

# AvnetCore: Datasheet

Version 1.0, July 2006

## Single-Channel HDLC Controller

### Intended Use:

- Frame Relay
- ISDN and X.25 protocols
- Logic consolidation

### Features:

- Conforms to International Standard ISO/IEC 3309 Specification
- Starting point for a custom design
- 16-bit/32-bit CCITT-CRC generation and checking
- Flag & Zero insertion and detection
- Full Duplex Operation allowed
- DC to 53 Mbps (STS-1) data rate
- Full synchronous operation
- Interface can be customized for user FIFO and DMA Requirements

### Targeted Devices:

- Accelerator<sup>®</sup> Family
- ProASIC<sup>PLUS</sup><sup>®</sup> Family
- A54SXA Family
- ProASIC<sup>®</sup>3 Family

### Core Deliverables:

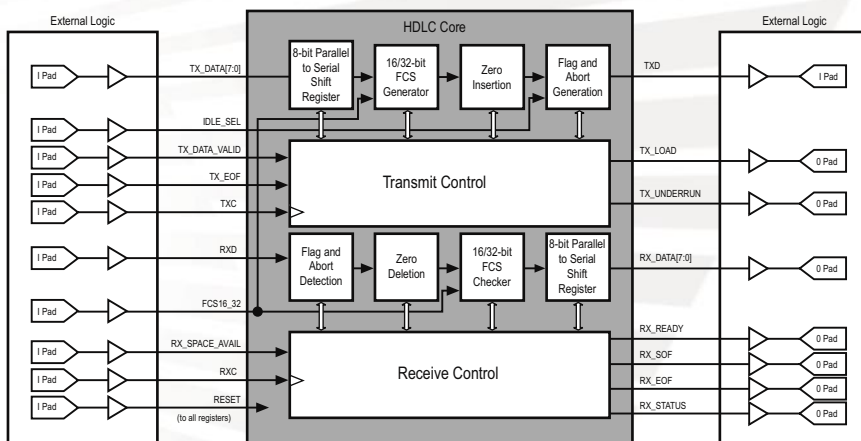
- Netlist Version
  - > Compiled RTL simulation model, compliant with the Actel Libero<sup>®</sup> environment
  - > Netlist compatible with the Actel Designer place and route tool
- RTL Version
  - > Verilog Source Code
  - > VHDL Source Code
- All
  - > User Guide
  - > Test Bench

### Synthesis and Simulation Support:

- Synthesis: Synplicity<sup>®</sup>
- Simulation: ModelSim<sup>®</sup>
- Other tools supported upon request

### Verification:

- Test Bench
- Test Vectors



Block Diagram

The MC-ACT-HDLC performs the most common functions of an HDLC controller. Data bytes are clocked into the device based on a divided version of the transmit clock. This data is then serialized and framed according to the rules of HDLC and sent out the serial transmit data pin. Receive frames are clocked into the receive data pin synchronous to the receive clock. The framing overhead is then stripped off and the data bytes are converted from serial to parallel and passed on through the parallel receive bus.

## Functional Description

### TRANSMITTER

The transmitter portion of the HDLC core will begin to transmit when the user's external logic asserts the TX\_DATA\_VALID signal. The transmitter will respond by asserting the TX\_LOAD signal to load the first byte of the packet. The timing diagram assumes that IDLE\_SEL is tied to a '1' and the transmitter is generating continuous '1' bits between frames. If IDLE\_SEL is set to a '0', the number of clocks from the assertion of TX\_DATA\_VALID to TX\_LOAD will vary from 5 to 12. Before the transmitter can begin to send data serially, it must send an opening flag (7E). Immediately after the flag is sent, the first byte is clocked out of the input shift register. Once a transmit frame has begun, the user is required to make sure that data is available for each subsequent requested byte. The transmitter will continue to request data by asserting TX\_LOAD until the user supplies a TX\_EOF signal. This informs the transmitter that the last byte is on the data bus. The transmitter then appends a 16- or 32-bit Frame Checking Sequence (FCS) to the transmitted data. After the FCS is sent, a closing flag (7E) byte is appended to mark the end of the frame. The HDLC Transmitter consists of the following blocks as shown in the block diagram.

