Sundance Multiprocessor Technology Limited **Design Specification**	Form : QCF51 Dated : 15 March 2001 Revision : 5.0 Approved :

Unit / Module Name:	
Unit / Module Number:	Sundance Data Link Interface
Used On:	
Document Issue:	
Date:	

CONFIDENTIAL

Outline Description

Sundance's TIMs offer up to six Sundance Digital Links (SDL) for short distance interfacing with other TIMs.

In a software point of view the SDL looks the same as a ComPort interface but transfers data faster.

In a hardware point of view, SDL was developed by Sundance to overcome the speed and asynchronous transfer issues linked to a ComPort interface. SDL is compatible with TI's ComPort standard.

The SDL can work in two modes:

- Fast mode: SDL protocol is used (not ComPort compatible). It provides a data rate up to 25MB/sec when clocked at 100MHz.
- Slow mode: ComPort compatible mode. It provides a data rate up to 10MB/sec when clocked at 100MHz.

SDL speed will increase to twice those values in the next version of the interface by using Double Data Rate (DDR) technology.

The SDL uses the ComPort links, which provide enough lines for the data and control signals. The 'C4x Protocol from Texas Instrument (ComPort protocol) defines Byte-wide links which can theoretically transmit at 20Mbytes/second asynchronously between TIMs.

A Sundance Data Link is configured as an 8-bit wide data bus, a strobe and 3 control lines to manage the interface. These control lines include Token Request and Acknowledge lines, a pair of lines to indicate a bus exchange, and a Busy signal to indicate when a transfer can be started.

The SDL is designed for short distance connections between TIMs.

Approvals		Date
Managing Director		
Software Manager		
Design Engineer		

Certificate Number FM 55022

Revision History

Date	Changes Made	Issue	Initials
10.04.01	**Initial version**	**1.0**	**E.P**
20.04.01	**Addition of a new chapter:SDB-SDL-ComPort**	**1.1**	**E.P**
05.05.01	**Overall re-writing to make a SDL look exactly like a Comm_port Interface on a software point of view.**	**1.2**	**E.P**
08.05.01	**Correction of Cut & Paste errors and modification of figure 3.**	**1.3**	**E.P**
10.05.01	**Correction of paragraph 2.2 about the use of the bi-directional pins**	**1.4**	**E.P**
01.07.03	**New specifications**	**1.5**	**JPA**
15/09/03	**Updated "outline desccription".** **Added vhdl files hierarchy .**	**1.6**	**JPA**
19/09/03	**Added SDL pinout.**	**1.7**	**JPA**
10/10/03	**Updated documentation**	**1.8**	**JPA**
10/02/04	**Corrected SDL pinout (3.2 section)** **Updated 2.1 section**	**1.9**	**JPA**

Table of Contents

Table of Figures

Table of Tables

1. Overview

The standard gives a TIM six Sundance Data links numbered from 0 to 5. Each link can be a transmitter or a receiver, and will switch automatically between these states depending on the way you use it. Writing to a receiver will cause a hardware negotiation (using NREQ and NACK signals as described later in this specification) that will reverse the state of both ends of the link.

1.1 SDL Initialisation

Following a processor reset, links initialise as transmitters or receivers. When you wire TIMs together you *must* make sure that you only ever connect links initialising as transmitters to links initialising as receivers; never connect two transmitters or two receivers.

For example, connecting link 0 of one TIM to link 4 of another is safe;

Always connect a transmitter at reset to a receiver at reset

On the SMT320/310Q carrier board the physical connection between SDL is made with FMS cables (Ref. SMT3xx-FMS). You must be careful when connecting the cables the make sure that one end is inserted in the opposite sense to the other. One end must have the *blue* backing facing out and the other must have the *silver* backing facing out.

