SPARTAN-II FPGA board with USB 2.0 interface

C1030-1202

Version	**1.2**
Date	**Januar 23, 2003**
Author	**Manfred Kraus**
Email	**support@cesys.com**
Updates	**http://www.cesys.com**

CESYS Gesellschaft für angewandte Mikroelektronik mbH
Zeppelinstrasse 6a
D – 91074 Herzogenaurach
Germany

Overview

1.1 Introduction

The USB2FPGA board is suited for the following applications:

- Design Demonstration
- Training and Education
- Rapid Prototyping
- Job lot production

1.2 Features

- XILINX XC2S200-5PQ208C FPGA
- 1 Mbyte SRAM (15 ns) connected to the FPGA
- 128 kByte SRAM connected to the USB 2.0 Controller
- USB 2.0 compliant device (Plug-and-Play)
- selectable self-powered or bus-powered
- user programmable clock source 1 MHz – 250 MHz
- Expansion port (96 pins)
- 4 Leds
- all FPGA pins routed to test connectors
- Driver for WIN 2000/XP included
- Sample code (Sourcecode) and Test-program included
- schematics included

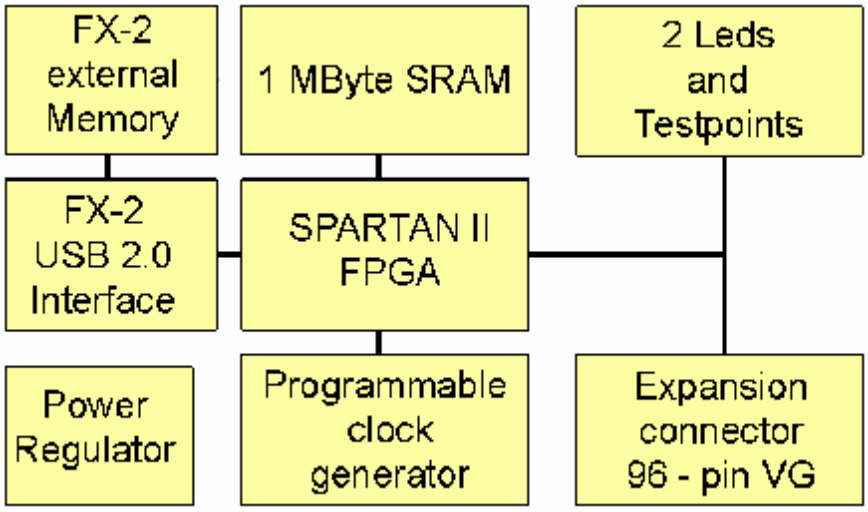
1.3 Requirenments

- PC with USB 2.0 interface running WIN XP or WIN 2000
- 10 Mbyte free harddisk space
- 128 Mbyte memory
- USB2FPGA board with USB-cable
- CESYS USB 2.0 drivers

