Dual HDLC

Disclaimer

Systemyde International Corporation reserves the right to make changes at any time, without notice, to improve design or performance and provide the best product possible. Systemyde International Corporation makes no warrant for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make any commitment to update the information contained herein.

Systemyde International Corporation products are not authorized for use in life support devices or systems. Nothing contained herein shall be construed as a recommendation to use any product in violation of existing patents, copyrights or other rights of third parties. No license is granted by implication or otherwise under any patent, patent rights or other rights, of Systemyde International Corporation. All trademarks are trademarks of their respective companies.

Every effort has been made to ensure the accuracy of the information contain herein. If you find errors or inconsistencies please bring them to our attention. In all cases, however, the Verilog HDL source code for the dualhdlc.v design defines “proper operation”.

Index

Revision History

Date	Description	Page(s)
09/14/2011	typos	8, 15, 16, 17
	changed some registers to read-only to save logic	13
	pinout change, due to FPGA requirements	11
10/12/2011	updated Chapter 4 with more details	13-18
	added Chapter 5	19-22
11/15/2011	Completed timing digrams	19-23

Overview

The Dual HDLC controller provides two full-duplex HDLC channels, each with 512-byte data FIFO buffers for both directions. Recognizing that in most applications the various HDLC options rarely change, the design uses pins to control these options rather than dedicated control registers. This essentially eliminates initialization requirements.

General Features:

- Optimized for an Actel A3P060 in a 100-pin VQFP package.

- Also suitable for embedding in an A2F060 or larger.

- 512 byte FIFO buffers for both receive and transmit in each channel, using the FPGA memories.

- 8-bit bus interface.

- 8-bit Baud Rate Generator in each channel.

- Serial clock is independent between channels.

HDLC Features:

- Automatic Flag generation and checking.

- Automatic zero-insertion and deletion.

- Automatic 16-bit CRC generation and checking.

- Optional 16-bit frame preamble.

- Idle line of Flags or Always High.

- Optional 8-bit address compare, with two address registers per channel.

- DPLL for clock recovery (runs at 16x or 32x data rate).

- Data encode/decode: NRZ, NRZI, Biphase-Level, Biphase-Mark or Biphase-Space.

Pin Descriptions

When the Dual HDLC controller is implemented in an Actel A3P060 or A3P125 a number of pins are predefined by Actel. Refer to the Actel documentation for the A3P060 and A3P125 for details about these pins.

Listed here are the pins specific to the Dual HDLC controller design. These pins can be arranged into three separate groups: the bus interface, mode control, and the serial interface.

Since logic resources are limited in an FPGA, this design does not use registers to set the majority of the options. Normally these options do not change dynamically anyway, so they can be tied either High or Low on the printed circuit board. If any of these mode control pins do need to change under software control, there are eight simple outputs that can be wired to control inputs as required by the user.

Bus Interface group:

M_ADDR_BUS[4:0] (inputs). The Address Bus selects the channel, as well as the register within the channel, according to the tables below:

M_ADDR_BUS	Register
00000	Serial A Data
00001	Serial A Last Data
00010	Serial A Common Status
00100	Serial A Receive Control
00110	Serial A Transmit Constant
01000	Serial A Broadcast Address
01010	Serial A Station Address
01100	Serial A Control Byte
01110	Serial A Divide Constant
10000	Serial B Data
10001	Serial B Last Data
10010	Serial B Common Status
10100	Serial B Receive Control

10110	Serial B Transmit Control
11000	Serial B Broadcast Address
11010	Serial B Station Address
11100	Serial B Control Byte
11110	Serial B Divide Constant

M_CLK (input, active-High). The Clock drives both the bus interface and serial interface portions of the design.

M_CSB (input, active-Low). The Chip Select selects the device for a read or write.

M_DATA_BUS[7:0] (bidirectional). The Data Bus is ysed to transfer data to or from the device. The Data bus is normally floating, and will only be active while both Chip Select and Read Strobe are Low.

M_RDB (input, active-Low). The Read Strobe, in conjunction with the Chip Select, enables the device for a read transaction.

M_RESETB (input, active-Low). The Master Reset completely initializes the device.

M_WRB (input, active-Low). The Write Strobe, in conjunction with the Chip Select, enables the device for a write transaction.