#### 8-bit Parallel-to-Serial Shift Register

This block is responsible for capturing the user's transmit data on the rising edge of TXC when the TX\_LOAD signal is asserted. Data is sent to the TXD pin and the FCS Generator at the same time.

#### 16/32-bit FCS Generator

The Frame Checking Sequence (FCS) Generator is used to calculate a CRC across the transmitted message. Two different polynomials can be selected by statically controlling the FCS16\_32 pin. The 16-bit FCS uses the polynomial  $x^{16} + x^{12} + x^5 + 1$  and is selected when the FCS16\_32 pin is a logic LOW. The 32-bit FCS uses the polynomial  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$  and is selected when the FCS16\_32 pin is a logic HIGH. Either type of FCS is complemented before being transmitted.

#### Zero Insertion

The transmitter is responsible for examining the frame content between the opening and closing flags and checking for 5 consecutive '1' bits, including the FCS bits. If 5 consecutive '1' bits are detected, a '0' bit is inserted into the serial transmission. This will allow the receiver to distinguish between an opening or closing flag and actual data.

#### Flag and Abort Generation

An opening flag is sent when the user asserts the TX\_DATA\_VALID signal. As soon as the last byte of the FCS has been transmitted, a closing flag is sent. If a transmission has been started and the TX\_DATA\_VALID signal is deasserted while the transmitter is requesting another byte, an underrun condition will occur. This condition will be reported with the TX\_UNDERRUN pin, but will also result in the transmitter sending 8 consecutive '1' bits. This is defined as an "abort" condition. The transmitter will inter-frame fill by driving out a contiguous stream of '1' bits or a repeating flag. If back-to-back frames are available to send (the user continues to assert TX\_DATA\_VALID), the transmitter will share the closing flag of the first frame with the opening flag of the second frame.

#### Serial Output

All data exits the transmitter on the TXD pin and transitions on the rising edge of TXC.

#### Transmit Control

The control state machines and interface timing for the transmitter are driven by the rising edge of TXC.

### RECEIVER

The receiver clocks serial HDLC frames in continuously through the RXD pin. When an opening flag is recognized, the receiver locks to all subsequent octet bytes. The user informs the receiver of the ability to store the frame by asserting the RX\_SPACE\_AVAILABLE input. The receiver informs the user that a data byte is available by asserting the RX\_READY signal. The receiver indicates the beginning of the frame by asserting the RX\_SOF signal. Bytes will continue being passed to the user until the receiver recognizes the closing flag. At this point, the last byte of the FCS sequence will be passed to the user coincident with the RX\_EOF signal. It must be stressed that the core does not contain the additional pipeline registers to "swallow" the 2 or 4 bytes of FCS, and these will therefore be passed on to the user. If this is undesirable, the corresponding pipeline should be added externally to keep these bytes from passing on as part of the received frame. After the reception of the frame has completed, the receiver will pass a byte of status information to the user by placing the status on the receive data bus and asserting the RX\_STATUS signal. The Receiver consists of the following blocks as shown in the block diagram.

#### Flag and Abort Detection

The receiver begins operation by hunting for an opening flag character. Once the flag has been recognized, the receiver begins to receive the incoming frame, but continues to monitor for a closing flag. Once the closing flag has been detected, the frame is complete. Once the receiver has detected an opening flag, it will monitor the serial data stream to see if 8 consecutive '1' bits are detected. This condition is defined as a receive abort and is reported to the user through a receive status bit. The receiver is capable of handling back-to-back frames where the closing flag of the first frame also acts as the opening flag of the second frame. The receiver will idle on either contiguous '1' bits or repeating flag characters.

#### Zero Detection

The receiver checks the incoming data frame to see if 5 consecutive '1' bits are received. If this condition is detected, the following zero is deleted from the incoming frame.