1.2 SDL blocks

Figure 1 shows the internal architecture of a single SDL:

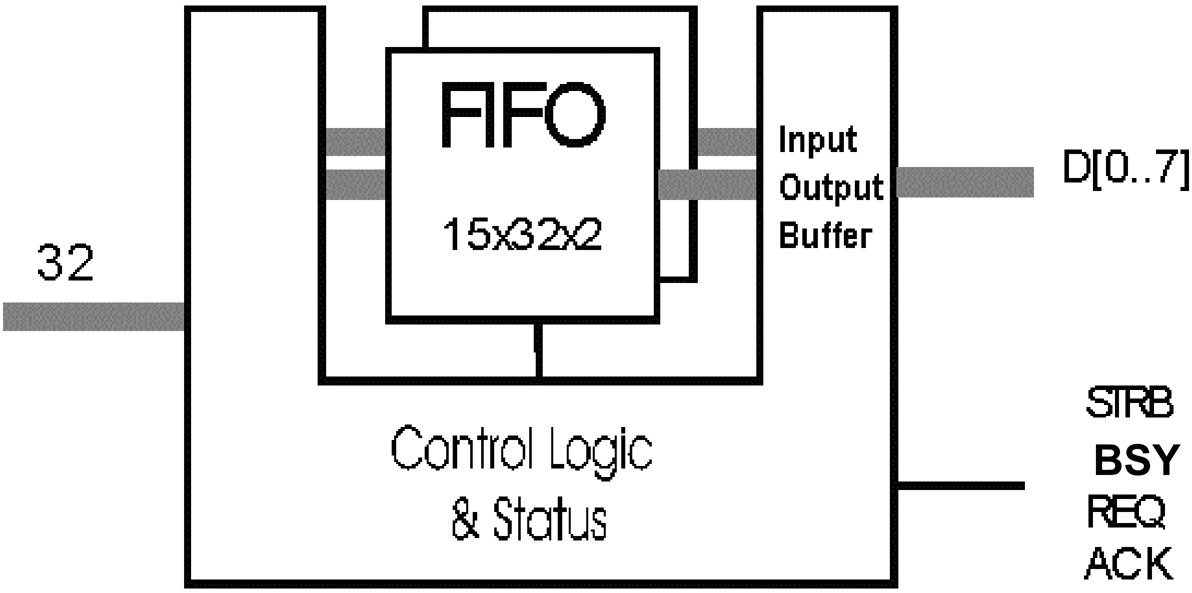

Figure 1: SDL Block Diagram

1.2.1 Input, Output FIFO and Buffers

A SDL is associated with two 15x32-bit unidirectional FIFO; one for input and one for output. An additional one-word buffer makes them appear as 16x32-bit FIFO. These allow the guaranteed transfer rate of 25MB/s to be achieved.

The standard FIFO size is 15x32 but it is not a fixed size. Depending on the requirements in your application, you or Sundance, can replace them by smaller or deeper ones.

1.2.2 Control Logic and Status

The control logic handles the Read, Write and arbitration operations.

Various Flags provide you with information on the current SDL Status.

1.2.3 Fast/Slow mode switching

The SDL can work in two modes:

- Fast mode: SDL protocol is used (not compatible with ComPort). It provides a data rate up to 25MB/sec.
- Slow mode: ComPort protocol is used. It provides a data rate up to 10MB/sec.

It is possible to switch between fast and slow mode “on the fly”. The interface will finish the current transfer, then switch to the next mode and send words available in its FIFO.

The source code provides a signal called “fast_mode“ to switch between fast and slow mode. See Table 4: SDL

2 SDL protocol

2.1 Data formatting

32-bit words are sent in four Byte packets with the least significant byte being sent first.

Each byte is transmitted on the falling edge of NSTRB.

On the transmission side the data is set on the bus at the rising edge of the strobe if the NBUSY line is high and so on the reception side a valid data can be sampled on the falling edge of the strobe. When the first word is received, if its FIFO is full, the receiver sets the NBUSY line to 0, and only releases it to 1 once it can receive another one (ie. At least one word has been read from the FIFO and current transfer is complete).