A_INTB, B_INTB (outputs, active Low). The Interrupt Request output is active when the serial channel is requests an interrupt. The specific type of interrupt request can be found by reading the Common Status register. This output will be de-asserted when the interrupting condition has been removed.

A_REOF, B_REOF (outputs, active High). The Receive End-Of-Frame output goes active when the data byte read from the receive buffer is the last byte in a received frame. This status information can be used to terminate a block transfer from the receive buffer. The Receive End-Of_frame output is cleared when the receive buffer is purged or when the next byte (from a subsequent frame) is read from the receive buffer.

Mode Control group:

A_ABRT_EN, B_ABRT_EN (inputs, active High). The Abort Enable input controls the operation of the receiver with respect to frames terminated by an Abort. Normally the receiver discards such frames by purging the receive buffer, but if this input is High aborted frames will be transferred to the receive buffer.

A_BRG_EN, B_BRG_EN (inputs, active High). The Baud Rate Generator Enable input controls the operation of the baud rate generator. Normally the baud rate generator is disabled, but if this input is High the baud rate generator is enabled and the Baud Rate Generator Output signal toggles. Note that the output of the baud rate generator is not connected to the serial port internally, but must be connected externally to clock the serial port.

A_CLK_EN, B_CLK_EN (inputs, active High). The Clock Enable input controls the outputs for the Receive Clock and the Transmit Clock outputs. Normally these outputs are disabled to save power, but if this input is High both of these clock outputs are enabled.

A_CRC_DIS, B_CRC_DIS (inputs, active High). The CRC Disable input controls the transmission of CRC at the end of a frame. Normally CRC is always appended to a transmit frame, but if this input is High CRC will not be appended. This input is only sampled at the end of a transmit frame at the point where the yes/no decision is made.

A_CRCERR_EN, B_CRCERR_EN (inputs, active High). The CRC Error Enable input controls the operation of the receiver with respect to frames received with a CRC error. Normally the receiver discards such frames by purgin gthe receive buffer, but if this input is High frames with CRC errors will be transferred to the receive buffer.

A_DEC_MODE[2:0], B_DEC_MODE[2:0] (inputs). The Decode Mode inputs select the data encoding used by both the receiver and the transmitter, according to the table below:

DEC_MODE	Data encoding
000	NRZ
010	NRZI
100	Biphase-Level
110	Biphase-Space
111	Biphase-Mark

A_DPLL_32X, B_DPLL_32X (inputs, active High). The Digital Phase Locked Loop 32x input controls the sampling rate for the digital phase locked loop. Normally the digital phase locked loop operates at 16 times the data rate, but if this input is High the operation is 32 times the data rate.

A_DPLL_EN, B_DPLL_EN (inputs, active High). The Digital Phase Locked Loop Enable input controls the operation of the digital phase locked loop. Normally

the digital phase locked loop is disabled, but if this input is High both the digital phase locked loop and the accompanying transmit prescaler are enabled and connected automatically to the receiver and transmitter.

A_FLAG_IDL, B_FLAG_IDL (inputs, active High). The Flag Idle input controls the idle line state for the transmitter. Normally the idle line condition is Mark (continuous unencoded ones), but if this input is High the transmitter will send continuous Flags between frames. This input is only sampled at the end of a transmit frame and every eight bit times thereafter until a new frame starts.

A_PREAM_EN, B_PREAM_EN (inputs, active High). The Preamble Enable input controls the transmission of a preamble word before each frame. Normally no preamble is transmitted, but if this input is High the transmitter will send the preamble pattern prior to the opening flag of a frame. This input is only sampled at the start of a transmit frame at the point where the yes/no decision is made. The table below shows the preamble pattern for each data encoding method.

DEC_MODE	Data encoding	Preamble
000	NRZ	0x5555
010	NRZI	0x0000
100	Biphase-Level	0x5555
110	Biphase-Space	0xFFFF
111	Biphase-Mark	0x0000

A_PORT[3:0], B_PORT[3:0] (outputs). The Port outputs are simple output bits controlled by the Receive Control register (bits 3-2) and the Transmit Control register (bits 1-0). The can be used for any purpose. Reset sets all of the outputs Low.

