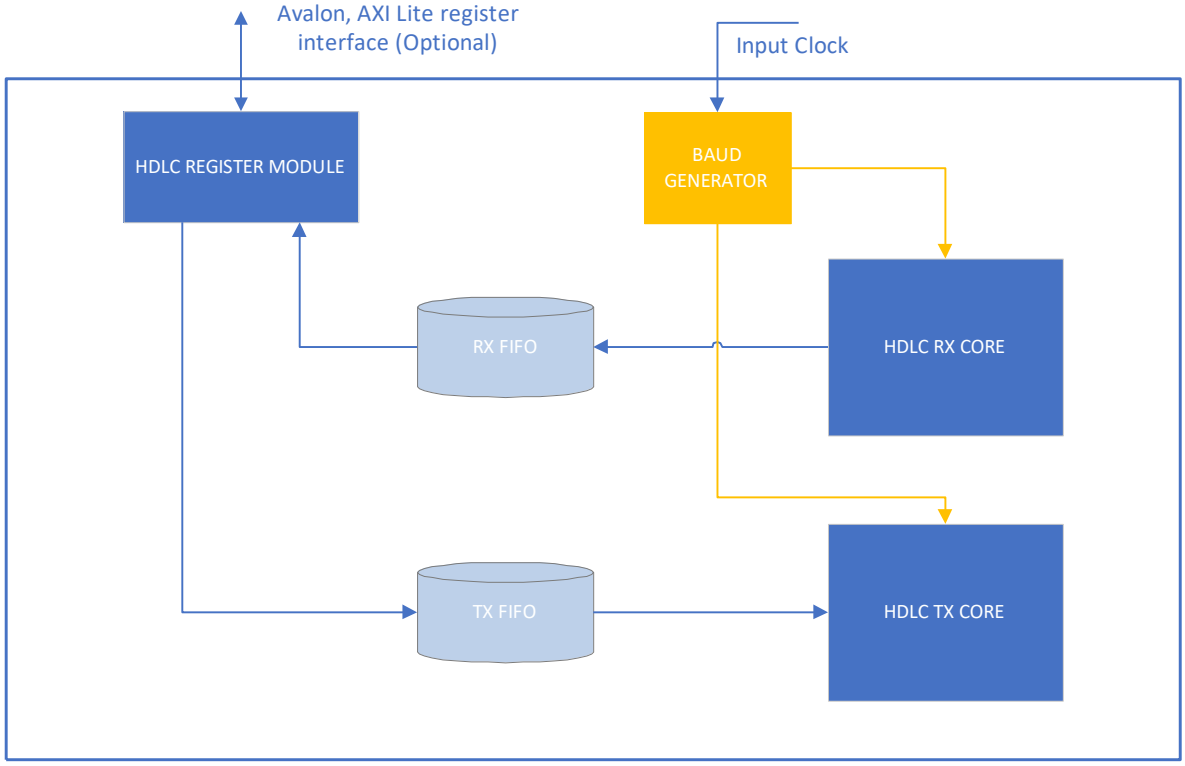

Figure 1. HDLC IP Block Diagram

1.1 LICENSING AND ORDERING INFORMATION

Information about licensing, pricing and availability is available on the company website (http://www.campera-es.com/), or after contacting the company or your local distributor. Three types of licenses keys may be available:

- Simulation Only License
- Full System Hardware Evaluation License
- Full License

1.2 DESIGN METHODOLOGY

The code is compiled following an internal coding standard, and the flow is developed under constant simulation check. The working code undergoes a linting procedure using Alint software to spot main code errors or generic issues.

Each piece of design is simulated in a dedicated testbench, possibly of self-checking type. Various combinations of generic parameters are tested, looking for critical situations. The whole span of admitted values is explored, and care is taken about treatment of values which are not representable or leading to standard errors.

Synthesis is tested on selected sample targets, to collect data about resource occupation and maximum operative frequencies.

2 PRODUCT SPECIFICATION

The CES_IO_HDLC module is part of the CES IO Library, and is available in several configurations

- Simple Interface or AXI/Avalon register mapped interface
- Run time programmable (baud rate, encoding, bit stuffing configurable at run time via proper registers) or configurable via generics at compile time
- Full Duplex

When the core is controlled through a standard bus protocol (AXI/Avalon) a register map interface with control, status TX and RX is used to control the UART operations.

3 HDLC IP CORE INTERFACE

CES HDLC IP Core is a state-of-the-art serial communication module, highly parameterizes that can be easily configured to meet the demands of each user application.

3.1 GENERIC DESCRIPTION

The generic parameters, to be set at pre-synthesis level, are listed in Table 1.