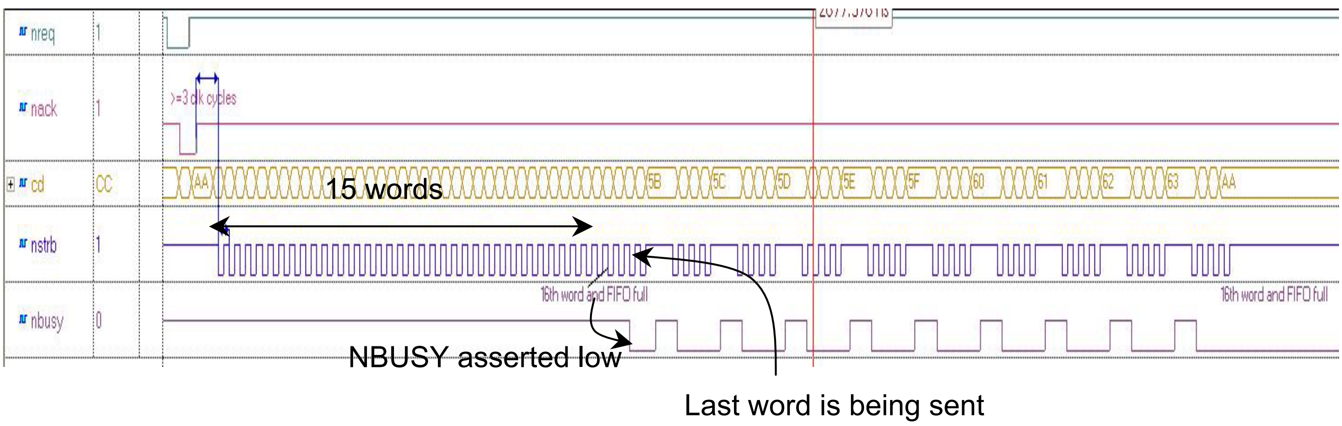

Figure 2: Data transmission timing diagram.

The timing diagram shows the transfer of sixteen 32-bit words. Then, the receiver asserts NBUSY low as long as its FIFO hasn’t been read (and as long as last word transfer isn’t completed). Then, every time a word is read from its FIFO (not shown on the diagram), NBUSY goes high and new word is transmitted. At the end, NBUSY stays low because the receiver’s FIFO isn’t read anymore.

The 32-bits transfer must be completed for NBUSY to go high again. If the transfer is stopped in the middle of the last word, NBUSY will stay low until the remaining bytes are received and FIFO is read.

2.2 Bus exchange

Two signals are used for the bus exchange.

The two bus-arbiters (one in each SDL interface) use NREQ and NACK signals to define which one transmits and which one receives.

The signals NREQ and NACK handle the handshaking arbitration between the two SDL interface:

- **Transmission**: The transmitter drives the strobe, the data signals to transmit and the NACK signal to reply to the bus request. The receiver drives the NREQ signal to request a bus exchange and the NBUSY to interrupt the transmission.
- **Request**: The receiver can, at any time, request the bus by activating NREQ. The transmitter will give the bus immediately after it has finished to transfer the current word.
- **Exchange**: The transmitter acknowledges the request made by activating NACK and frees the bus. The receiver becomes the transmitter and can then use the bus. At least one word will be transmitted before another bus exchange could happen.
- **Interruption**: At any time the receiver can interrupt the transmission without losing data if its reception FIFO gets full. The transmission will go on when the receiver's FIFO is NOT FULL again

The input and output busses are independent so the transmission is possible while emptying the FIFO containing the data previously received.

2.2.1 Bus exchange protocol

Figure 3 shows SDL bus exchange protocol.

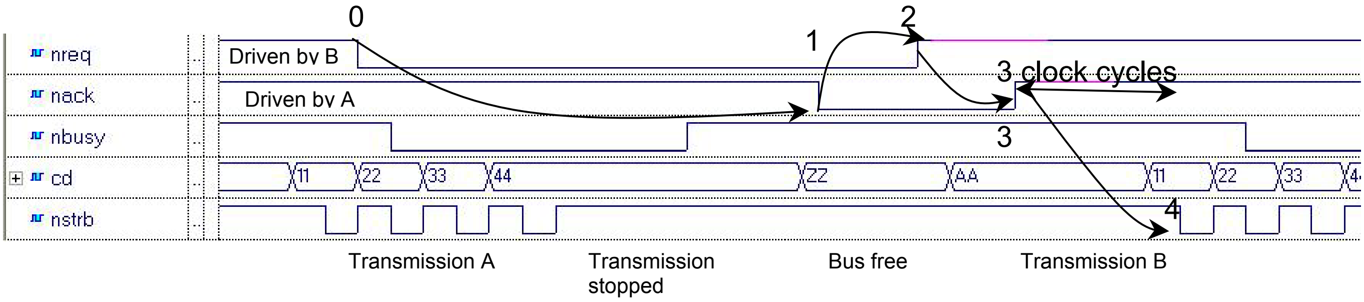

Figure 3: Bus exchange between interfaceA and interfaceB

0: B requests the bus by setting NREQ low.

