Default firmware for the SVEC card

Programmer's manual

CERN BE-CO-HT / Tomasz Włostowski

Table of Contents

1 Introduction

This document describes the default bitstreams that come preloaded with every SVEC card. As there are two FPGAs on the SVEC, there are two default bitstreams:

- **Bootloader bitstream**, residing in the System FPGA whose role is to start up the Application FPGA
- **Golden bitstream**, residing in the Application FPGA, which allows the device driver to enumerate the mezzanines.

2 The Bootloader

The System FPGA bootloader allows to boot the Application FPGA from the VME bus or from the onboard Flash memory and reprogram both the System and Application bitstreams in the flash via VME. The boot process goes as follows:

- Power up.
- SFPGA checks for presence of a valid bitstream file for the Application FPGA in the Flash memory.
- If a valid bitstream has been found, the AFPGA is initialized with it,
- if not, the Bootloader enters passive mode. Upon reception of a boot sequence, if gives access to the Flash for the host or lets it program the AFPGA directly.

2.1 VME Interface

The bootloader core supports only 32-bit data CR/CSR accesses from/to address range `0x70000 - 0x70020`, allowing for plug&play reprogramming of the cards only knowing their physical slot locations. All other transfers are ignored. The base address is `0x70000`, and corresponds to the `CSR` register. When the card is powered up, the VME interface stays in passive mode, monitoring VME accesses without ACKing them. This is to prevent conflicts with the CR/CSR space of the VME core in the Application FPGA.

2.2 Entering bootloader mode

In order to enter the bootloader, one needs to write a magic sequence of 8 following transfers: `0xde`, `0xad`, `0xad`, `0xbe`, `0xca`, `0xfe`, `0xba`, `0xbe` to `BTRIGR` register. Afterwards, read the `IDR` register. It should be equal to `SVEC` ASCII string encoded in HEX. Any other value indicates that the boot trigger sequence was not correctly recognized.

Note 1: Triggering bootloader mode causes automatic reset (un-programming) of the Application FPGA.

Note 2: Write operations to `BTRIGR` register while the core is still in passie mode will not be acknowledged on the VME bus and will most likely cause bus errors. Please ignore them.

2.3 Programming the AFPGA

Programming the Application FPGA directly via VME involves following steps:

- Reset the Xilinx Passive Serial boot interface by writing `CSR.SWRST` bit,
- Set download speed by writing `CSR.CLKDIV` field. Default value is `1`.
- Write `CSR.START` bit and set endianess via `CSR.MSBF` bit,
- Write the bitstream to the FIFO registers, observing FIFO full/empty status. Last transfer should have `FIFO_R1.XLAST` bit set to 1.

- Wait for assertion of `CSR.DONE`. `CSR.ERROR` bit indicates a problem during configuration (most likely, an invalid bitstream)
- Exit bootloader mode by writing 1 to `CSR.EXIT` bit.

Successful firmware download to the Application FPGA is indicated by turning on the "AFPGA DONE" LED.

2.4 Programming the Flash

SFPGA also allows raw access to the Flash memory (M25P128) via `FAR` register. The code below shows how to execute a single SPI transaction (command + N data bytes).

```
uint8 spi_transfer(int cs, uint8 data) {
  while (! FAR.READY);
  FAR.CS = cs;
  FAR.XFER = 1;
  FAR.DATA = data;
  while (! FAR.READY);
  return FAR.DATA;
}

void flash_command(uint8 command, uint8 data[]) {
  spi_transfer(0, 0);
  spi_transfer(1, cmd);
  for(i=0;i<length(data);i++)
    data[i] = spi_transfer(1, data[i]);
  spi_transfer(0, 0);
}
```

Low-level details about programming M25Pxxx series Flash memories can be found in their datasheets.

Note 1: It is advised to protect the System FPGA bitstream from being accidentally overwritten, as this will result in bricking the card and will require re-programming the flash via JTAG.

Note 2: The freshly-programmed bitstreams will be loaded into the FPGAs after power-cycling the card.

2.5 Flash memory organization

The flash memory of the SVEC contains 16 Megabytes of memory, that is 64k pages of 256 bytes. First 6 MBs are used for bitstream storage, the rest is available for the user application. The flash format is compatible with the SDB filesystem, with the SDB descriptor table located at offset `0x500000`. Locations of the bitstreams are fixed to:

- `0`: Raw bitstream for the System FPGA (up to 1 MB)
- `0x100000`: Bitstream for the Application FPGA (up to 4 MB)

An example script for building the default flash filesystem (containg the bootloader and golden bitstreams) is located in the `software/sdb-flash` subdirectory. The presence of SDB descriptor table at `0x500000`, as the SDB signature is checked by the bootloader to prevent booting up from a corrupted flash.

