

12 SYNCHRONOUS SERIAL INTERFACE

12.1 Overview

The synchronous serial interface (sync serial) in the ETRAX 100LX sends the transmission clock along with the data. Communication takes place between two parties: a master which generates the clock, and a slave. The ETRAX 100LX can be either the master or the slave.

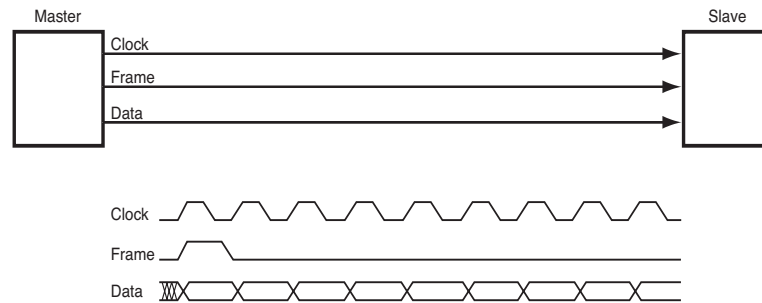


Figure 12-1 Simple Synchronous Serial Port

The sync serial interface has two synchronous serial ports. The first sync serial port is named *Synchronous Serial Port p1* and the other is *Synchronous Serial Port p3*. These two ports are multiplexed with Asynchronous Serial Ports p1 and p3 respectively, so Sync Serial Ports p0 and p2 do not exist.

12.2 Mode Selection

There are six different operation modes into which the sync serial ports of the ETRAX 100LX can be configured. Three different types of communication channels can be used, two of which are unidirectional, and one of which is bidirectional. Because there is one master and one slave in each type of channel, one synchronous serial port may be configured into one of the following six modes:

Mode	Description
Master Output	The ETRAX 100LX generates clock signals, and data is transferred to the external unit.
Master Input	The ETRAX 100LX generates clock signals, and data is transferred from the external unit.
Master Bidirectional	The ETRAX 100LX generates clock signals, and data is transferred in both directions.
Slave Output	The ETRAX 100LX sends data to the external unit depending on the incoming clock signals it receives.
Slave Input	The ETRAX 100LX receives data from the external unit depending on the incoming clock signals it receives.
Slave Bidirectional	The ETRAX 100LX sends and receives data from the external unit depending on the incoming clock signals it receives.

To properly configure the ETRAX 100LX, select the mode that is supported by the other circuit. If two ETRAX 100's are connected to each other, any of the six modes are available.

12 Synchronous Serial Interface

12.3 Pin Usage

12.3.1 Pin Configuration

The sync serial ports are multiplexed onto the same pins as some other interface applications, see chapter *19 Electrical Information*.

To select sync serial port p1, the **sermode1** bit in the R_GEN_CONFIG_II register must be set to **sync** (0x1). To select sync serial p3, the **sermode3** bit in the R_GEN_CONFIG_II register must be set to **sync** (0x1), and the **ser3** bit in the R_GEN_CONFIG register must be set to **select** (0x1).

For every configuration the sync serial interface has two output pins and two input pins, and the pin function is different depending on the mode of operation. To make all configurations complete, however, one extra bidirectional pin per port will be required. In order to use some synchronous serial port modes, 5 pins are needed.

Extra pins must be configured from General I/O Ports **pb4** or **pb7**, which is done with the register R_PORT_PB_SET. For a detailed description of the different registers, please refer to chapter 18.18 *Synchronous Serial Port Registers*.

12.3.2 Pin Usage in the Different Modes

The signal names at the I/O pins differ, depending upon the mode in use. The following tables show the different I/O pin assignments of the two synchronous serial ports in each mode of operation.

Synchronous Serial Ports - Master Output Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3				
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name	Interface Signal Name	Description
T17	rxd1	-	J18	$\overline{s1sel}$	-	ss1_in1 ss3_in1	Not used by the synchronous serial ports in Master Output mode.
V19	$\overline{cts1}$	ss1status	J19	$\overline{s1bsy}$	ss3status	ss1_in2 ss3_in2	Serial busy input to respective port.
U19	txd1	ss1clk	H20	s1sel	ss3clk	ss1_out1 ss3_out1	Serial clock output from respective port.
W20	$\overline{rts1}$	ss1data	B19	$\overline{s1en}$	ss3data	ss1_out2 ss3_out2	Serial data output from respective port.
W17	pb4	ss1frame	U16	pb7	ss3frame	ss1_io3 ss3_io3	Serial frame indicator output from respective port.