A_URUN_ABRT, B_URUN_ABRT (inputs, active High). The Underrun Abort input controls the transmitter response to an underrun. Normally an underrun condition will cause the transmission of the CRC and closing Flag, but if this input is High the transmitter will send an Abort (0x7F) in the case of an underrun. This input is only sampled at the time that an underrun occurs. If the CRC Disable input is High, then only a FLag or an Abort will be sent in the case of an underrun.

Serial Interface group:

A_BRG_OUT, B_BRG_OUT (outputs). The Baud Rate Generator output carries the square-wave output of the baud rate generator whenever it is enabled. This output is Low whenever the baud rate generator is disabled. To clock the serial port with the output of the baud rate generator, connect this pin to the Serial Clock input pin on the PC board.

A_SER_CLK, B_SER_CLK (inputs). The Serial Clock input carries the clock for the receiver and transmitter. If the digital phase locked loop is disabled this clock sets the data rate directly. If the digital phase locked loop is enabled this clock drives the digital phase locked loop and transmit divider, which run at either 16 or 32 times the data rate. When the digital phase locked loop is enabled the actual receive and transmit clocks may be output on the Serial Receive Clock and Serial Transmit Clock output pins.

A_SER_RCLK, B_SER_RCLK (outputs). The Serial Receive Clock output carries the clock that is being used by the receiver, at the receive data rate. This output is normally disabled (Low) unless the Clock Enable input is High.

A_SER_RTS, B_SER_RTS (outputs, active High). The Request To Send output carries a signal that can be used to enable a transmit driver in multi-drop applications. This output is active two bit times before the first bit of a frame (either the pre-amble or opening Flag) is valid on the Transmit Data output. This output remains active for two bit times after the last bit of a frame (either a Flag or Abort) has been completely transmitted. This output changes onthe falling edge of the transmit clock.

A_SER_RXD, B_SER_RXD (inputs). The Receive Data input carries the serial data for the receiver. The Receive Data input is sampled by the rising edge of the serial clock.

A_SER_TCLK, B_SER_TCLK (outputs). The Serial Transmit Clock output carries the clock that is being used by the transmitter, at the transmit data rate. This out-put is normally disabled (Low) unless the Clock Enable input is High.

A_SER_TXD, B_SER_TXD (outputs). The Transmit Data output carries the serial data from the transmitter. The Transmit Data changes on the falling edge of the serial clock.

Pin Assignments

When the design uses an Actel A3P060 or A3P125 in a 100-pin VQFP package these are the pin assignments:

pin	signal	pin	signal	pin	signal
1	GND	35	A_SER_TCLK	69	B_FLAG_IDL
2	A_ABRT_EN	36	A_SER_RCLK	70	B_DPLL_32X
3	A_CRC_DIS	37	Vcc	71	B_CRCERR_EN
4	A_CRCERR_EN	38	GND	72	B_CRC_DIS
5	A_DPLL_32X	39	VcciB1	73	B_ABRT_EN
6	A_FLAG_IDL	40	B_SER_RCLK	74	VMV0
7	A_DEC_MODE[2]	41	B_SER_TCLK	75	GNDQ
8	A_DEC_MODE[1]	42	B_SER_RXD	76	n.c.
9	GND	43	B_SER_TXD	77	B_REOF
10	A_DEC_MODE[0]	44	B_SER_RTS	78	A_REOF
11	A_CLK_EN	45	B_SER_CLK	79	M_WRB
12	Vcomplf	46	B_BRG_OUT	80	M_RDB
13	CLK	47	TCK	81	M_CSB
14	Vccplf	48	TDI	82	M_ADDR_BUS[0]
15	A_PORT[2]	49	TMS	83	M_ADDR_BUS[1]
16	A_PORT[1]	50	VMV1	84	M_ADDR_BUS[2]
17	Vcc	51	GND	85	M_ADDR_BUS[3]
18	VcciB1	52	Vpump	86	M_ADDR_BUS[4]
19	A_PORT[0]	53	NC	87	VcciB0
20	A_BRG_EN	54	TDO	88	GND
21	A_DPLL_EN	55	TRST	89	Vcc
22	A_PREAM_EN	56	Vjtag	90	M_DATA_BUS[0]
23	A_URUN_ABRT	57	B_URUN_ABRT	91	M_DATA_BUS[1]
24	VMV1	58	B_PREAM_EN	92	M_DATA_BUS[2]
25	GNDQ	59	B_DPLL_EN	93	M_DATA_BUS[3]
26	A_SER_CLK	60	B_BRG_EN	94	M_DATA_BUS[4]
27	B_PORT[2]	61	B_CLK_EN	95	M_DATA_BUS[5]
28	B_PORT[1]	62	B_DEC_MODE[0]	96	M_DATA_BUS[6]
29	B_PORT[0]	63	RESETB	97	M_DATA_BUS[7]
30	A_BRG_OUT	64	B_DEC_MODE[1]	98	B_INTB
31	B_PORT[3]	65	B_DEC_MODE[2]	99	A_INTB
32	A_SER_RTS	66	VcciB0	100	A_PORT[3]
33	A_SER_TXD	67	GND		
34	A_SER_RXD	68	Vcc		