Note: Both bitstreams must be in raw (`.bin` file extension) format. `.bit`, `.mcs`, `.xsvf` and other formats will not work. This also applies to direct VME boot mode.

3 The Golden Bitstream

The SVEC Application FPGA golden bitstream allows the SVEC device driver to:

- Query the board's serial number,
- Check the presence of the FMC mezzanines,
- Read out their I^2C identification EEPROMs.

The bitstream does not drive any of the mezzanine user/clock pins to protect from electrical damage resulting from mismatched I/O standards.

3.1 Block diagram

The bitstream encompasses I^2C, OneWire and GPIO modules from the `general-cores` library. For further details, refer to the library's manual.

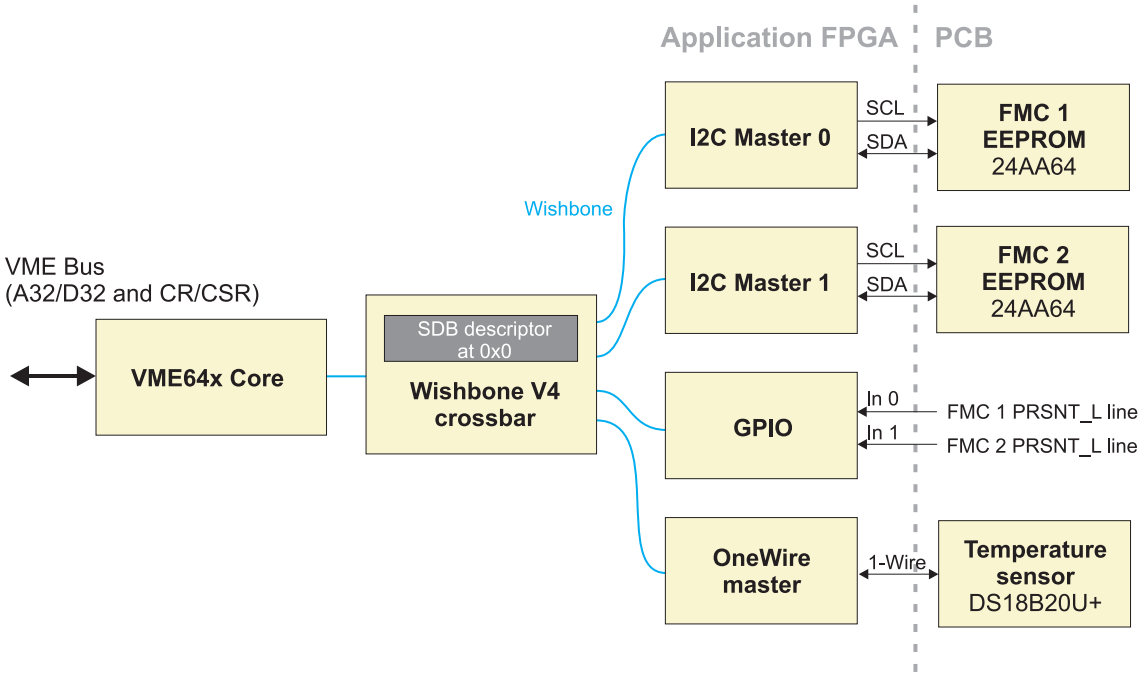

Figure 3.1: Block diagram of the SVEC golden gateware.

Presence detection is done by reading out the `PRSNT` lines (active low) through the GPIO port. EEPROM readout is performed via the two I^2C masters. Board serial number is equal to the serial number of the DS18B20U+ temperature sensor, accessible via the OneWire master. The clock freqency for `I^2C` and OneWire dividers calculation is 62.5 MHz.

3.2 Memory map

Warning: Please do not hardcode the base addresses of the cores, query them from the SDB descriptor. The SDB address of `0x0` is guaranteed to stay constant. Only A32/A24/D32/CSR address modifiers are supported.

Core	Base address	Library	Description
`sdb_rom`	`0x0`	`general-cores`	SDB descriptor.
`xwb_i2c_master`	`0x10000`	`general-cores`	I2C Master for accessing FMC 1 identification EEPROM.

xwb_i2c_master	0x11000	general-cores	I2C Master for accessing FMC 2 identification EEPROM.
xwb_onewire_master	0x12000	general-cores	OneWire Master for accessing temperature sensor/serial ID.
xwb_gpio_port	0x13000	general-cores	GPIO port for accessing FMC1/2 presence lines.