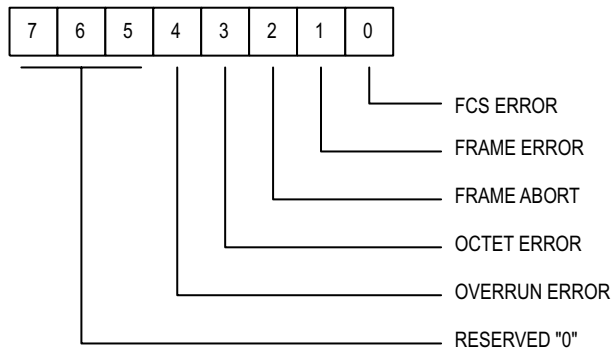
### 16/32-BIT FCS CHECK

The Frame Checking Sequence (FCS) Checker performs the same generator polynomial division as the transmitter across the entire transmitted message including the FCS field. The result of this polynomial division will be a constant remainder indicating the packet integrity. The receiver supports the same 16-bit and 32-bit FCS as the transmitter. The version is statically selected using the FCS16\_32 pin, the 16-bit version is selected by a logic LOW and the 32-bit version is selected with a logic HIGH.

### 8-BIT SERIAL SHIFT REGISTER

As serial data is clocked into the receiver, it is assembled back into bytes through a serial-to-parallel shift register. The receiver informs the user of a valid byte by asserting the RX\_READY signal. RX\_READY can be further qualified with additional signals to help the user track the progress of an incoming frame. RX\_SOF is asserted coincident with RX\_READY to indicate reception of the first octet of a frame. RX\_EOF is asserted coincident with RX\_READY to indicate the last byte of the receive FCS. The RX\_STATUS signal is asserted coincident with RX\_READY to indicate to the user that the receive data contains a valid byte of status information.

#### Status Byte



MDS3003

The status byte will be presented to the user at the end of the frame or after a receive error is detected. The receiver will inform the user of valid status on the RX\_DATA bus by the coincident assertion of RX\_READY and RX\_STATUS. The FCS ERROR will be set at the end of the frame if the remainder after polynomial division does not match the proper 16-bit or 32-bit constant.

The FRAME ERROR status bit will be set if a frame is received that is shorter than 32 bits when using the 16-bit FCS, and shorter than 48 bits when using the 32-bit CRC. There is no test to check for frame lengths that exceed a certain length. This bit will also be set when the OCTET ERROR is set.

The FRAME ABORT status bit will be set if the receiver has detected 8 consecutive '1' bits in a row after frame reception has begun.

The OCTET ERROR status bit is set whenever the closing flag is received on an odd bit boundary. The receiver tests to make sure all frames are an integral number of octets.

All remaining status bits are reserved and will be presented as '0'.

#### Receive Control

The control state machines and interface timing for the receiver is driven by the rising edge of RXC.

