PCI SS/GS SSE

Revision: A
December 2004

Contact

Sky Blue Microsystems GmbH
Geisenhausenerstr. 18
81379 Munich
Germany
www.skyblue.de
info@skyblue.de
Tel. +49 (0) 89 - 780297 0

Control Information

Control Item	Details
Document Owner	Tyler Nguyen
Information Label	**EDT Public**
Supersedes	None
File Location	F:/PCISS/pci_ss_sse.doc
Document Number	008-02382-00

Revision History

Revision	Date	Revision Description	Originator
A	13-Dec-04	New document	S Vasil

Contents

Overview

The PCI SS/GS SSE occupies one slot of a PCI local bus and has one 15-pin D connector. The board has three independent serial channels of one bit each and works in excess of 400 MHz (this may be limited by available PCI bus bandwidth). Drivers for Windows NT/2K/XP, Solaris, and Linux are provided, along with sample application programs.

Each channel accepts a differential data signal (two wires) and a differential clock (two more wires) at standard ECL signal levels. All incoming signals are terminated to –2 volts through 50 ohms.

Channel 0 and 1 are used as inputs only. Channel 2 is used as output only.

Note: The data and clock signal for the output channel are not terminated on this board; therefore, no signals will be present if hooked up to a receiving end without termination.

Included Files

Following is a list of files included on the CD that ship with each PCI SS/GS SSE. There are many more files on the CD that aren't listed here; this list represents files specifically used by the PCI SS/GS SSE.

ss_sse.readme	Instruction file.
ss_sse_framesync.readme	Frame sync and Reed-Solomon decoder notes
sseload.c	A utility for programming the Virtex-II Pro Xilinx and the PLL on the mezzanine board.
eclopt_sse.bit	Configuration file for main board Xilinx
sse_rs_decoder.bit	Configuration file for main board Xilinx with Reed-Solomon decoder
sseio.bit	Configuration file for mezzanine board Xilinx
sseio_asm.bit	Configuration file for mezzanine board Xilinx with ASM check
ssein.bit	Self-test configuration file with internal prbs code generator.

Installation

Software

Turn on the computer *without* the board installed.

Install the standard EDT PCI CD driver software from the CD-ROM or from www.edt.com. Reboot the machine with the board installed. It may be necessary to run some of the configuration files as root. Make sure your path includes the current directory.

Hardware

Insert the PCD SS SSE board into one of your PCI slots. Run `pciload` to list all EDT boards installed on the computer.

The Xilinx file is downloaded to the board's PCI interface Xilinx PROM using the *pciload* program

To see what boards are in the system, run pciload without any arguments:

```
pciload
```

To verify the PCI SS/GS SSE:

1. Navigate to the directory in which you installed the driver.
2. At the prompt, enter:

   ```
   pciload -v
   ```

   This will compare the current PCI Xilinx file in the package with what is currently on the board's PROM.

   Note: If more than one board is installed on a system, use the following, where N is the board unit number:

   ```
   pciload –u N -v
   ```

   Outcome: Dates and revision numbers of the PROM and File ID will be displayed. If these numbers match, there is no need for a field upgrade. If they differ, upgrade the flash PROM as follows:

   a. At the prompt, enter:

      ```
      pciload update
      ```

   b. Shut down the operating system and turn the host computer off and then back on again. The board reloads firmware from flash ROM only during power-up. Therefore, after running *pciload*, the new bit file is not in the Xilinx until the system has been power-cycled; simply rebooting is not adequate.

To see other pciload options, run:

```
pciload help
```

Note: The SSE interface firmware only works with the 4-channel PCI Xilinx firmware.

The following is an example of a correctly configured board:

```
%1> pciload

pcd unit 0 (pci ss-4):
  XC2S200 PCI FPGA, AMD 29LV08AB 8MB FPROM
  s/n ... , p/n ... , i/f fpga xcv600e, rev21 clock 10 Mhz, opt
600e
  Sector 0
  Sector 1
  Sector 2  PROM  id: <pciss4.ncd 2s200fg456 2003/02/08
16:21:23>
  Sector 3  PROM  id: <pciss4_5v.ncd 2s200fg456 2003/02/08
16:21:33>
```

To save current firmware into flash, run:

```
pciload –u <unit number> pciss4
```

The system must be shut down completely then rebooted before the new firmware in flash will take effect.