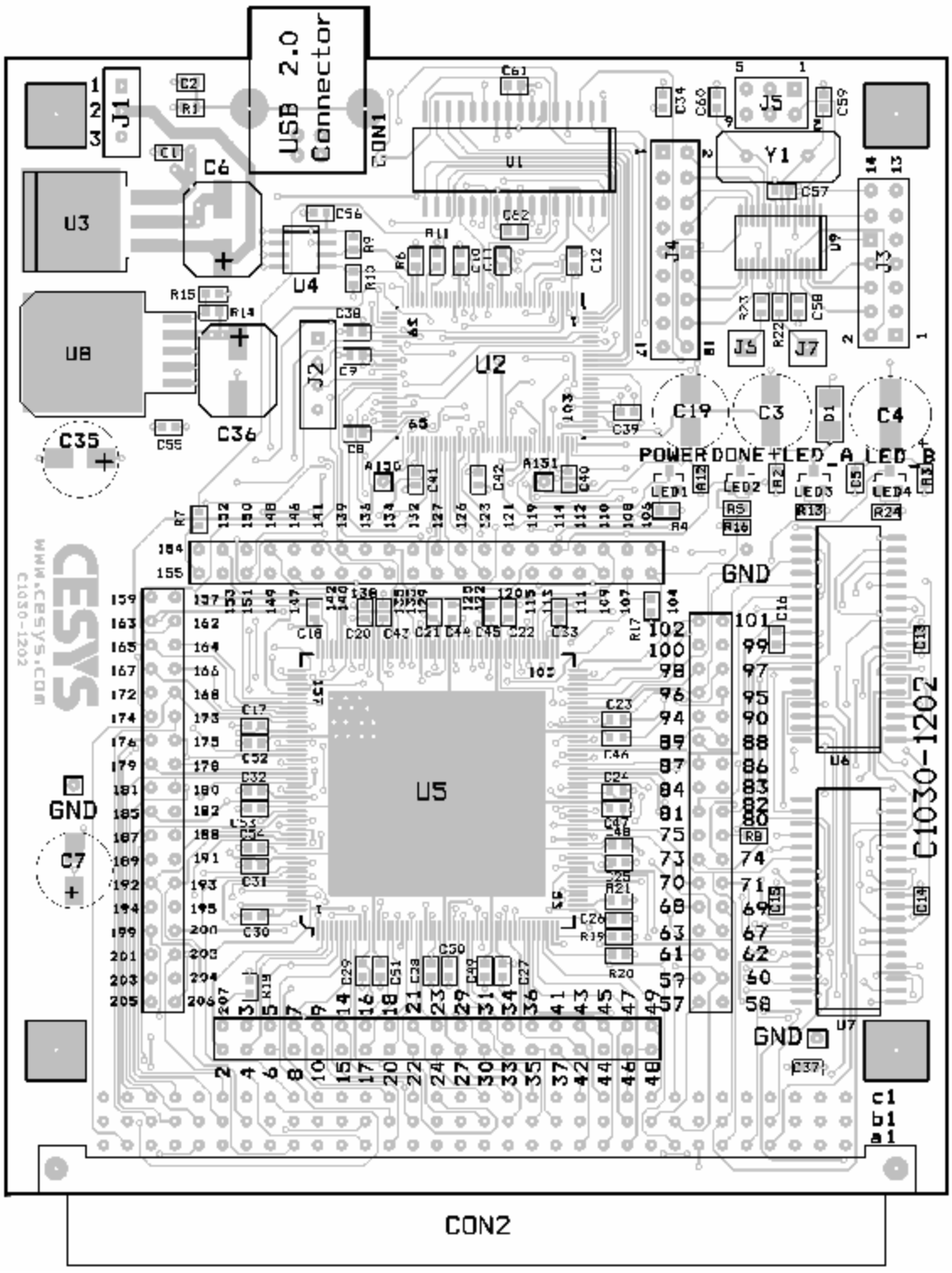
1.4 FPGA Design Tools

To simulate and synthesize your FPGA design you need appropriate tools. Xilinx offers a toolset called “ISE WebPack 5.1” free of charge on their website: http://www.xilinx.com. The ISE WebPack fully supports the XC2S200 Spartan-II FPGA. There are also other commercial tools available from Xilinx and various other vendors.

1.5 Block diagram



1.6 Connector diagram



1.7 Available FPGAs in Standard Version

The USB2FPGA development board is available with the following SPARTAN-II FPGA: XC2S200-5PQ208C

Device	Logic Cells	Typical System Gate Range (Logic and RAM)	CLB array	Total CLBs	Total Block Ram Blocks	Total Block RAM Bits
XC2S200	5,292	71,000 – 200,000	28x42	1,176	14	57,344