1: A replies by ending its transmission and setting NACK low.

2: B replies setting NREQ high.

3: A sets NACK high once NREQ is high and its transmission is finished. A becomes a receiver (frees the bus).

4: B drives the bus signals.

If the exchange is allowed and occurs in the middle of a transmission, the current transmission finishes and no extra word will be transferred before the bus ownership is switched.

3 SDL Connector and Pin-out

3.1 SDL connector

Note that when connecting SDLs with the FMS cables, to ensure pin 1 is connected to pin 1 on the alternative SDL, one end of the cable must be inserted opposite to the other, I.E. on one the blue backing must be facing out and on the other the silver of the connectors must be facing outwards.

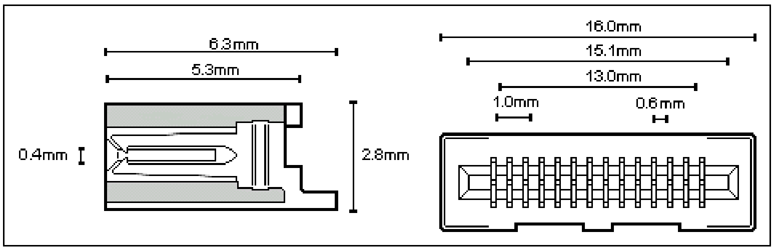

Figure 4: SDL connector

3.2 Pin-out

Pin N°.	Signal	Pin N°.	Signal
1	GND	2	DATA0
3	DATA1	4	DATA2
5	DATA3	6	DATA4
7	DATA5	8	DATA6
9	DATA7	10	NREQ
11	NACK	12	NSTRB
13	NBUSY	14	GND

Table 1: SDL connector pin-out

4 Electrical considerations

Input and Output levels

Input /Output Standard	VIL		VIH		VOL	VOH	IOL	IOH
	V, min	**V, max**	V, min	V, max	V, max	V, min	mA	mA
LVTTL	-0.5	0.8	2.0	5.5	0.4	2.4	24	-24

Table 2: Voltage specifications

VIL and VIH are recommended input voltages.

VOL and VOH are guaranteed output voltages.

IOL and IOH are guaranteed output current.

5 SDL/ComPort compatibility

5.1 Software compatibility

From DSP point of view, SDL and ComPort appears as 32-bits interfaces. Software written using the ComPorts does not need any alteration when the SDL interface is implemented in the on-board FPGA instead of the ComPort interface.

5.2 Hardware compatibility

ComPort and SDL don't need any hardware adaptation to be compatible.

6 Performances

6.1 Data rate

The table below presents the maximum data rate (in MB/sec) achieved for unidirectional transfers. TIM used is SMT365, motherboard used is SMT310Q.

	SDL Vs SDL fast mode	SDL Vs SDL slow mode	SDL Vs CP
FMS cable (20cm)	25	10	7.0
CP switch	25	11	7.7

Table 3: SDL Data rates

SDL Vs SDL: two SDL connected together.

SDL Vs CP: one SDL (in slow mode) connected to a ComPort.

6.2 FPGA integration considerations

Follow the post-synthesis statistics of the SDL design given by synthesiser Xilinx XST when targeting a Xilinx VirtexII1000:

Number of Slices:461 out of 5120 9%

Number of Slice Flip Flops: 353 out of 10240 3%

Number of 4 input LUTs: 590 out of 10240 5%

Design Statistics

Macro Statistics :

Registers : 288

1-bit register: 275

32-bit register : 2

4-bit register: 6

6-bit register: 1

8-bit register: 4

Adders/Subtractors : 8

4-bit adder : 7

6-bit adder : 1

Comparators : 4

6-bit comparator equal : 2

6-bit comparator not equal : 2

7 The source code

The interface has been developed using graphical entry tool Active-HDL in VHDL and is fully compatible with Xilinx XST synthesiser and Xilinx Modular Design methodology.