Programming Interface

Registers

Register Name	Mnemonic	I/O address	R/W	Reset
Serial A Data Register	SADR	0x00	R/W	xxxxxxxx
Serial A Last Data Register	SALDR	0x01	R/W	xxxxxxxx
Serial A Common Status Register	SACSR	0x02	R	00000001
Serial A Receive Control Register	SARCR	0x04	W	00000000
Serial A Transmit Control Register	SATCR	0x06	W	00000000
Serial A Broadcast Address Register	SABAR	0x08	W	11111111
Serial A Station Address Register	SASAR	0x0A	W	11111111
Serial A Control Byte Register	SACBR	0x0C	R/W	00000000
Serial A Divide Constant Register	SADCR	0x0E	W	00010001
Serial B Data Register	SBDR	0x10	R/W	xxxxxxxx
Serial B Last Data Register	SBLDR	0x11	R/W	xxxxxxxx
Serial B Common Status Register	SBCSR	0x12	R	00000001
Serial B Receive Control Register	SBRCR	0x14	W	00000000
Serial B Transmit Control Register	SBTCR	0x16	W	00000000
Serial B Broadcast Address Register	SBBAR	0x18	W	11111111
Serial B Station Address Register	SBSAR	0x1A	W	11111111
Serial B Control Byte Register	SBCBR	0x1C	R/W	00000000
Serial B Divide Constant Register	SBDCR	0x1E	W	00010001

Register Descriptions

Serial x Data Register		(SADR)	(Address = 0x00)
		(SBDR)	(Address = 0x10)
Bit(s)	**Value**	**Description**	
7:0	**Read**	**Receive buffer data.** The receive buffer is 512 bytes deep and is only capable of holding data for one frame at a time, because the buffer is purged in the case of an error frame. If both CRC-errored and Aborted frames are enabled for transfer to the buffer, misaligned frames will also be accepted. In this case it is possible to allow the receive buffer to contain data from more than one frame.	
	Write	**Transmit buffer data.** The transmit buffer is 512 bytes deep and is capable of holding data for more than one frame at a time, as long as the end of each frame is marked as such. Each transmit frame requires a command to begin transmission.	

Serial x Last Data Register		(SALDR)	(Address = 0x01)
		(SBLDR)	(Address = 0x11)
Bit(s)	**Value**	**Description**	
7:0	**Read**	**Receive buffer data.**	
	Write	**Transmit buffer data (last byte of frame).** The last byte of a frame should be written to this register to mark it as the last in the frame. This enables subsequent transmission of the CRC and closing Flag. As an alternative, the transmitter can be enabled to automatically send the CRC and closing Flag on underrun. This option is only recommended when the entire frame can be guaranteed to be transferred to the buffer.	