1.8 Description

USB2FPGA can be used as a development platform for designs with Xilinx SPARTAN II FPGAs as well as a component for job lot production.
A 96-pin VG connectors allows the attachment of external hardware to the FPGA.
The board is equipped with a XC2S200-5PQ208C XILINX FPGA, a member of the Spartan II family. This programmable logic device receives its internal functions after it has been configured by downloading a bit-stream that represents the design. The change of logic functions is possible at any time without the need of a device-programmer.
When the USB 2.0 interface of the USB2FPGA board is connected to a PC, the FPGA may be loaded with the desired configuration. The software that comes with the board permits to load new configurations anytime.
An jumper-programmable clock oscillator supplies a basic clock that can be used by the FPGA. This clock can be further multiplied or divided by the DLL inside the FPGA.
The 96-pin VG expansion connector of the USB2FPGA allows connections to I/O pins of the FPGA as well as to 3.3 V and GND. Many extensions can be attached directly without the need of an additional external power supply.

1.9 USB 2.0 Interface

The USB 2.0 interface of the board is implemented using an additional device – the CYPRESS FX-2. FPGA designs do not need to include USB specific code. Developers do not need to know the details of the USB bus. To enable communication between the FPGA and a user program running on the PC, the GPIF (General Purpose InterFace) of the FX-2 device is used.

v If your design works “stand-alone” and does not require any communication with the PC, you may ignore the USB interface details and use it only for downloading your design.

2 FPGA pin usage

2.1 FPGA I/O Pins

All FPGA I/O Pins use the I/O Standard LVTTL (3,3 Volt) but they accept 5 Volt Input signals without the need of level shifters or series resistors. Because the VCCO inputs of all banks are tied together in the PQ208 package, they are hardwired to 3,3 Volt on the USB2FPGA.

2.2 Leds

LEDs	
Led 1	Pin PA0 of FX-2 USB controller
Led 2	FPGA DONE Pin
Led 3 (also known as LED_A)	FPGA I/O Pin 102
Led 4 (also known as LED_B)	FPGA I/O Pin 101

Led 1 is controlled by the Firmware.
Led 2 lights up after the FPGA received a configuration bitstream.
Leds 3 and 4 light up when there is a low level at the corresponding FPGA Pin. The meaning of this LEDs is defined by the user's FPGA design.

2.3 Pinout Expansion Port CON2

The 96-pin "VG96 abc reverse" external expansion connector (DIN 41612) is of type "female". Please use the connector diagram to indicate pin 1. On some connectors, the numbers are printed upside down.
Mating connectors among others are: RS Components 476-025 or Farnell 104-986 or HARTING order number 0903 196 7921.
Each individual pin of the FPGA can be configured as input, output, or bi-directional. Make sure your FPGA design does not drive pins that should be an input and are already driven by external connected logic. This is also important for bi-directional signals.

