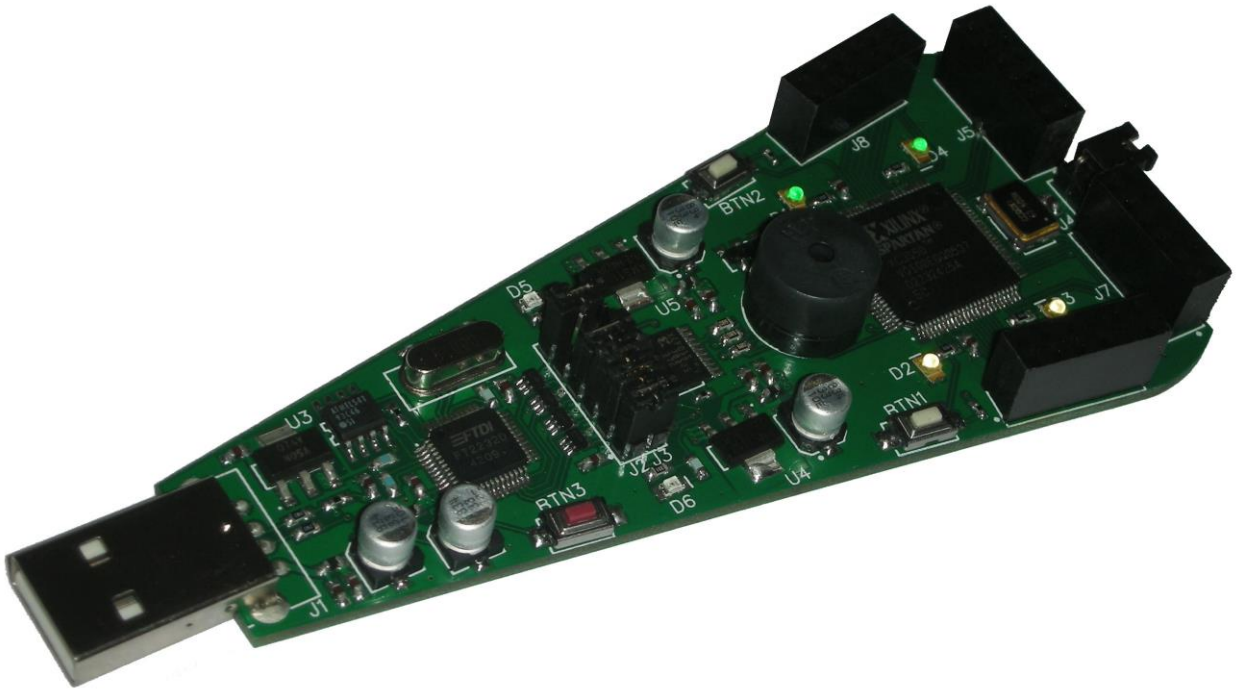
XILINX SPARTAN-3 FPGA KIT USER's GUIDE

MODEL: XK2F3 Revision B



Introduction

Xilinx SPARTAN-3 XK2F3 FPGA kit mainly oriented for FPGA beginners and students. Handy board design allows to overcome problems, which experience beginners, when starting to learn XILINX FPGA programming. This kit doesn't require external power supply or programming cable. Kit is powered from USB. Onboard USB controller allows JTAG programming and UART communication. Kit also has basic peripherals for easy FPGA projects. Experienced users also can find this kit useful, because it can be used as USB programmer, for programming any other external JTAG device, not necessarily XILINX. Documentation includes easy, step by step instructions, and demo projects, for complete beginners.

Features

1) XILINX XC3S50 VQ100 5C FPGA
- 50000 gates (1728 Logic Cells)
- 63 User I/O (42 routed on this board)
- 4x 18x18 hardware multipliers
- 4x 18K-bits block RAMs (72K)
- 2 Digital Clock Managers
- Fully supported by latest XILINX design tools

2) XILINX XCF01S VO20C FLASH
- 1 Mbit Platform Flash PROM
- IEEE Standard 1149.1/1532 Boundary-Scan (JTAG) support for programming
- Endurance of 20000 Program/Erase Cycles
- Fully supported by latest XILINX design tools

3) FTDI FT2232D USB controller
- USB to JTAG port A
- USB to UART port B
- Reset supervision
- Buffered port for programming external JTAG devices. Vref 1.65V -5.5V

4) Onboard IO peripherals
- 4 LEDs
- 2 push buttons
- Speaker
- 62.500 MHz CMOS crystal oscillator (can be converted to 50MHz)