Serial x Common Status Register		(SACSR) (Address = 0x02) (SBCSR) (Address = 0x12)
Bit(s)	**Value**	**Description**
7	**0**	**No receive interrupt pending.**
	1	**Receive Interrupt Pending.** This bit is set when the receiver completes a frame that is not reclaimed from the buffer. Bits 6-5 of this register are only valid while this bit is set. This bit is cleared by the Receive Interrupt Clear command. The INTB output is active (Low) while this bit is set.
6	**0**	**Frame received with error.** The frame transferred to the receive buffer either encountered a buffer overrun, contained misaligned data, had a CRC error, or was terminated by an Abort.
	1	**Frame received without error.** The frame transferred to the receive buffer had the correct CRC and was closed with a Flag. This bit is cleared by the Receive Interrupt Clear command.
5	**0**	**No receive buffer overrun.**
	1	**Receive buffer overrun.** A receive buffer overrun immediately sets both this bit and the Receive Interrupt Pending bit. In the case of an overrun the Abort Receive command should be issued to purge the receive buffer and force the receiver to search for a Flag. This bit is cleared by the Receive Interrupt Clear command.
4	**0**	**Receive buffer is empty.**
	1	**Receive buffer is not empty.** This bit indicates that data is available in the receive buffer. Note that the receive buffer will be purged by the receiver in the case of an errored frame, so care must be taken to ensure that a subsequent frame is not able to discard part or all of a valid frame.
3	**0**	**No transmit interrupt pending.**
	1	**Transmit Interrupt Pending.** This bit is set when the transmitter completes a frame. Bits 2-1 of this register are only valid while this bit is set. This bit is cleared by the Transmit Interrupt Clear command. The INTB output is active (Low) while this bit is set.
2	**0**	**Frame terminated with Abort.** This bit is cleared to indicate that the frame was terminated by an Abort.
	1	**Frame sent successfully.** This bit is set to indicate the the frame was terminated with a closing Flag. This bit is cleared by the Transmit Interrupt Clear command.
1	**0**	**No transmit buffer underrun.**
	1	**Transmit buffer underrun.** A transmit buffer underrun sets this bit and the Transmit Interrupt Pending, but only after the frame transmission is complete (closing Flag or Abort has been sent). This bit is cleared by the Transmit Interrupt Clear command.
0	**0**	**Transmit buffer is full.**
	1	**Transmit buffer is not full.** This bit indicates that there is room available to write data to the transmit buffer.

Serial x Receive Control Register		(SARCR) (Address = 0x04) (SBRCR) (Address = 0x14)
Bit(s)	**Value**	**Description**
7	**0**	**No operation.**
	1	**Abort Receive.** This command empties the receive buffer and forces the receiver to search for a Flag. Purging the receive buffer also clears the RxEOF output.
6	**0**	**No operation.**
	1	**Trigger Receive.** This command causes the receiver to search for a Flag. It also primes the DPLL to synchronize with the next transition on the SER_RXD input. The receive buffer is not purged by this command.
5	**0**	**No operation.**
	1	**Clear Receive Interrupt.** In addition to clearing the Receive Interrupt Pending bit, this command clears bits 6-5 of the Common Status register.
4		This bit is reserved.
3		**PORT[3].** This bit directly controls the state of the PORT[3] output.
2		**PORT[2].** This bit directly controls the state of the PORT[2] output.
1:0	**00**	**Accept All Frames.**
	01	**Accept Only Addressed Frames.** Accept only frames with address byte matching either the Station Address register or Broadcast Address register. If only one address match is required, simply program both address registers with the same value.
	1x	**Accept No Frames.** Accept no receive frames. This option should be selected if there is a possibility that an error frame may purge the receive buffer before the data for the current frame has been removed from the receive buffer.