CON 2 96-pin VG Expansion connector			
Pin	A	B	C
1	V5EXT	V5EXT	V5EXT
2	GND	GND	GND
3	V3	V3	V3
4	¬ FPGA I/O Pin 97	¬ FPGA I/O Pin 96	¬ FPGA I/O Pin 95
5	¬ FPGA I/O Pin 94	¬ FPGA I/O Pin 90	¬ FPGA I/O Pin 89
6	¬ FPGA I/O Pin 87	¬ FPGA I/O Pin 69	¬ FPGA I/O Pin 68
7	¬ FPGA I/O Pin 67	¬ FPGA I/O Pin 63	¬ FPGA I/O Pin 62
8	¬ FPGA I/O Pin 61	¬ FPGA I/O Pin 60	¬ FPGA I/O Pin 59
9	¬ FPGA I/O Pin 58	FPGA I/O Pin 57	FPGA I/O Pin 49
10	FPGA I/O Pin 48	FPGA I/O Pin 47	FPGA I/O Pin 46
11	FPGA I/O Pin 45	FPGA I/O Pin 44	FPGA I/O Pin 43
12	FPGA I/O Pin 42	FPGA I/O Pin 41	FPGA I/O Pin 37
13	FPGA I/O Pin 36	FPGA I/O Pin 35	FPGA I/O Pin 34
14	FPGA I/O Pin 33	FPGA I/O Pin 31	FPGA I/O Pin 30
15	FPGA I/O Pin 29	FPGA I/O Pin 27	FPGA I/O Pin 24
16	FPGA I/O Pin 23	FPGA I/O Pin 22	FPGA I/O Pin 21
17	FPGA I/O Pin 20	FPGA I/O Pin 18	FPGA I/O Pin 17
18	FPGA I/O Pin 16	FPGA I/O Pin 15	FPGA I/O Pin 14
19	FPGA I/O Pin 10	FPGA I/O Pin 9	FPGA I/O Pin 8
20	FPGA I/O Pin 7	FPGA I/O Pin 6	FPGA I/O Pin 5
21	FPGA I/O Pin 4	FPGA I/O Pin 3	FPGA I/O Pin 206
22	GND	GND	GND
23	FPGA I/O Pin 205	FPGA I/O Pin 204	FPGA I/O Pin 203
24	FPGA I/O Pin 202	FPGA I/O Pin 201	FPGA I/O Pin 200
25	FPGA I/O Pin 199	FPGA I/O Pin 195	FPGA I/O Pin 194
26	FPGA I/O Pin 193	FPGA I/O Pin 192	FPGA I/O Pin 191
27	FPGA I/O Pin 189	FPGA I/O Pin 188	FPGA I/O Pin 187
28	FPGA I/O Pin 181	FPGA I/O Pin 180	FPGA I/O Pin 179
29	FPGA I/O Pin 178	FPGA I/O Pin 176	FPGA I/O Pin 175
30	FPGA I/O Pin 174	FPGA I/O Pin 173	FPGA I/O Pin 172
31	FPGA I/O Pin 168	FPGA I/O Pin 167	FPGA I/O Pin 166
32	GND	GND	GND

V5EXT: if Jumper J1 is set to Position 1-2 (self powered mode), a 5 VDC power-supply must be connected here. For bus-powered applications this pins can be left unconnected.

V3: This pin is connected to the 3,3 Volt power supply of the board. Depending on the loaded FPGA design it can source up to 200 mA.

¬ **Attention:** THIS PINS ARE NOT 5 Volt tolerant !
This pin is also connected to the on-board RAM. It can only be used when the RAM is not needed.

2.4 FPGA Clock signals

FPGA clock signals	
FX2CLK	FPGA I/GCK0 Pin 80
IFCLK	FPGA I/GCK1 Pin 77
GCK3	FPGA I/GCK2 Pin 182
PLLCLK	FPGA I/GCK3 Pin 185

There are 4 clock sources on the USB2FPGA evaluation board. No matter which of them you use as the main clock for your design, you should synchronize all incoming asynchronous signals to it with a FlipFlop before using them internally. If you fail to do so, your design may work sometimes but not every time. One-hot state machines might lose their “hot”-state and become inoperable. Encoded state machines might enter wrong or illegal states.

FX2CLK

This is the clock, the USB controller FX2 uses. If you want to transmit or receive data using the USB interface, it is the easiest way to choose this clock as main clock for your design. The FPGA looks like external memory to the EZ-USB and all involved signals (Read, Write, Chipselect, ...) are synchronous to the EZ-USB’s clock.

IFCLK

This is the interface clock of the USB controller FX2 GPIF. If you want to transmit or receive data using the USB interface, it is the easiest way to choose this clock as main clock for your design. It defaults to 48 MHz, but its frequency can be altered by the FX-2 firmware.

GCK3

The GCK3 clock signal is connected to the FX-2 USB controller’s Timer 2 output. To use it, the firmware has to enable Timer 2.

PLLCLK clock signal

The frequency of the PLLCLK clock signal can be adjusted by the user (see the description of the PLLCLK oszillator below). The PLLCLK frequency and phase is independent of the FX-2 clock frequencies FX2CLK and IFCLK).

2.5 Power supply

The USB2FPGA board can be configured to be self powered or bus powered.