Pinout

Pin	Description	Input/Output
1	D0+	Input
2	D0–	Input
3	CLK0+	Input
4	CLK0–	Input
5	D2–	Output
6	Not used	
7	D1–	Input
8	CLK1–	Input
9	Ground	
10	D2+	Output
11	Not used	
12	D1+	Input
13	CLK1+	Input
14	CLK2–	Output
15	CLK2+	Output

Testing

One board

To test one board, run:

```
sseload -b ssein
```

This will program the main board with `eclopt_sse.bit` and the mezzanine board with `ssein.bit`. It will also set the PLL to 100 MHz.

The data for the two input channels are generated by an internal PRBS code generator.

Verify the data by running:

```
chkprbs15 -c 0 -n 2 -l 1000
```

Check for zero errors.

You can also set the clock to a different rate by running:

```
sseload -F frequency
```

Note: The frequency is in MHz, and it has to be in the range 50-800.

Board to Board:

To test two boards, run:

```
sseload -u 0
```

then run:

```
sseload -u 1
```

This will program both installed boards with `eclopt_sse.bit` and `sseio.bit` for the actual data acquisition.

Again the PLL of both boards are set to 100 MHz.

Generate PRBS data on one board by running:

```
genprbs15 -u unit -c 2 -n 1 -l 0
```

and verify on the other board by running:

```
chkprbs15 -u unit -c 0 -n 1 -l 1000
```

or (depending on what input channel to be used):

```
chkprbs15 -u unit -c 1 -n 1 -l 1000
```

Registers

The following registers describe the combo16io bitfile:

Command Register

Size 8-bit
I/O read-write
Address 0x00

Bit	Name	Description
2-0		Not used
3	CMD_EN	Enables the bitfile.
7-4		Not used

Command Register (mezzanine)

Size 8-bit
I/O read-only
Address 0xA0
Comments The value of the register reflects the value of the Command register on the mezzanine board.

Configuration 1 Register

Size 8-bit
I/O read-write
Address 0x0F

Bit	Name	Description
0	BSWAP	Byte swap
2-1		Not used
3	SSWAP	Short swap
7-4		Not used

Channel Enable Registers

Size 8-bit
I/O read-write
Address 0x10 and 0x11

Bit	Name	Description
2-0	CH_ENABLE	A 1 in a bit enables the corresponding DMA channel serial to parallel converter and internal PRBS15

	checker. The code checker is reset when bit is zero.
7–3	Not used.

Channel Enable Register (mezzanine)

Size	8-bit
I/O	read-only
Address	0xB0
Comments	The value of the register reflects the value of the Channel Enable register on the mezzanine board.

Least Significant Bit First Register

Size	8-bit
I/O	read-write
Address	0x16

Bit	Name	Description
2-0	LSB_First	When set, each bit of the DMA channel is moved toward the least significant bit of the parallel data word. When 32 bits are received, the word is written to computer memory. When cleared, the most significant bit of a 32-bit word is the first bit. Byte swap and short swap can also affect the order of bytes in a 32-bit word.
7–3		Not used.

Underflow Register

Size	8-bit
I/O	read-only
Address	0x18

Bit	Name	Description
1–0		Not used.
2	UNDERFLOW	When set to 1, this channel's transmit data (channel 2) has underflowed the internal FIFO since the last CMD_EN, CHANNEL_ENABLE. Reset by disabling channel and then enabling this register.
7–3		Not used.

Overflow Register

Size 8-bit

I/O read-only

Address 0x1A

Bit	Name	Description
1-0	OVERFLOW	When set to 1, this channel's data has overflowed the internal FIFO since the last CMD_EN, CHANNEL_ENABLE. Reset by disabling channel and then enabling this register.
7–2		Not used.

PLL Programming Register

Size 8-bit

I/O read-write

Address 0x20

Comments The program set_ss_vco uses this register to program the serial interface of the four PLL.