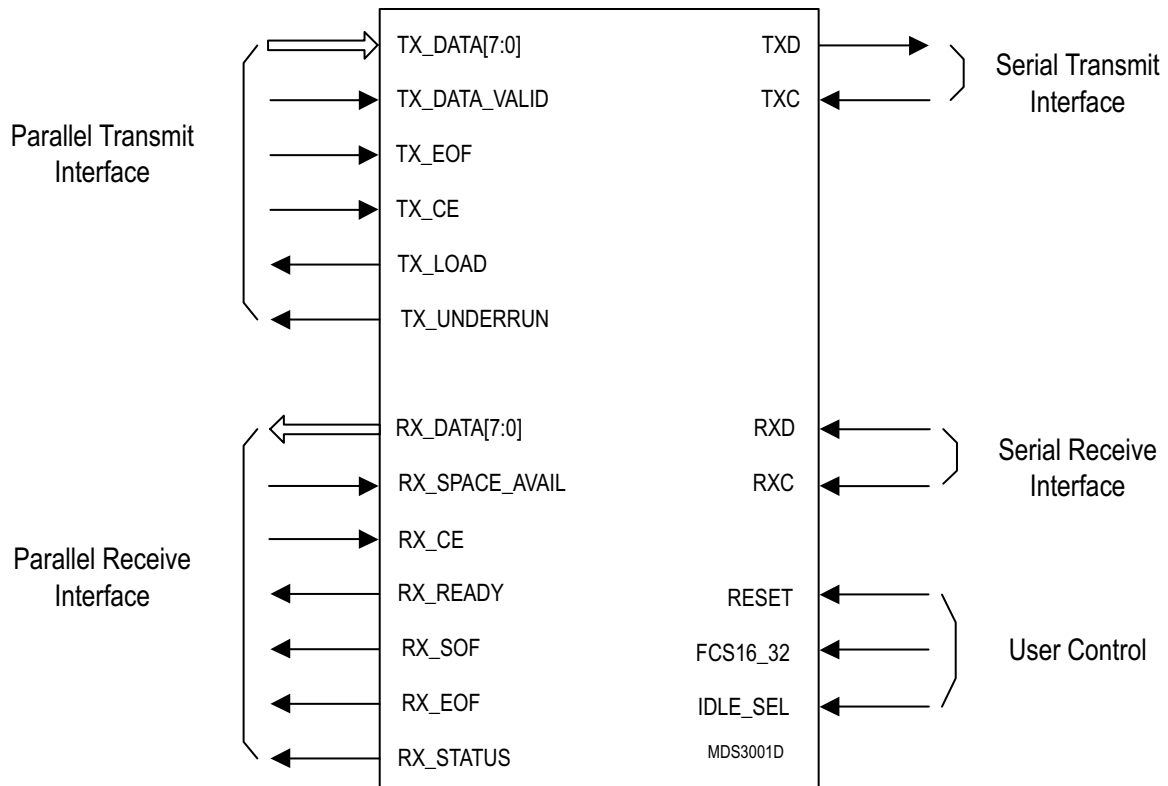


Figure 1: Logic Symbol

## Device Requirements

Family	Device	Utilization			Performance
		COMB	SEQ	Tiles	
Axcelerator	AX500	279	190	n/a	105 MHz
ProASIC <sup>PLUS</sup>	APA075	n/a	n/a	794	85 MHz
A54SXA	A54SX16A	315	191	n/a	92 MHz
ProASIC3	A3P250	n/a	n/a	590	120 MHz
ProASIC3E	A3PE600	n/a	n/a	590	124 MHz

Table 1: Device Utilization and Performance

## Verification and Compliance

Functional and timing simulation has been performed on the HDLC using VHDL and Verilog Test Benches. Simulation vectors used for verification are provided with the core. The HDLC core has been hardware tested with the TTC Fireberd 6000A Frame Relay Option. This core has also been used successfully in customer designs.

## Signal Descriptions

The following signal descriptions define the IO signals.

Signal	Direction	Description
TX_DATA[7:0]	Input	Transmitter Parallel Data Bus: 8-bit transmit data bus loaded synchronously based on the TX_LOAD signal and the TXC clock. This bus is driven by the user's transmit FIFO or RAM buffer.
TX_DATA_VALID	Input	Transmit Data Valid: An active high user input, synchronous to TXC, to inform the transmitter that an external packet is ready to send.
TX_EOF	Input	Transmit End-Of-Frame: An active high user input pulse, synchronous with TXC, to inform the transmitter that the current data byte is the last byte of a sending packet. This input should be coincident with TX_LOAD.
TX_CE	Input	Transmit Clock Enable: An active high user input, synchronous with TXC.
RESET	Input	Global Reset: Asynchronously resets all internal registers.
IDLE_SEL	Input	Idle Select: Selects the inter-frame idle fill type. When tied low the device sends continuous flags between frames and when tied high the device sends continuous ones between frames.
FCS16_32	Input	FCS Select: Selects the 16-bit FCS, $x^{16} + x^{12} + x^5 + 1$ , when tied low. Selects the 32-bit FCS, $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ , when tied high.
RX_SPACE_AVAIL	Input	Receive Space Available: An active high user input, synchronous to RXC, to inform the receiver that the external receive FIFO or buffer RAM can accept more data.