J1 Power source select	
Pin 1 – 2	self powered
Pin 2 – 3	bus powered

The default setting is “bus powered”. This means the power is provided by the USB bus. If the USB2FPGA is the only device on the USB bus, most computers should allow a maximum current of 400 to 500 mA. This may not be true for notebooks.

The option “self powered” requires an external power supply on CON2. Use this method if your design draws more current than your USB bus can deliver.
Attention: Be careful when using the external power connector. If you apply more than 5,5 Volts or if you reverse the polarity, the board will permanently fail and may not be reparable.

3 User Programmable clock generator

The PLLCLK oscillator may be programmed to any frequency between 1 MHz and 250 MHz by setting Jumpers on J3, J4 and J5. Refer to the ICS525-02 datasheet for details.
To determine the Jumper positions for a given frequency, use the online ICS525 calculator at www.icst.com/products/ics525inputForm.html.

The input frequency is 24 MHz. VDD is 3.3 Volt.

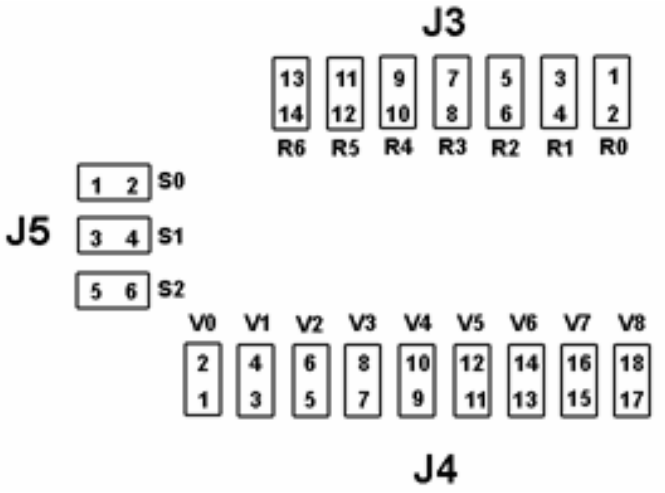


Figure 1 *Jumper J3, J5 and J4 Layout.*

The following tables show examples for common settings.

Attention: The value 0 means set the jumper, 1 means leave the jumper position empty.

Frequency [MHz]	S 2	S 1	S 0	R 6	R 5	R 4	R 3	R 2	R 1	R 0	V 8	V 7	V 6	V 5	V 4	V 3	V 2	V 1	V 0
14,31818	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	1
32,00000	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
40,00000	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0
60,00000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1
100,0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
133,0000	1	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	0	1

For Designs, that need to be clocked faster than 40 MHz, consider using the SPARTAN-II DLLs to double the input clock frequency internally.

4 Testpins

4.1 FPGA Testpins

All FPGA pins are routed to testpoints to ease the connection of measurement equipment like Logic Analyzers. The relationship between FPGA pins and Testpoints is printed on the USB2FPGA board and shown in the Connector-Diagram above.