The source code can be provided upon signing a Non-Disclosure Agreement.

7.1 Interface pinout

Table 4 shows SDL interface pinout.

Signal name	**I/O**	**Size (bits)**	**Description**
clk100	I	1	Interface clock.
dsp_read	I	1	Input FIFO read signal.
dsp_write	I	1	Output FIFO write signal.
fast_mode	I	1	Fast/slow mode selection signal. 0 = slow mode 1 = fast mode
fifo_in_reset	I	1	Input FIFO reset signal. 1 = FIFO reset
fifo_out_reset	I	1	Output FIFO reset signal. 1 = FIFO reset
nack_in	I	1	Input for NACK signal
nbusy_in	I	1	Input for NBUSY/NRDY signal
nreq_in	I	1	Input for NREQ signal
nstrb_in	I	1	Input for NSTRB signal
reset	I	1	Reset control input. A logical 1 will reset the interface.
reset_state	I	1	SDL initial state control signal. 0 = transmitter at reset 1 = receiver at reset
cd_in	I	8	Input for SDL data bus.
data_in	I	32	Output FIFO data bus.
high_width	I	2	NSTRB pulse high width. 00 = 2 clk100 cycle. 01 = 3 clk100 cycle. 10 = 4 clk100 cycle. 11 = 5 clk100 cycle.
low_width	I	2	NSTRB pulse low width. 00 = 2 clk100 cycle.

			01 = 3 clk100 cycle. 10 = 4 clk100 cycle. 11 = 5 clk100 cycle.
data_t	O	1	SDL bus tri-state control. 1 = bus tri-state
nack_out	O	1	Output for NACK signal
nack_t	O	1	NACK signal tri-state control. 1 = bus tri-state
nbusy_out	O	1	Output for NBUSY/NRDY signal
nbusy_t	O	1	NBUSY/NRDY signal tri-state control. 1 = bus tri-state
nreq_out	O	1	Output for NREQ signal
nreq_t	O	1	NREQ signal tri-state control. 1 = bus tri-state
nstrb_out	O	1	Output for NSTRB signal
nstrb_t	O	1	NSTRB signal tri-state control. 1 = bus tri-state
cd_out	O	8	Output for SDL data bus.
data_out	O	32	input FIFO data bus.
status	O	32	SDL status see (See http://www.sundance.com/docs/Firmware.pdf)

Table 4: SDL interface pinout

7.2 File hierarchy

SDL interface is composed of the following files:

- SDL_top_if
    - Inputfifo15x32
    - Outputfifo15x32
    - SDL.vhd
        - Transmitter_i.vhd
            - Pulse_generation.vhd
            - Fsm_control.vhd
        - Receiver_i.vhd
            - Fsm_data.vhd
            - Fsm_fifo_flag.vhd
            - Fsm_write_buf.vhd
            - Fsm_nbusy.vhd

The following pictures present an overview of the interconnexion of different blocks composing the interface.

