

HDLC Protocol Controller

High-Level Data Link Control Protocol

Overview

The HDLC Protocol Controller IP core implements High-Level Data Link Control (HDLC) protocol and its derivatives such as LAPB or LAPD.

Link Access Procedure – Balanced (LAPB), the subset of HDLC protocol is used for the public networks that use the X.25 communications protocol. Link Access Procedure Channel D (LAPD) is intended for the ISDN implementations.

Functional features of HDLC Protocol Controller IP core are based on Siemens HSCX 82525 chip. Therefore programs written for HSCX chip can be used with HDLC Protocol Controller IP core with minor changes.

Benefits

- ◆ Hardware support for LAPB/LAPD means only minor frame processing in the target microcontroller
- ◆ Support for an external DMA controller speeds up data exchange between CPU and HDLC Protocol Controller
- ◆ Built-in timer for retransmission
- ◆ Support for line speeds of up to 10Mbit/s in FPGA/ASIC projects
- ◆ Shared single interrupt line for 11 internal interrupts means no need for complicated interrupt controllers in CPU
- ◆ Easy channel multiplication (dual channel version available as standard delivery on request)

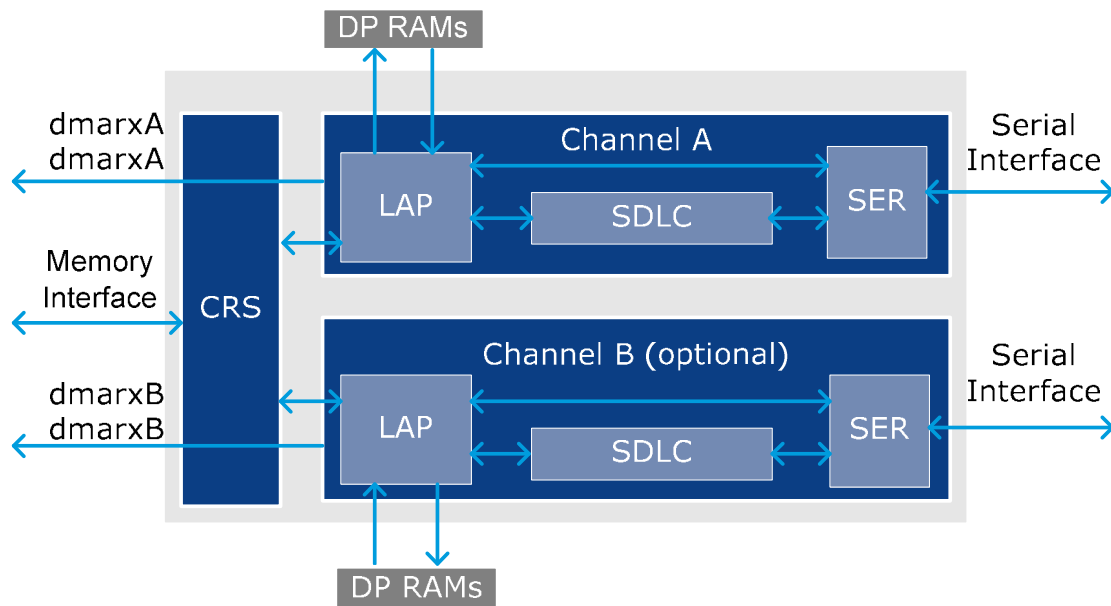
Applications

- ◆ X.25 link control
- ◆ ISDN applications
- ◆ Physical link maintenance and quality monitoring of wide area networks
- ◆ General purpose telecommunication applications