Serial x Transmit Control Register		(SATCR) (Address = 0x06) (SBTCR) (Address = 0x16)
Bit(s)	**Value**	**Description**
7	**0**	**No operation.**
	1	**Abort Transmit.** This command purges the transmit buffer and forces the transmitter to send an Abort. The current byte in the transmitter will complete, followed by the Abort (0xFE), and the Transmit Interrupt Pending bit will be set once the transmission of the Abort is complete. The transmitter will ignore this command during the transmission of a closing Flag and while the transmitter is idle, although the transmit buffer will still be purged in these cases.
6	**0**	**No operation.**
	1	**Trigger Transmit.** This command may be issued before data is written to the buffer, during the buffer fill, or after the buffer is full. If no data is in the buffer the transmitter will wait for the buffer to contain at least one byte before starting transmission. This command cannot be issued again until after the Transmit Interrupt Pending bit is set to one at the end of a frame.
5	**0**	**No operation.**
	1	**Clear Transmit Interrupt.** In addition to clearing the Transmit Interrupt Pending bit, this command clears bits 3-2 of the Common Status Register.
4		This bit is reserved.
3		**PORT[1].** This bit directly controls the state of the PORT[1] output.
2		**PORT[0].** This bit directly controls the state of the PORT[0] output.
1:0	**00**	**Transmit Buffer Only.** The transmit frame payload is the contents of the buffer, plus the optional CRC.
	01	**Station Address plus Transmit Buffer.** The transmit frame payload is the Station Address register, followed by the contents of the transmit buffer, plus the optional CRC.
	10	**Station Address, Control Byte, plus Transmit Buffer.** The transmit frame payload is the Station Address register, followed by the Control Byte register, followed by the contents of the transmit buffer, plus the optional CRC.
	11	**Station Address, Control Byte Only.** The transmit frame payload is the Station Address register, followed by the Control Byte register, plus the mandatory CRC (four bytes in total). With this option the transmitter will immediately begin frame transmission after the Trigger Transmit command, because no data is required from the transmit buffer.

Serial x Broadcast Address Register		(SABAR)	(Address = 0x08)
		(SBBAR)	(Address = 0x18)
Bit(s)	**Value**	**Description**	
7:0	**Write**	**Broadcast Address.** This register is used for the second address compare in the receiver.	

Serial x Station Address Register		(SASAR)	(Address = 0x0A)
		(SBSAR)	(Address = 0x1A)
Bit(s)	**Value**	**Description**	
7:0	**Write**	**Station Address.** This register is used for the address compare in the receiver. This register is optionally used as the first byte of a transmit frame.	

Serial x Control Byte Register		(SACBR)	(Address = 0x0C)
		(SBCBR)	(Address = 0x1C)
Bit(s)	**Value**	**Description**	
7:0		**Control Byte.** This register is optionally used as the second byte of a transmit frame.	

Serial x Divide Constant Register		(SADCR)	(Address = 0x0E)
		(SBDCR)	(Address = 0x1E)
Bit(s)	**Value**	**Description**	
7:0	**Write**	**Time Constant.** The contents of this register are the time constant for the baud rate generator. The baud rate generator counts modulo 2*(n+1).	

Timing Diagrams

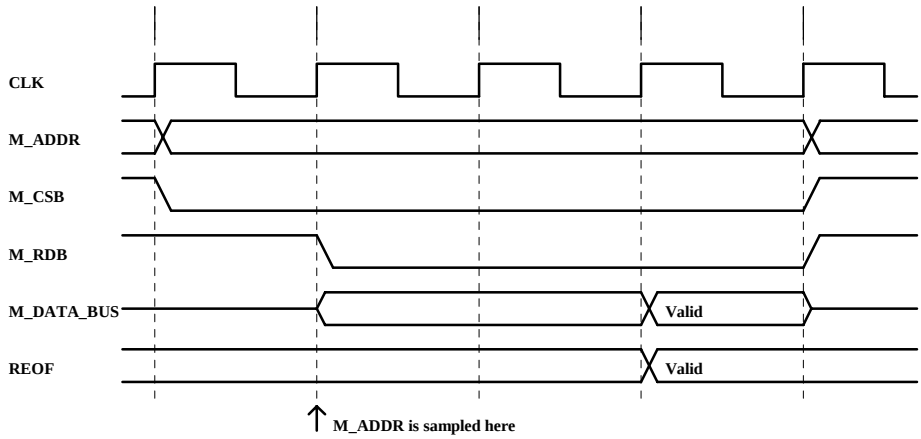
The design uses a synchronous bus interface to simplify the logic, and a read or write transaction requires at least four clock cycles.

Read Transaction

The read transaction is shown in the diagram below. Although the address is shown valid for the duration of the transaction, in reality it is sampled as shown in the diagram, for both read and write transactions.

Note that when reading from a buffer the data output changes to the final value during the transaction. This is due to the way that the memories in the FPGA operate. When reading from a register the data is valid for the duration of the transaction.