Synchronous Serial Ports - Master Input Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3			Interface Signal Name	Description
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name		
T17	rxd1	ss1data	J18	$\overline{s1sel}$	ss3data	ss1_in1 ss3_in1	Serial data input to respective port.
V19	$\overline{cts1}$	ss1status	J19	$\overline{s1bsy}$	ss3status	ss1_in2 ss3_in2	Serial empty input to respective port.
U19	txd1	ss1clk	H20	s1sel	ss3clk	ss1_out1 ss3_out1	Serial clock output from respective port.
W20	$\overline{rts1}$	ss1frame	B19	$\overline{s1en}$	ss3frame	ss1_out2 ss3_out2	Serial frame indicator from respective port.
W17	pb4	-	U16	pb7	-	-	Not used by the synchronous serial ports in Master Input mode.

Synchronous Serial Ports - Slave Output Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3			Interface Signal Name	Description
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name		
T17	rxd1	ss1clk	J18	$\overline{s1sel}$	ss3clk	ss1_in1 ss3_in1	Serial clock input to respective port.
V19	$\overline{cts1}$	ss1frame	J19	$\overline{s1bsy}$	ss3frame	ss1_in2 ss3_in2	Serial frame indicator to respective port.
U19	txd1	ss1data	H20	s1sel	ss3data	ss1_out1 ss3_out1	Serial data output from respective port.
W20	$\overline{rts1}$	ss1status	B19	$\overline{s1en}$	ss3status	ss1_out2 ss3_out2	Serial empty output from respective port.
W17	pb4	-	U16	pb7	-	-	Not used by the synchronous serial ports in Slave Output mode.

12 Synchronous Serial Interface

Synchronous Serial Ports - Slave Input Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3			Interface Signal Name	Description
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name		
T17	rxd1	ss1clk	J18	$\overline{s1sel}$	ss3clk	ss1_in1 ss3_in1	Serial clock input to respective port.
V19	$\overline{cts1}$	ss1frame	J19	$\overline{s1bsy}$	ss3frame	ss1_in2 ss3_in2	Serial frame indicator to respective port.
U19	txd1	-	H20	s1sel	-	ss1_out1 ss3_out1	Not used by the synchronous serial ports in Slave Input mode.
W20	$\overline{rts1}$	ss1status	B19	$\overline{s1en}$	ss3status	ss1_out2 ss3_out2	Serial busy output from respective port.
W17	pb4	ss1data	U16	pb7	ss3data	ss1_io3 ss3_io3	Serial data input to respective port.

Synchronous Serial Ports - Master Bidirectional Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3			Interface Signal Name	Description
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name		
T17	rxd1	ss1status	J18	$\overline{s1sel}$	ss3status	ss1_in1 ss3_in1	Serial busy input to respective port.
V19	$\overline{cts1}$	ss1idata	J19	$\overline{s1bsy}$	ss3idata	ss1_in2 ss3_in2	Serial data input to respective port.
U19	txd1	ss1clk	H20	s1sel	ss3clk	ss1_out1 ss3_out1	Serial clock output from respective port.
W20	$\overline{rts1}$	ss1odata	B19	$\overline{s1en}$	ss3odata	ss1_out2 ss3_out2	Serial data output from respective port.
W17	pb4	ss1frame	U16	pb7	ss3frame	ss1_io3 ss3_io3	Serial frame indicator output from respective port.

Synchronous Serial Ports - Slave Bidirectional Mode							
Synchronous Serial Port p1			Synchronous Serial Port p3			Interface Signal Name	Description
Solder Ball	Chip Pin Name	Mode Signal Name	Solder Ball	Chip Pin Name	Mode Signal Name		
T17	rxd1	ss1clk	J18	$\overline{s1sel}$	ss3clk	ss1_in1 ss3_in1	Serial clock input to respective port.
V19	$\overline{cts1}$	ss1frame	J19	$\overline{s1bsy}$	ss3frame	ss1_in2 ss3_in2	Serial frame indicator to respective port.
U19	txd1	ss1status	H20	s1sel	ss3status	ss1_out1 ss3_out1	Serial busy output from respective port.
W20	$\overline{rts1}$	ss1odata	B19	$\overline{s1en}$	ss3odata	ss1_out2 ss3_out2	Serial data output from respective port.
W17	pb4	ss1idata	U16	pb7	ss3idata	ss1_io3 ss3_io3	Serial data input to respective port.