TXC	Input	Serial Transmit Clock: A user provided clock for all transmit activity. All transmit functions take place on the low to high transition of TXC.
RX_CE	Input	Receive Clock Enable: An active high user input, synchronous with RXC.
RXD	Input	Serial Receive Data: An input for serial receive data, sampled on the rising edge of RXC.
RXC	Input	Serial Receive Clock: A user provided clock for all receive activity. All receive functions take place on the low to high transition of RXC.
TX_UNDERRUN	Output	Transmit Underrun: An active high output pulse, synchronous to TXC, from the transmitter indicating an underrun error. This occurs, after the start of frame transmission, if TX_DATA_VALID is deasserted when TX_LOAD is asserted.
TX_LOAD	Output	Transmit Load: An output pulse from the transmitter, synchronous to TXC, that acts as a clock enable signal to the external transmit buffer to request an input byte.
TXD	Output	Serial Transmit Data: Provides the serial transmit data and transitions on the rising edge of the TXC clock.
RX_EOF	Output	Receive End-Of-Frame: An active high pulse, synchronous to RXC, to inform the user that the current receive byte is the last byte (either 2 or 4) of the Frame Checking Sequence. This pulse is coincident with the RX_READY pulse.
RX_STATUS	Output	Receive Status: An active high pulse, synchronous to RXC, to inform the user that receive frame status is being output on the RX_DATA bus. RX_STATUS is coincident with the RX_READY signal.
RX_SOF	Output	Receive Start-Of-Frame: An active high pulse, synchronous to RXC, to inform the user that the current receive data byte is the first byte of a frame. This pulse is coincident with the RX_READY pulse.
RX_DATA[7:0]	Output	Receive Parallel Data Bus: 8-bit receive data bus providing the user output data synchronous to RX_READY and the RXC clock. This bus is tied to the user's receive FIFO or RAM buffer. This same bus is used to report frame status at the end of a receive.
RX_READY	Output	Receive Ready: An active high pulse from the receiver, synchronous to RXC, that acts as a clock enable signal to the external receive buffer to output a received byte. The STATUS pin distinguishes receive data from frame status.