Because the REOF signal is a direct output from the buffer, it goes active at the same time that the last data in a frame is read from the buffer

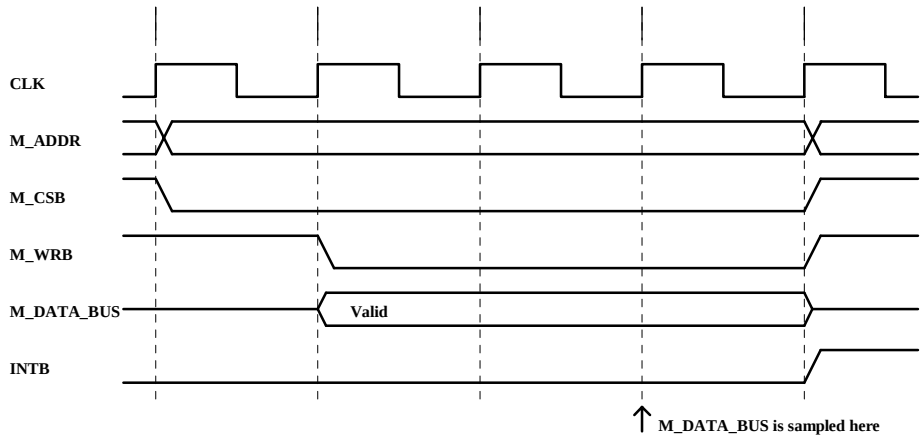


Write Transaction

The write transaction is shown in the diagram below. Although the data is shown valid for the duration of the write strobe, in reality it is sampled as shown in the diagram.

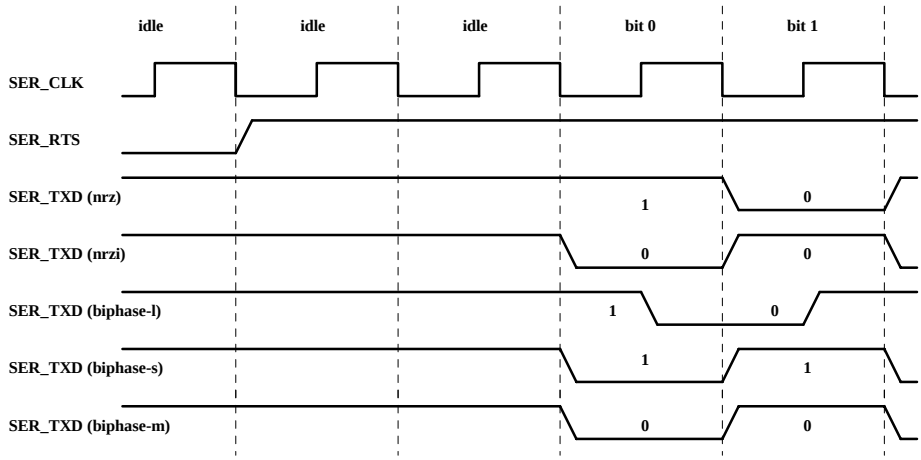
The timing of the de-assertion of the INTB signal timing is shown here for reference. This signal will go inactive in response to a Clear Interrupt command in either the Receive Control register or the Transmit Control register.

The timing of the de-assertion of the REOF signal in response to a Receive Abort command is identical to the timing if the de-assertion of the INTB signal.



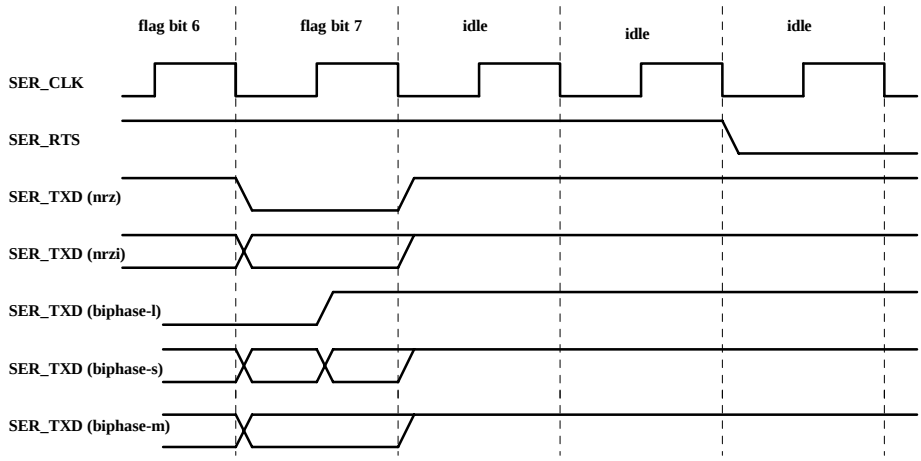
Transmit Frame Start

The timing at the start of a transmit frame is shown in the diagram below. The SER_RTS signal is always asserted exactly two serial clocks prior to the first bit (either preamble or openning Flag) of the frame. The diagram below shows the case of a preamble.