Features

- ◆ LAPB/LAPD controlling machine providing
 - modulo 8 frame numbering
 - modulo 128 frame numbering
 - one- or two-byte addressing
 - automatically generated responses
- ◆ PLB and generic system bus interfaces
- ◆ Serial Peripheral Interfaces
 - Bit stuffing
 - BOF and EOF flags generation
 - Support for RTS/CTS modem lines
 - Support for CD modem line
 - Collision detection in bus configuration
- ◆ Receive Length Check
- ◆ Three modes of receive operation
 - auto mode (with address recognition and control field insertion)
 - non-auto mode (with address recognition)
 - transparent mode (without address recognition)
- ◆ Receive transmit blocks
 - Interrupt transfer mode
 - DMA transfer mode
- ◆ Separate FIFOs
 - 64-bytes long receive FIFO
 - 64-bytes long transmit FIFO
 - 16-bytes long address FIFO for storing up to 16 small frames in the receive FIFO
- ◆ Single or dual independent channel versions
 - Separate DMA lines for each channel
 - Common data bus and interrupt line

Block diagram



Functional description

The HDLC Protocol controller is partitioned into modules as shown in figure above and described below.

CSR

Converts signals on the host interface to the internal SFR interface. Switches data and memack between channels. Host interface is similar to external memory interface that is used by the R8051XC2 IP core.

LAP

Controls the transmission and provides LAPB/LAPD support. LAP has internal timer and FIFO's for data buffering. It generates all interrupts and DMA requests. There are 11 interrupt flags sources per channel to determine the exact source of interrupt.

LAP component integrates:

- ◆ Internal timer
- ◆ HDLC protocol engine
- ◆ 64B FIFO in transmit direction
- ◆ 64B FIFO in receive direction
- ◆ 16B FIFO for multiple frame storage in receive direction
- ◆ Special Function Registers section for configuration
- ◆ Engine for automatic address and control field insertion
- ◆ Transfer engines for interrupt and DMA transfers

SDLC

Provides serial data coding/decoding, bit stuffing, BOF and EOF generation, address recognition and CRC check/generation. It has small FIFO queues in both directions. The FIFO's are 3-bytes deep and can be implemented as registers.

SER

Provides support for flow control and bus configuration. It also provides:

- ◆ RTS/CTS flow control
- ◆ CD sense for enabling/disabling receiver
- ◆ collision detection in bus mode
- ◆ bus IDLE state detection
- ◆ four different clock modes

Serial Interface

Interface consists of serial input and output lines, rts, cts, cd. cts input can be used as a feedback from the bus to detect collision. It provides driver enable line for the external tristate buffer.

That line can be programmed to go active while frame is being transmitted or only when "0" is transmitted. There are also clock inputs that are used in several clock modes. Those clocks (rxclk and txclk) are internally synchronized with the main host clock. This means that there is only single global clock in the design.

AFIFO, RFIFO, TFIFO interfaces

These are interfaces to DP RAM memories that should be implemented on the chip level. DP RAM memories are used by FIFO's in the HDLC Protocol Controller IP core. RFIFO and TFIFO have a 6-wire address buses and 8-bit data buses. AFIFO needs 4-wire address buses and 8-bit data buses.

Pin Description

Name	Type	Polarity/ Bus size	Description
General host interface signals			
clk	in	rise	Clock
rst	in	high	Reset
Host Interface			
memrd	in	high	Read request
memwr	in	high	Write request
memaddr	in	8	Address bus
datai	in	8	Data input
datao	out	8	Data output
int	out	high	Interrupt line
memack	out	high	Memory acknowledge
dmarxA	out	high	Rx request channel A
dmarxB	out	high	Rx request channel B ¹
dmatxA	out	high	Tx request channel A
dmatxB	out	high	Tx request channel B ¹
Serial Interface – Channel A			
rxdA	in	1	Data input
rxclkA	in	1	Receive clock
ctsA/cxdA	in	low	Clear To Send / Collision Detection
rxstrA	in	high	Receive Strobe
txstrA	in	high	Transmit Strobe
cdA/frsyncA	in	high	Carrier detect / Frame Sync
txdA	out	1	Data output
txclkA	out	1	Transmit clock
rtsA	out	low	Request To Send
denA	out	low	External driver enable
Serial Interface – Channel B¹			
rxdB	in	1	Data input
rxclkB	in	1	Receive clock
ctsB/cxdB	in	low	Clear To Send / Collision
rxstrB	in	high	Receive Strobe
txstrB	in	high	Transmit Strobe
cdB/frsyncB	in	high	Carrier detect / Frame Sync
txdB	out	1	Data output
txclkB	out	1	Transmit clock
rtsB	out	low	Request To Send
denB	out	low	External driver enable
AFIFO Interface – Channel A²			
adibA	in	8	Data Input
aweaA	out	high	Write Enable
aenbA	out	high	Read Enable
aaddrA	out	4	Write Address
aaddrbA	out	4	Read Address
adoaA	out	8	Data Output

