# *SSD8IO-MF Configuration Package*
## *(Synchronous Serial Device – 8 I/O Multi-Function)*

The SSD8IO-MF is a configuration package that enables the PCI CDa board to transfer eight channels of synchronous serial input/output between an external device and a PCI bus host computer.

Additional handshaking and clocking signals are available on channels 0–3, while channels 4–7 provide functionality that is identical to that provided by standard ssd16io.bit configuration files (see the SSD16IO Addendum).

Channels 0 and 1 are output channels; each channel has a data enable output signal and a device busy input signal in addition to the normal clock and data signals.

The output data enable signals are programmable via configuration registers to provide blocks of 32 * M bits of data valid true separated by N bits of data valid false where M and N are values set in the configuration registers. (See Enable Burst Delay Register and Enable Burst Size Register.) Output clock signals can be sourced from either an internal PLL or an external clock input via the connector pinout.

Channels 2 and 3 are input channels; each channel has a data enable input signal in addition to the normal clock and data signals. Data is latched only when the data valid input signal is true.

Channels 4 through 7 can be configured as either input or output channels in pairs. No additional handshaking signals are provided; however, the output clock can be sourced independently from channels 0 and 1.

The SSD8IO-MF package was developed using the Xilinx Project Navigator. The VHDL source is available and the project is set up so that you can use it if you wish to do so. Contact EDT for details.

### Related Manuals

Detailed documentation on EDT's C software library routines, helpful for writing your applications, is available on EDT's website.

# Installation

In the instructions below, substitute appropriate values for the placeholders in *italics*.

To install the SSD8IO_MF package:

1. Install the Pcd driver software by following the instructions on the EDT installation disk sleeve or on the EDT website (www.edt.com).
2. Install the board assembly in the host computer as specified by the computer manufacturer.
3. To configure the board, at the command prompt, enter:

   ```
   initpcd -u unit number -f configuration file
   ```

   For example, to configure board 0 with the sample configuration file provided, enter:

   ```
   initpcd -u 0 -f pcd_config/ssd8io_mf.cfg
   ```

## About the Software and Firmware

The SSD8IO configuration package ships with the following SSD8IO_MF-specific software:

`ssd8io_mf.bit` VHDL configuration file for the UI FPGA on the PCI CDa board (LVDS or RS422). Implements the multi-function firmware capabilities described herein.

Sample software initialization files (editable text files that you can customize for your own applications) for all board configurations are in the `pcd_config` subdirectory of the distribution directory, including:

`ssd8io_mf.cfg` Sample configuration file to configure the SSD8IO_MF for operation with `ssd8io_mf.bit`.

A compatible 16-channel FPGA configuration file must be loaded in the PCI FPGA on the PCI CDa board. This is:

`cda16_classic.bit` The VHDL configuration file for the PCI FPGA on PCI CDa boards.

To load this file, go to the driver directory installed on the driver.

`pciload –u 0 cda16_classic.bit`

## FPGA Configuration Files

The PCI CDa board implements the DMA interface using two field-programmable gate arrays (FPGAs), referred to as the PCI FPGA and the UI (user interface) FPGA:

The *PCI FPGA* communicates with the host computer over the PCI bus. It implements the DMA engine, which transfers data between the board and the host computer, and loads its firmware on powerup from flash ROM located on the main board.

The *UI FPGA* transfers data between the user device and the PCI FPGA; in some instances, it also sends the data to the mezzanine board. The UI FPGA or mezzanine board may also process the data in some manner, depending on the application.

FPGA configuration files define the firmware required for the PCI FPGA and the UI FPGA.

- *PCI FPGA* configuration files are in the `flash` subdirectory of the EDT top-level distribution directory.
- *UI FPGA* configuration files are in the `bitfiles` subdirectory of the EDT top-level distribution directory.

## The PCD Device Driver

The PCD device driver is the software running on the host that allows the host operating system to communicate with the SSD8IO_MF. The driver is loaded into the kernel upon installation, and thereafter runs as a kernel module. The driver name and subdirectory is specific to each supported operating system; the installation script handles those details for you, automatically installing the correct device driver in the correct operating system-specific manner.

## Sample Applications and Utilities

Along with the driver, the FPGA configuration files, and the software initialization files, the installation disk includes applications and utilities that you can use to initialize and configure the board, access registers, or test the board. For many of these applications and utilities, C source is also provided, so that you can use them as starting points to write your own applications. The most commonly useful are described below.

Software is updated regularly; the latest versions are available on our website at www.edt.com/software.html. We encourage you to use the latest versions for new installations. For existing applications, upgrade only if you have a specific reason to do so.