Generic	Type	Description
g_programmable	integer	Select if the module is programmable through its control register (1) or not (0)
g_fifo_depth	integer	Tx input and rx output fifo depth
g_freq	integer	System input frequency (Hz)
g_baud_rate	integer	Baud rate value
g_baud_tick	integer	Oversampling factor of serial data (8/16/32)
g_encoding	std_logic_vector	HDLC data encoding: “000” NRZ, “001” Manchester, “010” NRZI (3 bits)
g_crc_polynomial	std_logic_vector	CRC polynomial selection (16 bits)

g_nrz	std_logic	NRZI change level
g_max_frame_size	natural	Maximum frame size (bytes)
g_bit_stuff	std_logic	Bit stuffing activation: (1) active, (0) inactive
g_address	std_logic_vector	Rx address selection (8 bits)
g_rst_lvl	std_logic	default reset level

Table 1. HDLC generic list and description

3.2 PORT DESCRIPTION

The module ports are listed in Table 2

Port name	Type	Direction	Description
clk_i	std_logic	input	System clock.
rst_i	std_logic	input	General reset.
din_i	std_logic_vector	input	Input data
din_valid_i	std_logic	Input	Data valid for input data
ready_for_data_o	std_logic	Output	Request for input data (input fifo less than ¾ full)
txempty_o	std_logic	Output	Input fifo empty
tx_o	std_logic	Output	Output data from Tx
tx_clk_o	std_logic	Output	Output data Tx clock
din_rx_i	std_logic	Input	Incoming Rx serial data
din_rx_clk_i	std_logic	Input	Incoming Rx serial data clock
rxempty_o	std_logic	Output	Output fifo empty
dout_o	std_logic_vector	Output	Output data
dout_req_i	std_logic	Input	Output data request
stat_reg_o	std_logic_vector	Output	Status register
ctrl_reg_i	std_logic_vector	Input	Control register

Table 2. HDLC ports and description

4 GENERAL DESCRIPTION

The HDLC module is a bit-oriented synchronous data link layer protocol used for point-to-multipoint connections. It can be controlled by the user through the control and status registers. The module receives and sends data over a standard HDLC serial line.

HDLC frames are marked by the word "01111110" (0x7E) in the beginning and at the end. In Table 3

Flag	Address	Control	Information	FCS	Flag
8 bits	8 bits	8 or 16 bits	Variable length 8xn bits	16 or 32 bits	8 bits

Table 3. HDLC Frame Structure

The frame check sequence (FCS) is computed over the Address, Control and Information fields and can be either 16-bit CRC-CCITT or 32-bit CRC-32, defined by the module generics.

4.1.1.1 Bit Stuffing

The receiving device recognizes the beginning and the end of the frame by identifying 6 1-bits in a row. Therefore, all other frame sequences cannot have more than 5 1-bits in a row. In order to avoid this an additional 0 every time 5 consecutive 1-bits appear in the data. In the receiver, the reverse procedure is executed, after 5 1-bits a zero is removed. Bit stuffing can be activated and deactivated by the **g_bit_stuff** generic parameter.

4.1.1.2 Encodings

The module supports three different types of encodings, defined by the generic parameter **g_encoding:**

- **NRZ (Non-return -to-zero):** 1 is represented by logic '1' and 0 is represented by logic '0'
- **Manchester encoding:** In this module IEEE. 802.3 Manchester encoding is implemented. In this case a logic 0 is represented by a high-low signal sequence and a logic 1 by a low-high signal sequence (Figure 2).
- **NRZI (Non-return-to-zero-inverted):** This encoding distinguishes data bits by the absence or presence of a transition at a clock boundary. The logic level for which a transition is made is defined by generic parameter **g_nrz** (Figure 3).

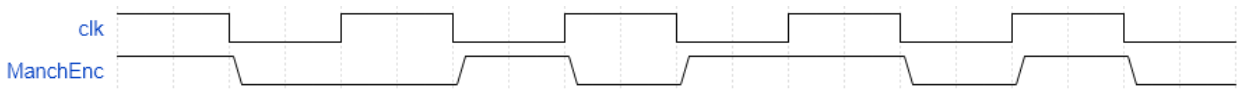

Figure 2. Manchester Encoding sequence "01100"

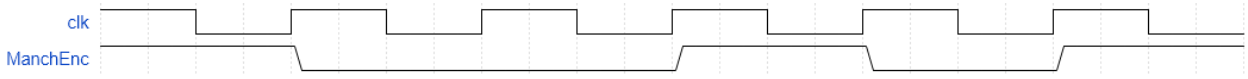

Figure 3. NRZI Encoding sequence "01100" (transition on logic level '0'

4.1.1.3 Registers

The control register can be used to change the parameters of the HDLC IP at runtime (the change can occur only when the IP is idle, not during transmission or receiver operation – only between frames). Bit 15 of the Control register is the send frame command.

The status register ports information on the fifo status, end of frame, and error identification encoded for both rx and tx.

Detailed description of both registers can be found in Table 4.