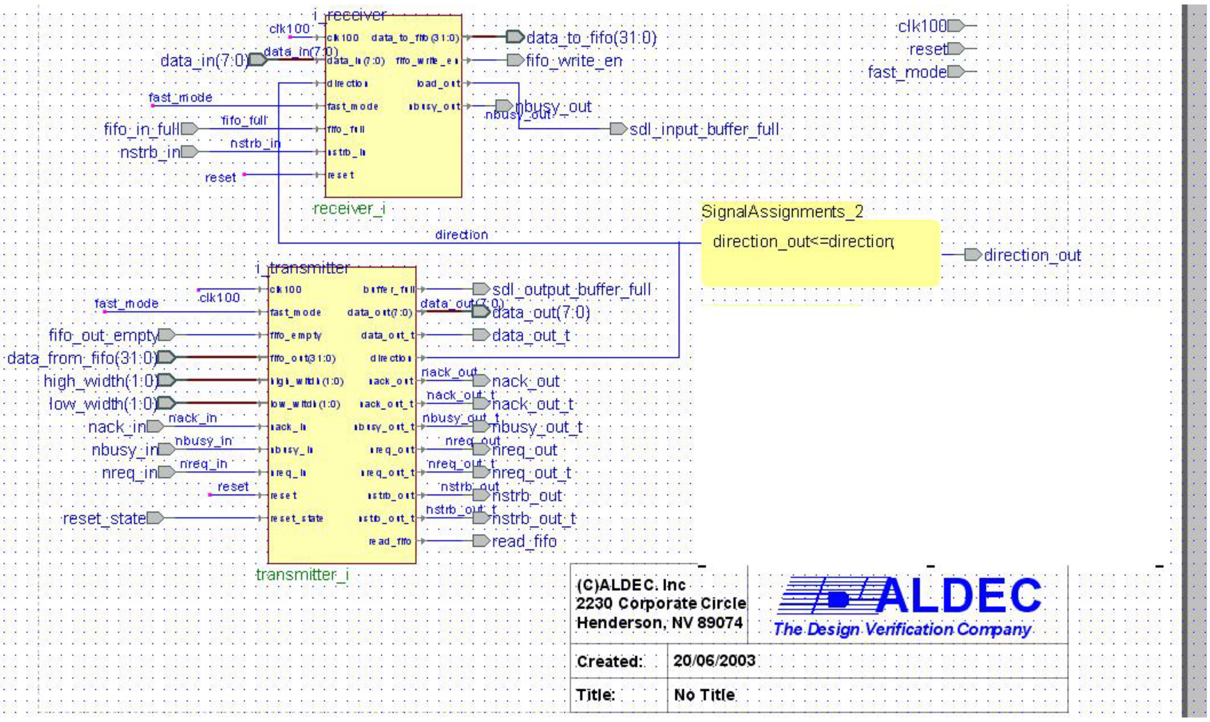

Figure 5: SDL.vhd

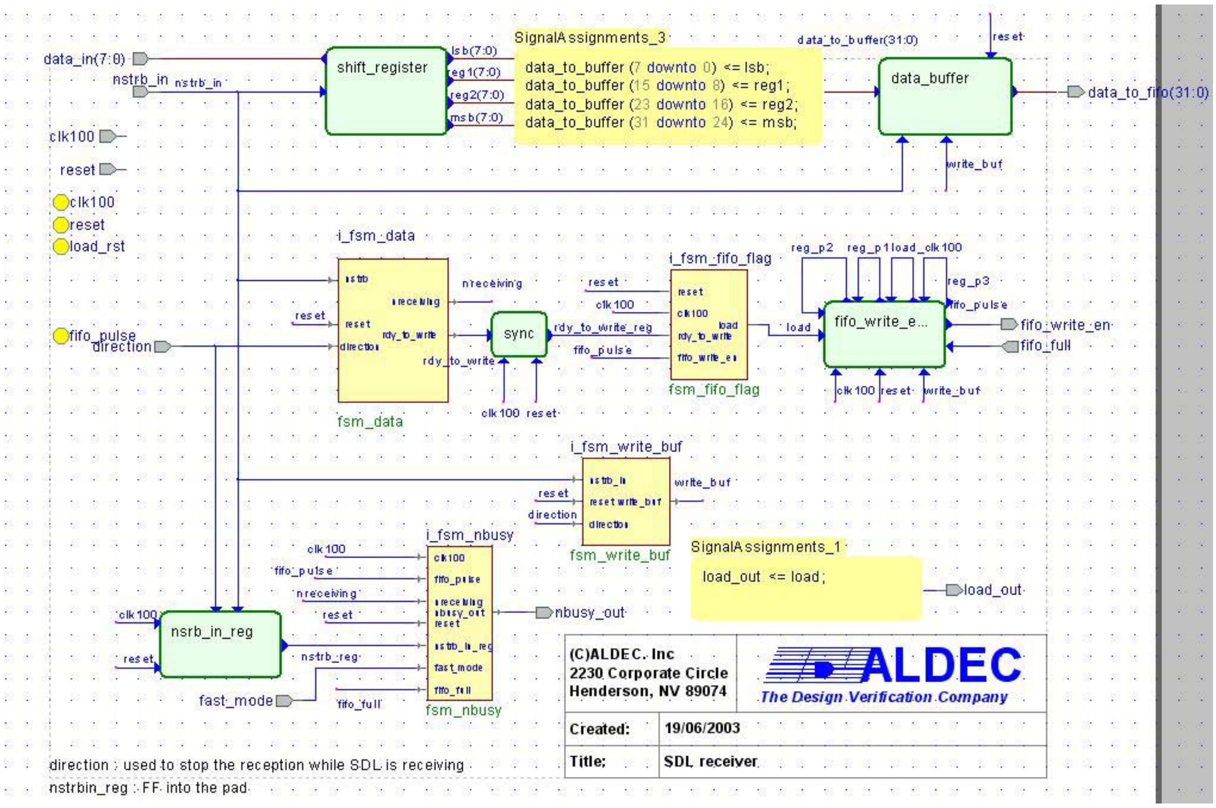