12.4 Synchronous Serial Port Registers

The sync serial ports are served by two sets of dedicated registers, one set for each port. The table below summarizes the purpose of each register.

Registers	Function
R_SYNC_SERIAL_PRESCALE	A 32-bit write-only register that selects clock source and frame sync rate for both ports.
R_SYNC_SERIAL1_REC_DATA R_SYNC_SERIAL3_REC_DATA	32-bit read-only registers containing data in from the serial receivers.
R_SYNC_SERIAL1_REC_WORD R_SYNC_SERIAL3_REC_WORD	16-bit read-only registers containing data in from the serial receivers.
R_SYNC_SERIAL1_REC_BYTE R_SYNC_SERIAL3_REC_BYTE	8-bit read-only registers containing data in from the serial receivers.
R_SYNC_SERIAL1_STATUS R_SYNC_SERIAL3_STATUS	32-bit read-only registers containing status information from the ports.
R_SYNC_SERIAL1_TR_DATA R_SYNC_SERIAL3_TR_DATA	32-bit write-only registers containing data out to the serial receivers.
R_SYNC_SERIAL1_TR_WORD R_SYNC_SERIAL3_TR_WORD	16-bit write-only registers containing data out to the serial receivers.
R_SYNC_SERIAL1_TR_BYTE R_SYNC_SERIAL3_TR_BYTE	8-bit write-only registers containing data out to the serial receivers.
R_SYNC_SERIAL1_CTRL R_SYNC_SERIAL3_CTRL	32-bit write-only registers for configuration and control of each port.

For more detailed information about sync serial registers, refer to chapter 18.18 *Synchronous Serial Port Registers*.

12.5 Configuration

Besides the different modes described in section 12.2 *Mode Selection*, a number of things can be configured in the R_SYNC_SERIAL x _CTRL register:

- Word length can be either 8, 12, 16, 24 or 32 bits.
- A number of different frame synchronization modes can be configured.
- The ports can be controlled by the CPU or by DMA.
- Data can be sent with either the lsb or msb first.
- The active clock edge may be either positive or negative.
- All control signals can be individually inverted.
- The receiver and transmitter can be enabled and disabled individually as well.

12.6 Word Length

The sync serial interface supports a transmitted/received word length of either 8, 12, 16, 24 or 32 bits. The word length is configured with the **wordsize** field in R_SYNC_SERIAL x _CTRL.

Note 1: To avoid unnecessary repetition, the x in R_SYNC_SERIAL x _CTRL above and elsewhere in this chapter, stands for 1_ and 3_, representing Synchronous Serial Ports p1 and p3.

When the sync serial port is controlled by the CPU, transmit data is written to R_SYNC_SERIAL x _TR_DATA, and receive data is read from R_SYNC_SERIAL x _REC_DATA. Each read or write corresponds to one received or transmitted word of the selected word length.

The **wordsize** lsb bits are used by the interface. For word lengths other than 32-bits, the upper bits of R_SYNC_SERIAL x _TR_DATA are ignored by the transmitter, and the upper bits of R_SYNC_SERIAL x _REC_DATA contain invalid data.

The lower 8 bits or 16 bits of the transmit data can be written in the registers R_SYNC_SERIAL x _TR_BYTE and R_SYNC_SERIAL x _TR_WORD respectively. Each write to one of these registers will result in the transmission of one word of the selected word length. If the word length is wider than the written register, the remaining bits will be the same as in the previously transmitted word.

There are 8-bit and 16-bit registers for the data read as well, R_SYNC_SERIAL x _REC_BYTE and R_SYNC_SERIAL x _REC_WORD respectively.