Appendix A System FPGA register map

A.0.0.1 Memory map summary

Address	Type	Prefix	Name
0x0	REG	CSR	Control/status register
0x4	REG	BTRIGR	Bootloader Trigger Register
0x8	REG	FAR	Flash Access Register
0xc	REG	IDR	ID Register
0x10	FIFOREG	FIFO_R0	FIFO 'Bitstream FIFO' data input register 0
0x14	FIFOREG	FIFO_R1	FIFO 'Bitstream FIFO' data input register 1
0x18	REG	FIFO_CSR	FIFO 'Bitstream FIFO' control/status register

A.0.0.2 CSR - Control/status register

Bits	Access	Prefix	Default	Name
0	W/O	START	X	Start configuration
1	R/O	DONE	X	Configuration done
2	R/O	ERROR	X	Configuration error
3	R/O	BUSY	X	Loader busy
4	R/W	MSBF	X	Byte order select
5	W/O	SWRST	X	Software resest
6	W/O	EXIT	X	Exit bootloader mode
13...8	R/W	CLKDIV	X	Serial clock divider

Field	Description
START	write 1: starts the configuration process. write 0: no effect
DONE	read 1: the bitstream has been loaded read 0: configuration still in progress
ERROR	read 1: an error occured during the configuration (DONE/INIT_B timeout) read 0: configuration was successful
BUSY	read 1: the loader is busy (can't start configuration yet) read 0: the loader is ready to re-configure the FPGA
MSBF	write 1: MSB first (big endian host) write 0: LSB first (little endian host)
SWRST	write 1: resets the loader core write 0: no effect
EXIT	write 1: terminate bootloader mode and go passive (VME only)
CLKDIV	CCLK division ratio. CCLK frequency = F_sysclk / 2 / (CLKDIV + 1)

A.0.0.3 BTRIGR - Bootloader Trigger Register

Bits	Access	Prefix	Default	Name
7...0	W/O	BTRIGR	X	Trigger Sequence Input

Field	Description
BTRIGR	Write a sequence of 0xde, 0xad, 0xbe, 0xef, 0xca, 0xfe, 0xba, 0xbe to enter bootloader mode (VME only)

A.0.0.4 FAR - Flash Access Register

Provides direct access to the SPI flash memory containing the bitstream.

Bits	Access	Prefix	Default	Name
7...0	R/W	DATA	X	SPI Data
8	R/W	XFER	X	SPI Start Transfer
9	R/O	READY	X	SPI Ready
10	R/W	CS	X	SPI Chip Select

Field	Description
DATA	Data to be written / read to/from the flash SPI controller.
XFER	write 1: initiate an SPI transfer with an 8-bit data word taken from the DATAfield write 0: no effect
READY	read 1: Core is ready to initiate another transfer. DATA field contains the data read during previous transaction. read 0: core is busy
CS	write 1: Enable target SPI controller write 0: Disable target SPI controller

A.0.0.5 IDR - ID Register

Bits	Access	Prefix	Default	Name
31...0	R/O	IDR	X	Identification code

Field	Description
IDR	User-defined identification code (g_idr_value generic)

A.0.0.6 FIFO_R0 - FIFO 'Bitstream FIFO' data input register 0

Bits	Access	Prefix	Default	Name
1...0	W/O	XSIZE	X	Entry size
2	W/O	XLAST	X	Last xfer

Field	Description
XSIZE	Number of bytes to send (0 = 1 byte .. 3 = full 32-bit word)
XLAST	write 1: indicates the last word to be written to the FPGA

A.0.0.7 FIFO_R1 - FIFO 'Bitstream FIFO' data input register 1

Bits	Access	Prefix	Default	Name
31...0	W/O	XDATA	X	Data

Field	Description
XDATA	Subsequent words of the bitstream

A.0.0.8 FIFO_CSR - FIFO 'Bitstream FIFO' control/status register

Bits	Access	Prefix	Default	Name
16	R/O	FULL	X	FIFO full flag
17	R/O	EMPTY	X	FIFO empty flag
18	W/O	CLEAR_BUS	X	FIFO clear
7...0	R/O	USEDW	X	FIFO counter

Field	Description
full	1: FIFO 'Bitstream FIFO' is full 0: FIFO is not full
empty	1: FIFO 'Bitstream FIFO' is empty 0: FIFO is not empty
clear_bus	write 1: clears FIFO 'Bitstream FIFO write 0: no effect
usedw	Number of data records currently being stored in FIFO 'Bitstream FIFO'