Figure 6: receiver_i.vhd

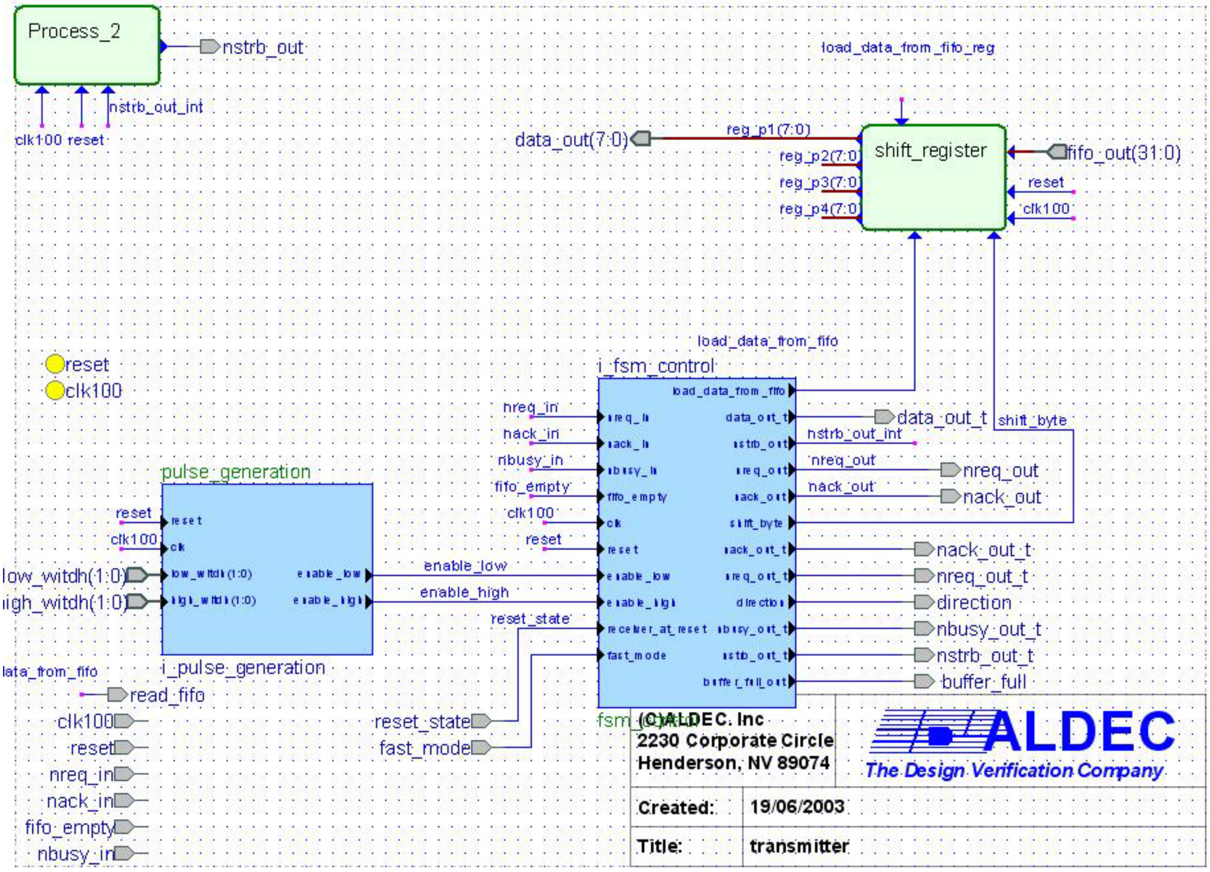

Figure 7: transmitter_i.vhd