### *Sample Applications*

| | |
|---|---|
| `rd16` | Performs simple ring buffer input on a specified channel. |
| `wr16` | Performs simple ring buffer output on a specified channel. |
| `simple_read` | Performs DMA input without using ring buffers. Data is therefore subject to interruptions, depending on system performance. |
| `simple_write` | Performs DMA output without using ring buffers. Data is therefore subject to interruptions, depending on system performance. |
| `simple_getdata` | Serves as an example of a variety of DMA-related operations, including reading the data from the connector interface and writing it to a file, as well as measuring input rate. |
| `simple_putdata` | Serves as an example of a variety of DMA-related operations, including reading data from a file and writing it out to the connector interface. |

### *Utility Files*

| | |
|---|---|
| `initpcd` | A utility for initializing and configuring the SSD8IO. |
| `pciload` | A utility for loading the PCI FPGA firmware. |
| `pdb` | Utility application that enables interactive reading and writing of the PCI CDa UI FPGA registers. |

### *Testing Files*

Various C source, executable, and FPGA configuration files are available for testing (see Testing), including at least:

`ssd8iomftest.c` Tests the PCI CDa Multifunction configuration.

## Building Applications

Executable and PCD source files are at the top level of the EDT PCD driver distribution directory. If you need to rebuild an application, therefore, run `make` in this directory.

Windows and Solaris users must install a C compiler. For Windows, we recommend the Microsoft Visual C compiler; for Solaris, the Sun WorkShop C compiler. Linux users can use the `gcc` compiler typically included with your Linux installation. If Solaris or Windows users wish to use `gcc`, contact `tech@edt.com`.

After you've built an application, use the `--help` command line option for a list of usage options and descriptions.

# Testing

After installing a loopback connector (described in the Pinouts section), run the test program

```
ssd8iomftest -u N
```

where N is the unit number of the PCI CDa board under test. The test is menu-driven, and it includes tests to send data across the loopback cable, set the buffer size, and adjust the enable signal block size and the number of inter-block clocks.

For details, see `ssd8iomftest.c`.

# Register Modifications

In the SSD8IO-MF Configuration Package, the following registers are modified from those described in the SSD16IO Addendum.

## 0x21 Ungated Channels Clock Select Register

Size 8-bit

I/O read-write

Access EDT_SS_CLK_SEL

Comment Selects output clock timing source. Internal clock is the default. External clocks let you select an input channel's clock to serve as the output transmit clock for channels 4-7.

| Bit | Name | Description | | | | |
|---|---|---|---|---|---|---|
| 7-0 | | See hexadecimal values below. | | | | |
| | | 0x00 | Internal from PLL1 | 0x0A | External, ch. 9 input clock | |
| | | 0x01 | External, ch. 0 input clock | 0x0B | External, ch. 10 input clock | |
| | | 0x02 | External, ch. 1 input clock | 0x0C | External, ch. 11 input clock | |
| | | 0x03 | External, ch. 2 input clock | 0x0D | External, ch. 12 input clock | |
| | | 0x04 | External, ch. 3 input clock | 0x0E | External, ch. 13 input clock | |
| | | 0x05 | = External, ch. 4 input clock | 0x0F | External, ch. 14 input clock | |
| | | 0x06 | External, ch. 5 input clock | 0x10 | External, ch. 15 input clock | |
| | | 0x07 | External, ch. 6 input clock | 0x20 | External, EXTCLKIN input clock | Not available for ECL |
| | | 0x08 | External, ch. 7 input clock | 0x40 | Enable PLL0 out on EXTCLKIN for board under test (testing only) | |
| | | 0x09 | External, ch. 8 input clock | | | |

## 0x35 Enable Burst Delay Register

Size 8-bit

I/O read-write

Access SSD8IO_MF_BURST_DELAY

| Bit | Name | Description |
|---|---|---|
| 7-0 | BURST_DELAY | For output channels 0-1, sets the number of clock cycles to delay between enable bursts. |

## 0x36 Enable Burst Size Register

Size 8-bit

I/O read-write

Access SSD8IO_MF_BURST_SIZE

| Bit | Name | Description |
|---|---|---|
| 7–0 | BURST_SIZE | For output channels 0 and 1, sets the size of the enable burst bits multiplied by 32. For a 128-bit enable burst, this register would be set to 4. |

## 0x37 Gated Channels Clock Select Register

Size 8-bit

I/O read-write

Access SSD8IO_MF_CLK_SEL

Comment Selects output clock timing source for channels 0 and 1. The internal clock is the default. External clocks let you select an input channel's clock to serve as the output transmit clock for channels 0 and 1.