When the sync serial interface is DMA driven, each transmitted/received word is represented by 1, 2 or 4 bytes in memory, depending on the selected word length:

Selected Word Length (bits)	Size In Memory (bytes)
8	1
12	2
16	2

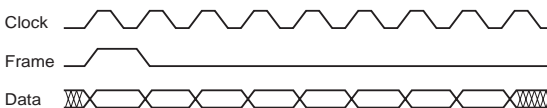
Selected Word Length (bits)	Size In Memory (bytes)
24	4
32	4

12.7 Frame Synchronization

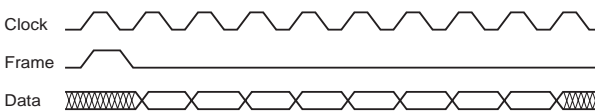
12.7.1 Frame Synchronization Modes

The following figures show the different types of frame synchronization configurations in R_SYNC_SERIALx_CTRL:

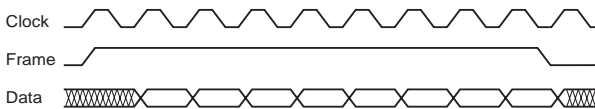
Bit



Early, Bit



Early, Extended



Word



Early, Word

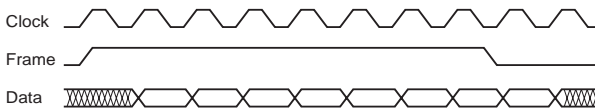


Figure 12-2 Synchronous Serial Type Configurations

Note 1: When the `f_syncsize` field in R_SYNC_SERIALx_CTRL is set to *extended*, the field `f_synctype` must be set to *early*.

12.7.2 Frame Strobe Generation

When the sync serial interface operates in Master mode, the frame strobe output interval is controlled by the **word_rate** and **frame_rate** fields in R_SYNC_SERIAL_PRESCALE. The selected values are common to both ports.

The **word_rate** field selects the interval, in number of bit clocks, between the start of consecutive words transmitted or received. The word rate interval length is the value in the **word_rate** field + 1.

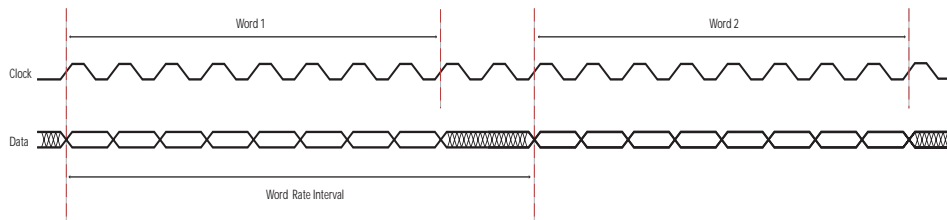


Figure 12-3 Word Rate Interval

The maximum value for the word rate interval is 1024 bits (i.e. **word_rate** field = 1023). The minimum value is restricted by the word length, see table 12-1 below:

Frame Sync Type: Bit or Word			Frame sync type: Early Bit, Early Word or Early Extended		
Word Length	Minimum word_rate Value	Minimum Word Rate Interval	Word Length	Minimum word_rate Value	Minimum Word Rate Interval
8	7	8	8	8	9
12	11	12	12	12	13
16	15	16	16	16	17
24	23	24	24	24	25
32	31	32	32	32	33

Table 12-1 Word length

The **frame_rate** field selects the number of words received or transmitted between each frame strobe output. The frame strobe interval is the value in the **frame_rate** field + 1. If the **frame_rate** field is set to 0, a frame strobe will be output for each word received or transmitted. This is the appropriate setting for most applications.

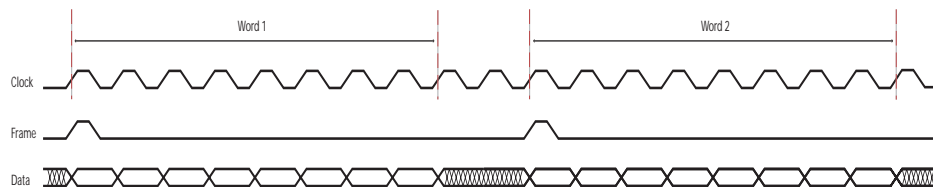


Figure 12-4 Frame Strobe Output with **frame_rate** Set to 0.

12.7.3 Stream Mode

It is possible to turn off frame synchronization, so that no frame strobe output is generated and the incoming frame strobe is ignored. Words will be transmitted or

received back-to-back regardless of the word rate setting. This mode is used if the transferred data contains embedded synchronization.