Table 2: Core I/O Signals

## Timing

Since the ATM Forum specification fully defines the line side of the UTOPIA Level 3 interface, timing for that is not replicated here. Instead, only user (FIFO) interface timing information is presented here. The figure below shows the functional timing for FIFO reads and writes.

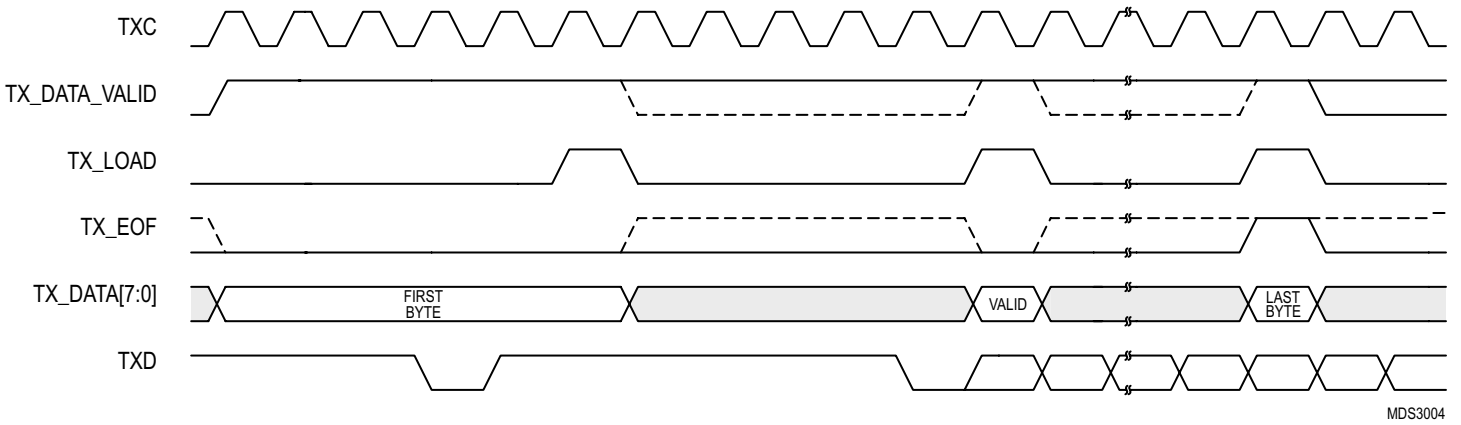


Figure 3: Transmit Timing

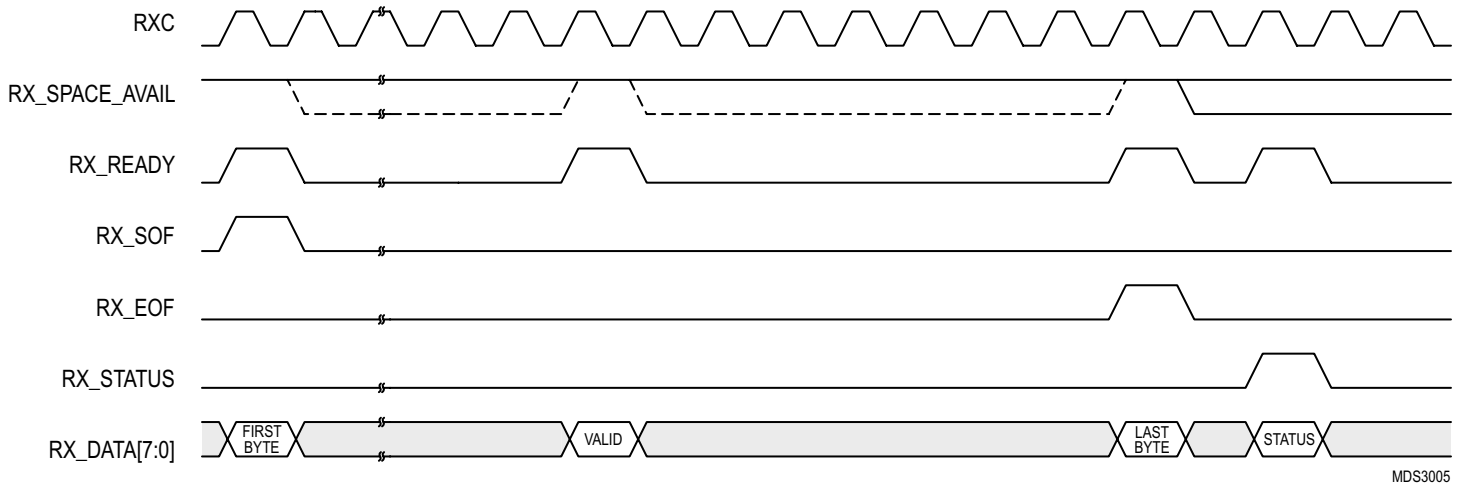


Figure 4: Receive Timing

## Recommended Design Experience

For the source version, users should be familiar with HDL entry and Actel design flows. Users should be familiar with Actel Libero Integrated Design Environment (IDE) and preferably with Synplify and ModelSim.

## Ordering Information

The CORE is provided under license from Avnet Memec for use in Actel programmable logic devices. Please contact Avnet Memec for pricing and more information.

Information furnished by Avnet Memec is believed to be accurate and reliable. Avnet Memec reserves the right to change specifications detailed in this data sheet at any time without notice, in order to improve reliability, function or design, and assumes no responsibility for any errors within this document. Avnet Memec does not make any commitment to update this information.

Avnet Memec assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction, if such be made, nor does the Company assume responsibility for the functioning of undescribed features or parameters. Avnet Memec will not assume any liability for the accuracy or correctness of any support or assistance provided to a user.

Avnet Memec does not represent that products described herein are free from patent infringement or from any other third-party right. No license is granted by implication or otherwise under any patent or patent rights of Avnet Memec.

AvnetCore products are not intended for use in life support appliances, devices, or systems. Use of a AvnetCore product in such application without the written consent of the appropriate Avnet Design officer is prohibited.

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

### Contact Information:

#### North America

10805 Rancho Bernardo Road  
Suite 100  
San Diego, California 92127  
United States of America  
TEL: +1 858 385 7500  
FAX: +1 858 385 7770

#### Europe, Middle East & Africa

Mattenstrasse 6a  
CH-2555 Brügg BE  
Switzerland  
TEL: +41 0 32 374 32 00  
FAX: +41 0 32 374 32 01

### Ordering Information:

#### Part Number

**MC-ACT-HDLC-NET**  
**MC-ACT-HDLC-VLOG**  
**MC-ACT-HDLC-VHDL**

#### Hardware

Actel HDLC Controller Netlist  
Actel HDLC Controller Verilog  
Actel HDLC Controller VHDL

#### Resale

Contact for pricing  
Contact for pricing  
Contact for pricing



[www.em.avnet.com/actel](http://www.em.avnet.com/actel)