4.2 Other testpins

Testpins	
J6	PLLCLK
J7	Basic clock source of the FX-2 (24 MHz)
TP1, TP2, TP3	GND

4.3 FX-2 interface to the FPGA

FX-2 Pin number	Signal name	FPGA Pin number
89	FIFOADR0_BUSY	FPGA I/O Pin 154
90	FIFOADR1_INT	FPGA I/O Pin 107
91	PKTEND	FPGA I/O Pin 140
92	FLAGD	FPGA I/O Pin 141
108	TIMER0	FPGA I/O Pin 133
109	DONE	FPGA Done Pin 104
110	GCK3	FPGA GCK2 Pin 182
112	PROGRAM	FPGA PROGRAM Pin 106
4	SLRD	FPGA I/O Pin 139
5	SLWR	FPGA I/O Pin 138
6	RDY2	FPGA I/O Pin 111
7	RDY3	FPGA I/O Pin 112
8	RDY4	FPGA I/O Pin 113
69	FLAGA	FPGA I/O Pin 136
70	FLAGB	FPGA I/O Pin 110
71	FLAGC	FPGA I/O Pin 109
72	GPA0	FPGA I/O Pin 129
73	GPA1	FPGA I/O Pin 127
74	GPA2	FPGA I/O Pin 125
75	GPA3	FPGA I/O Pin 123
76	GPA4	FPGA I/O Pin 122
77	GPA5	FPGA I/O Pin 121
78	GPA6	FPGA I/O Pin 120
79	GPA7	FPGA I/O Pin 132
115	GPA8	FPGA I/O Pin 134
44	FD0	FPGA I/O Pin 153
45	FD1	FPGA I/O Pin 146
46	FD2	FPGA I/O Pin 142
47	FD3	FPGA I/O Pin 135
54	FD4	FPGA I/O Pin 126
55	FD5	FPGA I/O Pin 119
56	FD6	FPGA I/O Pin 115
57	FD7	FPGA I/O Pin 108
102	FD8	FPGA I/O Pin 165
103	FD9	FPGA I/O Pin 164
104	FD10	FPGA I/O Pin 163
105	FD11	FPGA I/O Pin 162
121	FD12	FPGA I/O Pin 152
122	FD13	FPGA I/O Pin 151
123	FD14	FPGA I/O Pin 150
124	FD15	FPGA I/O Pin 149
28	INT4	FPGA I/O Pin 148
106	INT5#	FPGA I/O Pin 147
32	IFCLK	FPGA GCK1 Pin 77
1	FX2CLK	FPGA GCK0 Pin 80

The FX-2 GPIF can be used to implement data communication between the USB controller and the FPGA.

See chapter “Data path FPGA – PC” for details.

4.4 Memory Interface

FPGA ç è Memory Interface	
SRAM signal name	**FPGA Pin number**
RAM_A0	FPGA I/O Pin 58
RAM_A1	FPGA I/O Pin 60
RAM_A2	FPGA I/O Pin 62
RAM_A3	FPGA I/O Pin 67
RAM_A4	FPGA I/O Pin 69
RAM_A5	FPGA I/O Pin 90
RAM_A6	FPGA I/O Pin 95
RAM_A7	FPGA I/O Pin 97
RAM_A8	FPGA I/O Pin 99
RAM_A9	FPGA I/O Pin 100
RAM_A10	FPGA I/O Pin 98
RAM_A11	FPGA I/O Pin 96
RAM_A12	FPGA I/O Pin 94
RAM_A13	FPGA I/O Pin 89
RAM_A14	FPGA I/O Pin 87
RAM_A15	FPGA I/O Pin 68
RAM_A16	FPGA I/O Pin 63
RAM_A17	FPGA I/O Pin 61
RAM_A18	FPGA I/O Pin 59
RAM_D0	FPGA I/O Pin 74
RAM_D1	FPGA I/O Pin 81
RAM_D2	FPGA I/O Pin 83
RAM_D3	FPGA I/O Pin 86
RAM_D4	FPGA I/O Pin 84
RAM_D5	FPGA I/O Pin 82
RAM_D6	FPGA I/O Pin 75
RAM_D7	FPGA I/O Pin 73
RAM_WE	FPGA I/O Pin 88
RAM_OE	FPGA I/O Pin 70
RAM_CE1	FPGA I/O Pin 71
RAM_CE2	FPGA I/O Pin114

4.5 Unused FPGA Pins

These pins may be accessed via their associated test points. They are not routet to any other target: FPGA TMS Pin 2, FPGA TDO Pin 157, FPGA TDI Pin 159, FPGA TCK Pin 207.

5 Downloading FPGA designs

5.1 Using file format EXO

Before downloading a design, it must be converted to the EXORmacs format using the Xilinx “Prom File Formatter” tool. Choose Type: “byte-wide” and “single Prom”. The resulting *.EXO file can be downloaded to the USB2FPGA evaluation board via USB interface. See the description of XC2USB_DIAG.EXE for details.