Register Name	Description

Control Register	(RW) Data (7 downto 0) – HDLC address Data (10 downto 8) – Baud rate "000" – 9600 "001" –19200 "010" – 38400 "011" – 57600 "100" – 115200 ← Default value "101" – 230400 "110" – 460800 "111" – 921600 Data (13 downto 11) – HDLC encoding "000" – NRZ "001" – Manchester "010" – NRZ(I) Data(14)-- NRZI level selection Data(15) – Send Frame: '1' active, '0' inactive Data(16) – Bit stuffing control: '1' active, '0' inactive
Status Register	(RO) (8bits used) Format: Data (0) --rx fifo empty: 1 rx fifo is empty, 0 is not empty Data (1) --rx fifo full: 0 rx fifo is not full, 1 is full Data (2) – rx crc match: 1 match 1, not match 0 Data (3) – rx input error: 1 more than 6 consecutive '1', 0 no error Data (4) – rx address error: 1 address no match, 0 address match Data (5) – rx frame size error: 1 frame size over limit, 0 no error Data (6) – tx fifo empty: 1 fifo empty, 0 fifo not empty Data (7) – tx fifo almost full: 1 fifo almost full, 0 fifo not almost full Data (8) – eof flag: 1 eof arrived, 0 no eof

Table 4. HDLC IP register description

4.2 Functional Timing

Data can be written in FIFO by the user, when the ready_for_data_o signal is '1'.

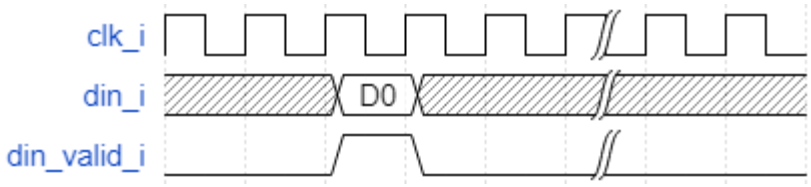

Figure 4. Writing input data in FIG0

On RX side the user can check the interrupt_o output signal (see Figure 5) or the rxempty_o (output FIFO empty signal, see Figure 6) or read the status register and monitor the RX empty flag, polling the status.

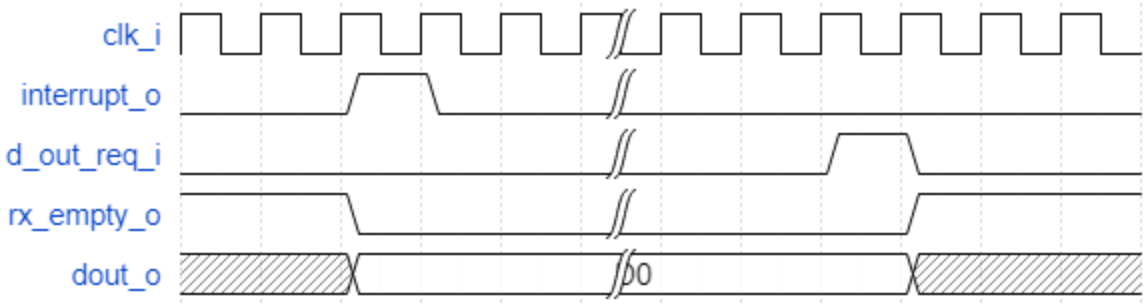

Figure 5. Interrupt monitoring. The user can read after the interrupt at any time

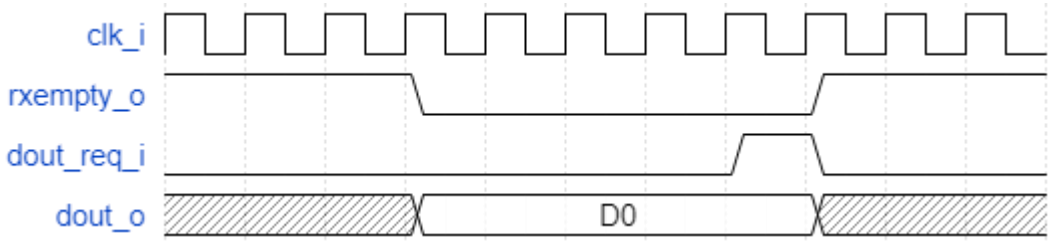

Figure 6. Polling the status register of send a read request when the data is available

5 SIMULATION

The HDLC modules are all tested on hardware and simulations, with self-checking assertions.

An example VHDL test bench is provided for use in a suitable VHDL simulator.

6 IMPLEMENTATION STATISTICS

The following table shows some synthesis tests. Examples are given of the resource utilization in selected standard FPGAs. The tool that has been for the synthesis evaluation is Mentor Graphics Precision RTL Plus 2018.1.0.18:

Please ask your local distributor or contact us if you need resource usage for other devices.

Family Device	Fmax (MHz)	LCs	Registers	Memory	Mult/DSP
Intel MAX 10	223	213 (ALMs)	194	2304 (Memory Bits)	0
Xilinx Artix 7	279	287 (LUTs)	208	1 BRAM 4 Distributed RAMs	0

7 PRODUCT SUPPORT

3 months maintenance with

- Delivery of the IP Core and documentation updates, minor and major versions changes
- Email support

8 ORDERING INFORMATION

Contact info@campera-es.com or your local distributor.