12.8 Clocking

12.8.1 Clock Generator

There are three possible clock sources for the sync serial interface:

- Internally generated codec clock.
- Externally generated codec clock.
- Baudrate clock from the async serial port.

Clock selection is made in register R_SYNC_SERIAL_PRESCALE. Fields `clk_sel_u3` and `clk_sel_u1` are used to select either the codec clock or baudrate clock. The source for the codec clock (internal or external) is defined when *operation mode* is selected.

Internally Generated Codec Clock

The base clock is 4.096 MHz.

For selectable clock division, the base clock is then divided using a prescaler with a division factor of 1 to 128 before it is used by the sync serial port. The division factor is set in the `prescaler` field. The selected value is common to both ports.

Baudrate Clock from Async Serial Ports

When this mode is used, the clock is generated from the baud rate clock generator which is used in the async serial ports. This clock generator may be configured in a number of different ways, which is selected in the R_ALT_SER_BAUDRATE register and in the `tr_baud` field of R_SYNC_SERIAL_X_CTRL. For a description of the different possibilities, refer to chapter 11.5 *Baud Rate Selection*.

Before this clock is used, the frequency is divided by two. Thus, the selectable frequency settings in the `tr_baud` field of the R_SYNC_SERIAL_X_CTRL register, are half of the selected frequency for the asynchronous serial ports.

For the baudrate selection modes that do not use the `tr_baud` field, the following formula applies:

$$\text{sync serial bit clock} = \frac{\text{async serial baud rate}}{2}$$

12.8.2 Data Sampling

Sampling of incoming signals is controlled by the `clk_polarity` field in R_SYNC_SERIAL_X_CTRL. The outgoing clock may be inverted with the `clk_driver` field.

12 Synchronous Serial Interface

When clocks are not inverted, output signals are changed on the rising edge, and input signals are sampled on the falling edge of the clock, see figure 12-5 below.

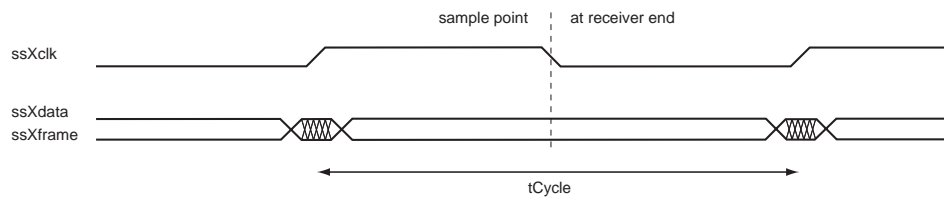


Figure 12-5 Master Output/Slave Input

12.8.3 Clock Gating

When clock gating is turned on (**clk_mode** is set to **gated** (1)), every negative clock edge is a bit strobe. When nothing is sent, the clock is held low. However, if clock gating is turned off (**clk_mode** is set to **normal** (0)), the clock will run continuously. Figure 12-6 below shows an example of the input mode with clock gating on:

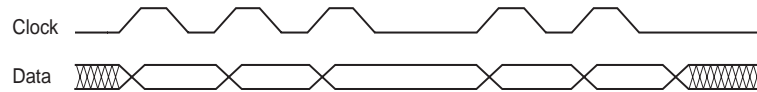


Figure 12-6 Input Mode with Clock Gating On

Note that clock gating does not work correctly with **flow_ctrl** set to **enable** (1) because the transmitter will stop clocking and never sample the status signal after its initial rise.

If more than one word is transferred between frames, as in figure 12-7 below, the clock must be gated for the receiver to understand the incoming data because the word strobe signal is not sent (only the frame strobe).

With clock gating turned on, the clock will only be active when the information is sent. In this case, every active clock edge contains information, and as a result, many words per frame can be understood by the receiver.

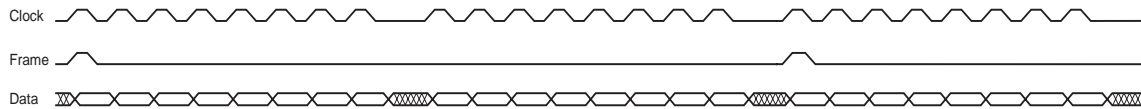


Figure 12-7 Output Mode with Clock Gating on, 2 Words Per Frame

The number of words per frame is selected in **R_SYNC_SERIAL_x_PRESCALE**.