5.2 Using file format RBT

The Xilinx FPGA tools do not generate a rawbit “RBT” file by default. There is an option, that must be checked to generate this file.

Open Xilinx ISE Project Navigator and load your design.
Right-click on “Generate Programming File” and select “Properties”
Make sure the box “Create ASCII configuration file” is checked.
Click “OK”.

Important: If design changes are not reflected in the “real world” check if the option to generate an RBT file is checked. If not you are downloading the same old RBT file even if your input design has changed.

5.3 Done Led

When the “DONE” pin of the FPGA is high, LED 2 will light up. This indicates that the FPGA has been configured successfully.

5.4 Resetting FPGA Designs

Signal PE0/T0OUT also known as signal TIMER0 (FPGA I/O Pin 133) can be used as a Reset signal in customer designs. While downloading an FPGA configuration bitstream TIMER0 is held high. After the configuration is done, TIMER0 is made low. It is possible to switch TIMER0 high or low any time after download to re-initialize the design.
In most cases you need to synchronize TIMER0 to the FPGA clock. If you fail to do so, “one-hot” state machines may loose their hot bit or counters may count wrong on the first clock edge after Reset.

6 Sample Designs

The sample designs can be found on the CD-ROM that comes with the USB2FPGA board. Some of them have been synthesized using Xilinx WebPack.
Use “File->Open Project” to load them. If you work with another design system, you will use only the VHDL files and the UCF files of the samples. The files with the extension “UCF” are needed to constrain the I/O Pins to their correct locations. You can also load the precompiled samples without recompiling them to check their functionality.
See “Sample Designs Tutorial” for details.

6.1 Global design hints

Although this document is not an introduction to FPGA design technics there are a few pitfalls that are awful enough to stress you for days.

- Dont use asynchronous logic

Here is one example how asynchronous logic can make your design fail: If you use the output of an comparator as the clock signal of a FlipFlop, every short needle that comes out of the comparator will clock the FlipFlop. You cannot put an capacitor on the output of the comparator as you would try in “real life” hardware. And most probably the design will work as long as you test it, but will fail if you send it out to the customer.
The solution is to convert the design to synchronous logic. Source all FlipFlops with a global clock and use the output of the comparator as “clock enable” of the target-FlipFlop. This way signals inside the FPGA will change state only on (rising) clock edges of the global clock. The time between two consecutive clock edges will allow the signals to travel from the Q outputs of the FlipFlops through combinatorial logic to the D inputs of the target FlipFlops.
In practice, there are almost no reasons to use asynchronous logic.

- Synchronize external signals

If you use external signals that may change levels at any time, it is neccessary to synchronize them first. State machines, for example, that jump to different states depending on unsynchronized external signals may jump to undefined states or loose their one-hot bit.
The easiest way to synchronize external signals is to use an INFF and clock it with the global system clock. A more detailed analysis of the problem shows that sometimes more than one FlipFlop is needed to synchronize external signals due to metastability issues that become relevant at high clock frequenzies. In most cases one single FlipFlop will do the job.

- Always drive input pins

Input pins, that are left floating may cause trouble - even if their state (high or low) is not relevant to the design. This is especially a problem with external buses that have more than one driver. If no driver is active, the signal is undefined and may float to an voltage between high and low. It is a good idea to use pullup, pulldown or weak-keeper in this situation.

- Double-check the pinout

If you use an FPGA I/O Pin as an output and it is also driven from another source, a lot of current may flow and destroy the pins output driver.
Pins without any location constraints are placed by the FPGA design tool at arbitrary locations.

7 Where to get information

7.1 Newsgroups

There are two newsgroups that discuss FPGA and VHDL related themes:

comp.arch.fpga and comp.lang.vhdl

7.2 Links

The list of links on the CESYS homepage www.cesys.com may also be usefull.

7.3 Books

"VHDL Design, Representation and Synthesis"
James R.Armstrong, F.Gail Gray
Prentice Hall, ISBN 0-13-021670-4