Name	Type	Polarity/ Bus size	Description
RFIFO Interface – Channel A²			
rdibA	in	8	Data Input
rweaA	out	high	Write Enable
renbA	out	high	Read Enable
raddrA	out	4	Write Address
raddrbA	out	4	Read Address
rdoaA	out	8	Data Output
TFIFO Interface – Channel A²			
tdibA	in	8	Data Input
tweaA	out	high	Write Enable
tenbA	out	high	Read Enable
taddrA	out	4	Write Address
taddrbA	out	4	Read Address
tdoaA	out	8	Data Output

Notes:

- 1 Optional, for dual channel version only
- 2 The same set of signals for optional Channel B

Verification Methods

The functional verification of the synthesizable core is performed with the HDL testbench.

The top level of the testbench contains a synthesizable block of the MAC and additional environment such as stimulus vectors generator and output vectors comparator.

To support top-down verification, the Aldec Active-HDL and MTI ModelSim were used to perform the functional simulation based on the set of tests.

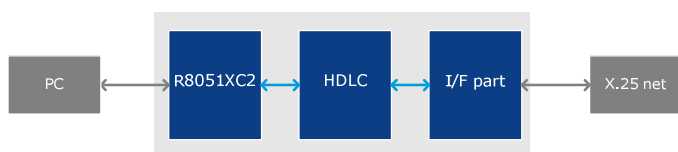
Xilinx Virtex-E and Altera Stratix technologies were used as target for post route verification. Mentor Graphics Leonardo Spectrum, Xilinx Foundation ISE, Synopsys Design Compiler and Altera Quartus tools are used to generate the post-route EDIF netlists.

Configurability

There are two top-level architectures available, both allowing implementing single or double channel core.

Please contact Evatronix directly to discuss any specific requirements.

Example Application



The example application presents usage of HDLC Protocol Controller IP core as support for X.25 network. It handles LAPB protocol. The HDLC Protocol Controller IP core is implemented together with internal dual port RAM. The R8051XC2 microcontroller is used as a support for PLP.

Related Products

HDLC Connectivity Platform – an application platform that integrates the R8051XC2 microcontroller with the HDLC .

Standard Deliverables

- ◆ Verilog or VHDL source code
- ◆ Synthesis support (Synopsys[®], Cadence[®]) with a complete set of synthesis scripts
- ◆ Simulation support (Mentor Graphics[®], Cadence[®]) with a set of scripts and macros
- ◆ Extensive HDL Test Bench that instantiates:
 - Example HDLC Protocol Controller design chip for single channel version
 - Test Bench Environment design
 - External DP RAM
 - Clock generator
 - Monitors that compares your simulation results with the expected results
 - Stimulators for simulating host and serial devices
- ◆ A set of tests for functional and post-route testing with corresponding set of expected results
- ◆ Documentation:
 - Design Specification
 - Verification Specification and Test Plan
 - Integration Manual with User Guide
- ◆ PLB version can be delivered with following add-ons:
 - HDLC Peripheral module for Xilinx EDK tool
 - Xilinx Platform Studio Quick Startup Guide
 - HDLC Self-test software procedure

Delivery Options

The following options may be ordered according to user's requirements.

- ◆ EDIF netlist for FPGA and low volume production
- ◆ Annual maintenance
- ◆ On-site support