| Bit | Name | Desription | | | | |
|---|---|---|---|---|---|---|
| 7-0 | | See hexadecimal values below. | | | | |
| | | 0x00 | Internal from PLL1 | 0x0A | External, ch. 9 input clock | |
| | | 0x01 | External, ch. 0 input clock | 0x0B | External, ch. 10 input clock | |
| | | 0x02 | External, ch. 1 input clock | 0x0C | External, ch. 11 input clock | |
| | | 0x03 | External, ch. 2 input clock | 0x0D | External, ch. 12 input clock | |
| | | 0x04 | External, ch. 3 input clock | 0x0E | External, ch. 13 input clock | |
| | | 0x05 | = External, ch. 4 input clock | 0x0F | External, ch. 14 input clock | |
| | | 0x06 | External, ch. 5 input clock | 0x10 | External, ch. 15 input clock | |
| | | 0x07 | External, ch. 6 input clock | 0x20 | External, EXTCLKIN input clock | Not available for ECL |
| | | 0x08 | External, ch. 7 input clock | 0x40 | Enable PLL0 out on EXTCLKIN | |
| | | 0x09 | External, ch. 8 input clock | | for board under test (testing only) | |

## 0x38 Gated Channels Receive Mode Register

Size 8-bit

I/O read-write

Comment Selects receive mode for custom channels. Mode 0 builds 32-bit words, while data valid is true; if valid goes false before a 32-bit word is completed, the upper bits are zero-filled to form a 32-bit word. Mode 1 is identical, except that it limits the number of 32-bit words to the SSD16IO_MF_BURST_SIZE value per strobe of data valid.

| Bit | Name | Description |
|---|---|---|
| 7-1 | | Reserved |
| 0 | | 0 or 1. Selects receive mode 0 or 1 as explained above. |

# Pinouts

The SSD8IO-MF Configuration Package connects your device to the PCI CDa main board via the 80-pin connector, as shown below.

Signals labeled “not used” are connected to wires; your firmware can access these signals.

| Pin | Signal | | Loopback | Pin | Signal | |
|---|---|---|---|---|---|---|
| 1 | ground | | | 41 | ground | |
| 2 | not used | | | 42 | not used | |
| 3 | CH2D+ | IN | | 43 | CH0D+ | OUT |
| 4 | CH2D- | | | 44 | CH0D- | |
| 5 | CH2CLK+ | IN | | 45 | CH0CLK+ | OUT |
| 6 | CH2CLK- | | | 46 | CH0CLK- | |
| 7 | CH3D+ | IN | | 47 | CH1D+ | OUT |
| 8 | CH3D- | | | 48 | CH1D- | |
| 9 | CH3CLK+ | IN | | 49 | CH1CLK+ | OUT |
| 10 | CH3CLK- | | | 50 | CH1CLK- | |
| 11 | CH6D+ | | | 51 | CH4D+ | |
| 12 | CH6D- | | | 52 | CH4D- | |
| 13 | CH6CLK+ | | | 53 | CH4CLK+ | |
| 14 | CH6CLK- | | | 54 | CH4CLK- | |
| 15 | CH7D+ | | | 55 | CH5D+ | |
| 16 | CH7D- | | | 56 | CH5D- | |
| 17 | CH7CLK+ | | | 57 | CH5CLK+ | |
| 18 | CH7CLK- | | | 58 | CH5CLK- | |
| 19 | EXTCLKIN+ | | | 59 | EXTCLKIN- | |
| 20 | not used | | | 60 | not used | |
| 21 | not used | | | 61 | not used | |
| 22 | not used | | | 62 | not used | |
| 23 | not used | | | 63 | not used | |
| 24 | CH8D+ | Ch 0 - | | 64 | CH10D+ | Ch 2 - |
| 25 | CH8D- | ENA | | 65 | CH10D- | SYNC |
| 26 | not used | | | 66 | not used | |
| 27 | not used | | | 67 | not used | |
| 28 | CH9D+ | Ch 1 - | | 68 | CH11D+ | Ch 3 - |
| 29 | CH9D- | ENA | | 69 | CH11D- | SYNC |
| 30 | not used | | | 70 | not used | |
| 31 | not used | | | 71 | not used | |
| 32 | CH12D+ | Ch 0 - | | 72 | not used | |
| 33 | CH12D- | BUSY IN | | 73 | not used | |
| 34 | not used | | | 74 | not used | |
| 35 | not used | | | 75 | not used | |
| 36 | CH13D+ | Ch 1 - | | 76 | not used | |
| 37 | CH13D- | BUSY IN | | 77 | not used | |
| 38 | not used | | | 78 | not used | |
| 39 | not used | | | 79 | not used | |
| 40 | ground | | | 80 | ground | |



Sky Blue Microsystems GmbH
Geisenhausenerstr. 18
81379 Munich, Germany
+49 89 780 2970, info@skyblue.de
www.skyblue.de

In Great Britain:
Zerif Technologies Ltd.
Winnington House, 2 Woodberry Grove
Finchley, London N12 0DR
+44 115 855 7883, info@zerif.co.uk
www.zerif.co.uk