Bit	Name	Description
3-0	PLL_STROBE	Connected to the strobe inputs of PLL 3 to 0, respectively.
5-4		Not used
6	PLL_DATA	Connected to all four PLL serial data inputs.
7	PLL_SCLK	Connected to all four PLL serial clock inputs.

PLL 0 Divider Registers

Size 8-bit

I/O read-write

Address 0x24 and 0x25

Comments This register is set by set_ss_vco. It is a post-scalar divider used to achieve lower frequencies than the PLLs can be programmed to. After this division, the clocks are divided by two again to even the duty cycle. Set_ss_vco considers all these effects. PLL0 is used as the clock for the Reed-Solomon decoder.

Note: p110_clock / (serial_clock/8) needs to be greater than 660/255.

Bit	Name	Description
15–0	PLL0DIV	Programmable post-divider.

Board ID Register

Size	8-bit
I/O	read-only
Address	0x7F
Access	EDT_BOARDID
Comments	This register returns a unique four-bit code corresponding to the mezzanine board installed. The codes used are:

F	Combo I/O board
E	Combo II I/O board, RS422
D	Combo III I/O board, LVDS
B-9	Reserved
8	ECL I/O
4	Reserved for SSE (high-speed serial ECL) I/O
1	LVDS I/O board
0	RS422 I/O board

Bit	Name	Description
3-0	BOARD_ID	Reads back the ID of the mezzanine board installed.
7-4		Not used; always reads 0.

PLL Programming Register

Size	8-bit
I/O	read-write
Address	0x80
Comments	The program `sseload` uses this register to program the high-speed PLL on the mezzanine board. The reference clock iput of the PLL comes from a 16 MHz oscillator.

Bit	Name	Description
0	SLOAD	Connected to the sload input of the PLL.
1	SDATA	Connected to the sdata input of the PLL.
2	SCLK	Connected to the sclk input of the PLL.
7–3		Not used.

LED Control Register

Size	8-bit
I/O	read-write
Address	0x82
Comments	This register controls the two LEDs on the mezzanine board.

Bit	Name	Description
0	BLINK	Set to 1 to “blink” one of the two LEDs.
7–1		Not used.

Clock Control Register

Size	8-bit
I/O	read-write
Address	0x84
Comments	This register selects the clock source for the output channel.

Bit	Name	Description
0	CLK_ENABLE	Set to 1 to turn off the output clock.

1	CLK_SELECT	Set to 0 to use high-speed PLL (50–800 MHz).
		Note: To set, run:
		`sseload –u <unit> -F <freq>`
		Set to 1 to use PLL1 from main board to go below 50 MHz.
		Note: To set, run:
		`set_ss_vco –u <unit> -F <freq> 1`

7–2		Not used.

Frame Synchronization/Reed-Solomon Decoder

The Reed-Solomon decoder conforms to Consultative Committee for Space Data Systems (CCSDS) 101.0-B-6 (255,223) with a five-way interleave.

The Reed-Solomon decoder is embedded in input channel 0.

The PCI SS/GS SSE input channels support three modes of operation:

1. Send raw data to the host (input channels 0 and 1)(default)
2. Detect synchronization, then send raw data to the host in multiples of the frame size (input channel 0 and 1).
3. Detect synchronization, decode the data, then send the decoded data frames to the host (input channel 0 only).

Configuration Files

sse_rs_decoder.bit	Main board Xilinx configuration file
sseio_asm.bit	Mezzanine board Xilinx configuration file

Run `sseload -R` to program the main board with `sse_rs_decoder.bit` which has an internal Reed-Solomon Decoder.

The mezzanine board will also be programmed with `sseio_asm.bit` which checks for the attached synchronization marker (also called the frame synchronization pattern) before sending a block of data to the decoder.