12.9 Flow Control

In the `R_SYNC_SERIALx_CTRL` register, if the `flow_ctrl` field is set to **off** (0), the amount of transferred data is defined by the selected word rate. Data must be available when they are to be sent, and storage must be available before more data is received. If this is not the case, the failing unit will go into an error state which may be detected by reading the status register: `R_SYNC_SERIALx_STATUS`.

When `flow_ctrl` is set to **on** (1), one or more status signals between units are used to pause the transfer until the receiver is prepared. This will delay word and frame synchronization. The master unit is paused by the slave if the slave can not deliver or receive data. If the master runs out of buffers, the start frame/word is delayed internally until the congestion is cleared. An overrun/underflow error will be reported if the transmitter or receiver over/under run their FIFO's.

When the `error` field in `R_SYNC_SERIALx_CTRL` is set to **ignore** (1), the transmission is continued after next frame sync even if an error is detected. When `error` is set to **normal** (0), the transfer is halted when an error is detected. The status signal should be seen as the way the slave halts the transfer if data/storage is not available, and the frame signals the way the master controls the same variables.

The frame pulse, represented by a dotted line, is skipped.

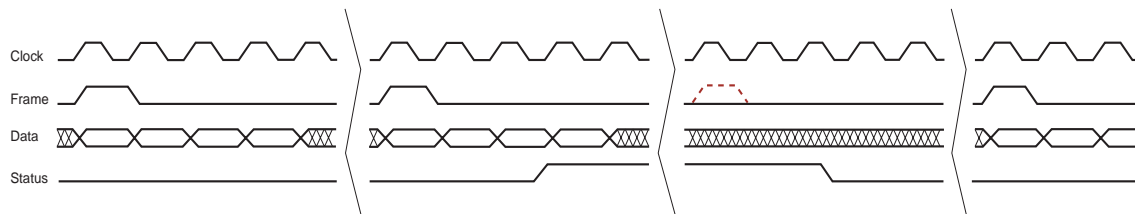


Figure 12-8 Flow Control

12.10 Interrupts

There are two interrupts for each sync serial port. The same interrupt bits in the `R_IRQ_MASK1_RD` register that are used for the asynchronous serial ports are also used for the sync serial interface: fields `ser1_ready`, `ser1_data`, `ser3_ready`, and `ser3_data`. For detailed information see chapter 18 *Interrupts*.

The `ser1_ready` or `ser3_ready` interrupt is generated when the sync serial port is ready to accept a new data word to be transmitted, i.e. whenever the `tr_ready` field of the `R_SYNC_SERIALx_STATUS` register is set. The interrupt is cleared when new data is written to one of the port data transmit registers

`R_SYNC_SERIALx_TR_DATA`, `R_SYNC_SERIALx_TR_WORD` or `R_SYNC_SERIALx_TR_BYTE`.

The `ser1_data` or `ser3_data` interrupt is generated when the sync serial port has received a new data word, i.e. whenever the `data_avail` field of the `R_SYNC_SERIALx_STATUS` register is set. The interrupt is cleared when the data is read from one of the port data receive registers `R_SYNC_SERIALx_REC_DATA`, `R_SYNC_SERIALx_REC_WORD` or `R_SYNC_SERIALx_REC_BYTE`.

12.11 Using The Sync Serial Ports with DMA

The sync serial ports can be driven by the CPU or by DMA. Sync serial port 1 uses internal DMA channel 8 for output, and DMA channel 9 for input. Port 3 uses DMA channel 6 for output and DMA channel 7 for input.

When DMA is used, the **dma_enable** field in R_SYNC_SERIAL1_CTRL register must be set, and the ports must be connected to their respective DMA channels, which is done in the R_GEN_CONFIG register.

For input transfers with DMA, the incoming data will be stored in the fifo of the DMA channel, and will not be written out to memory until the fifo is filled with 16 or 32 bytes (depending on the DMA burst length set in the R_BUS_CONFIG register). The DMA can be forced to write out the fifo contents to memory, by setting the appropriate bits in the R_SET_EOP register. The DMA will then set an **eop** in the current DMA descriptor, and advance to the next descriptor.

For further information on DMA operation, see chapter 7 *DMA*.