Transmit Frame End

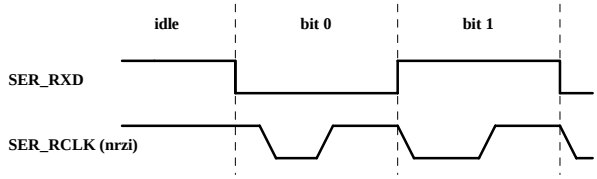
The timing at the end of a transmit frame is shown in the diagram below. The SER_RTS signal is always de-asserted exactly two serial clocks after the last bit of the closing Flag of the frame.



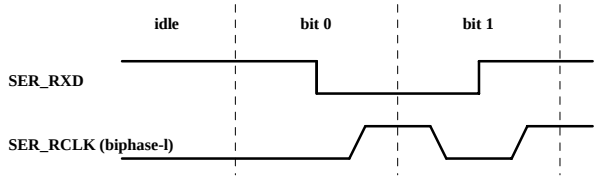
DPLL Receive Clock Start-up

The timing at the start of a receive frame when using the DPLL is shown in the figures below.

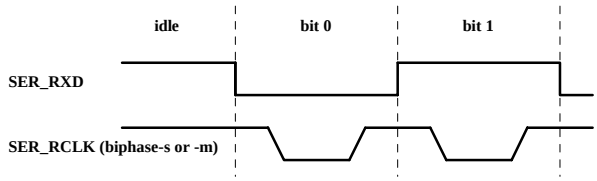
In the case of NRZI encoding, shown below, the initial falling edge of the clock is delayed from the normal timing. This is not a problem because the falling edge of the receive is not used to decode the data.



In the case of Biphase-L encoding, shown below, the receive clock starts cleanly with a rising edge. Only the rising edge of the clock is used to decode Biphase-L, so the first bit will be correctly decoded.



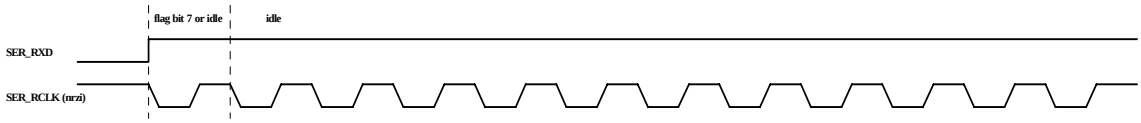
In the case of Biphase-S and Biphase-M encoding, shown below, the receive clock starts cleanly with a falling edge. With Biphase-S and Biphase-M decoding both edges of the receive clock sample the receive data for the decoding, and the first bit will be correctly decoded..



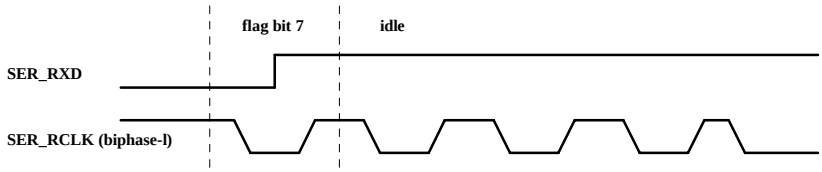
DPLL Receive Clock Shut-down

The timing at the end of a receive frame when using the DPLL is shown in the figures below.

In the case of NRZI encoding, shown below, the DPLL-generated receive clock continues for some time at the end of a frame. This is because it takes eight bit times to recognize the idle line condition.



In the case of Biphase-L encoding, shown below, the receive clock stops three bit times after the last properly-encoded bit is received.



In the case of Biphase-S and Biphase-M encoding, shown below, the receive clock stops two bit times after the last properly-encoded bit is received.