Note: PLL 0 is used as the decoding clock for the Reed-Solomon Decoder and needs to be set to 66 MHz in order to keep up with a 200 MHz serial stream. This is necessary because the decoder has a processing delay of 660 symbol clocks. With a 200 megabit per second bitstream (25 megabytes per second), the decoder needs to process the data at a much faster rate so it can decode continuously. The rule is:

```
(Pll0's freq / (serial_freq/8) ) >(greater) 660/255
```

After running `sseload -u unit -R`, set the decoder clock by running:

```
set_ss_vco -u unit -F 66000000 0
```

Sending a File

The `wfile.c` is a simple program that reads from a file and sends the data from the file to the output DMA channel. Run:

```
wfile -u unit -c channel file_name
```

Reading and Storing to a File

The `rfile.c` is a simple program that reads the data from the input DMA channel and writes the data to a file. Run:

```
rfile -u unit -c channel file_name
```

Note: Both `rfile` and `wfile` need to be added to the `makefile` targets.

Reed-Solomon Decoder Registers

Synchronization Pattern Registers

Size	8-bit each
I/O	read-write
Address	0x86, 0x87, 0x88, 0x89

Bit	Name	Description
7–0	SYNC[7–0]	These four 8-bit registers comprise a 32-bit synchronization pattern.
15–8	SYNC[15–8]	
23–16	SYNC[23-16]	
31–24	SYNC[31–24]	

Mask Registers

Size	8-bit each
I/O	read-write
Address	0x90, 0x91, 0x92, 0x93
Comments	MASK is low by default, thus it ignores all bits of the synchronization pattern.

Bit	Name	Description
7–0	MASK[7–0]	These four 8-bit registers comprise a 32-bit mask allowing some bits of the synchronization pattern to be ignored.
15–8	MASK [15–8]	
23–16	MASK [23-16]	
31–24	MASK [31–24]	

Frame Length Registers

Size	8-bit each
I/O	read-write
Address	0x94, 0x95
Comments	A value of 0 reflects a frame length of 65536.

Bit	Name	Description
7–0	FRAMEL [7–0]	This 16-bit number defines the frame length size.
15–8	FRAMEL [15–8]	

Check/Flywheel Count Register

Size	8-bit each
I/O	read-write
Address	0x98

Bit	Name	Description
3–0	CHKCNT	Check frame count (0–15)
7–4	FLYCNT	Flywheel frame count (0–15)

Control Register

Size	8-bit each
I/O	read-write
Address	0x99
Comments	This register controls the mode of operation.

Bit	Name	Description
0	FS_DET	Set to 1 to turn on frame synchronization.
1	RS_EN	Set to 1 to turn on Reed-Solomon decoder.
2		Not used.
3	STAT_EMBED	Set to 1 to append status such as current state of the frame synchronization engine and/or Reed-Solomon error statistics to the end of the data frames.
7–4		Not used.

Notes:

- RS_EN only works on input channel 0. There is no Reed-Solomon decoder on input channel 1.
- When RS_EN is high, incoming encoded Reed-Solomon data is assumed to conform to CCSDS (255,223) with a five-way interleave, so the frame length is fixed at 255*5.
- When RS_EN is low and FS_DET is high, the frame length is set by the 16-bit FRAMEL register. Data is sent to host as raw but in multiples of the frame size.
- When both RS_EN and FS_DET are low, data is sent to the host as raw without any boundary guarantee.
- Embedding status into DMA data streaming (STAT_EMBED='1'):
  - When RS_EN is 1, two status bytes will be appended to the end of the decoded data frames (size is fixed at 223*5):
    - First byte: frame sync status
      - Bit 0: 1means lock, 0 means flywheel

- Bits 7-4: current flywheel count if in flywheel

- Second byte: error statistics
  - Since there are five blocks in one frame (interleave=5), the lower five bits determine whether they're good or bad; 1 means bad and 0 means good.
  - Bit 0 => 1st block
  - Bit 1 => 2nd block
  - Bit 4 => 5th block
- Good means that the decoder was able to correct corrupted data if there were any.
- Bad means that the decoder failed to correct corrupted data. A bad block has more errors than the Reed-Solomon decoder can correct.

- When FS_DET is 1 and RS_EN is 0:
  - One status byte will be appended to the end of the data frames, size is set by FRAMEL register
  - Bit 0: 1 means lock, 0 means flywheel
  - Bit 7-4: current flywheel count if in flywheel

Contact

Sky Blue Microsystems GmbH
Geisenhausenerstr. 18
81379 Munich
Germany
www.skyblue.de
info@skyblue.de
Tel. +49 (0) 89 - 780297 0