5) Handy configuration
- Configuration from onboard USB programmer
- Configuration from external programmer
- Mode select jumper (JTAG or FLASH)
- DONE LED
- Reset push button

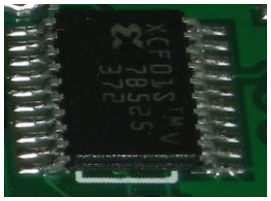
6) Onboard power supply
- Kit is powered from USB
- 5V (USB controller)
- 3.3V (IO, PERIPHERALS)
- 2.5V (VCCAUX)
- 1.2V (CORE VOLTAGE)
- POWER-ON LED

7) 32 independent I/O routed to the connectors
   - Four 2x6 female connectors (8 IO, 2 VCC, 2 GND)
   - Peripheral modules support
   - 2.54mm pitch for all connectors
   - 3.3V IO voltage
8) Small and handy PCB outline

Instructions

1) XILINX XC3S50 is a smallest FPGA, from SPARTAN-3 family. Small, but quite good, if we compare it to any CPLD. This particular board has XC3S50 VQ100 5C FPGA (100 pin package, speed grade 5, commercial temperature range 0°C to +85°C).

- SPARTAN-3 FPGA family datasheet: http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf



2) XILINX XCF01S VO20C is 1 Mbit platform Flash PROM with JTAG interface. There are 1,048,576 configuration bits in XCF01S Flash. XC3S50 FPGA requires 439,264. Demo design will be stored to flash during manufacturing. With this demo, most of the kit functions can be tested.

- Platform flash PROM user guide: http://www.xilinx.com/support/documentation/user_guides/ug161.pdf

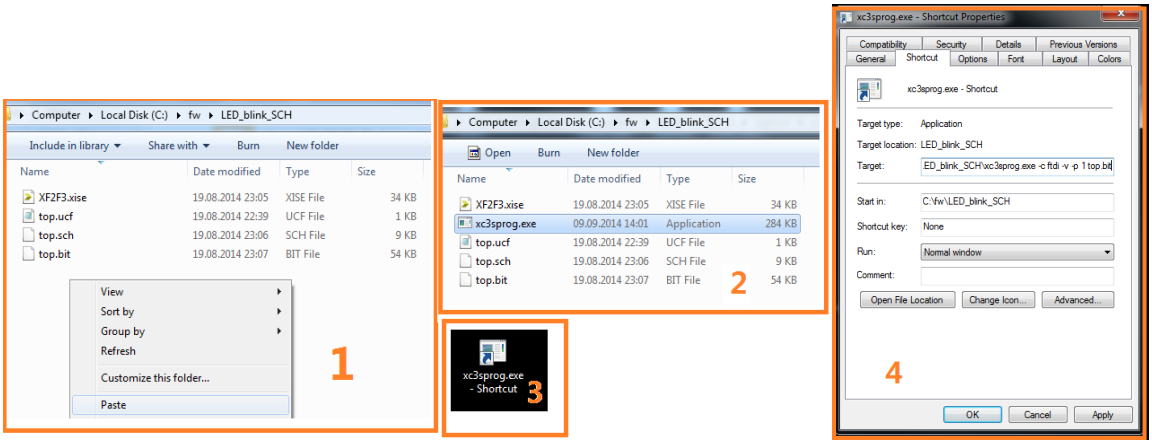
3) XK2F3 FPGA kit has FTDI FT2232D USB controller onboard. FT2232D provides RESET supervision for FPGA IC. PORT B of FT2232D is configured as UART interface. RX and TX signals are connected to FPGA pins 75 and 74. Demo design, stored in FLASH memory, demonstrates simple VHDL UART communication. Use these settings to establish UART connection: 9600-8-n-1. PORT A of FT2232D is configured as JTAG interface. It is connected to buffer IC's.

They are needed to amplify driving current and to widen the supported voltage levels.

Buffer IC's are connected to J2 header. J3 header is connected to JTAG chain of FPGA and FLASH. By default, J2 is connected to J3, with jumpers, so that onboard programmer is connected to boards JTAG chain. If jumpers from J2 and J3 are removed, then, external JTAG devices can be connected to J2, for programming. Programmer can work with Vref from 1.65v to 5.5v. This includes 1.8v, 2.5v, 3.3v and 5v standard signal levels. You can program any JTAG device, not necessarily XILINX. Also, J3 connector can be used, if you want to connect external programmer, and program XK2F3 kit with something else.

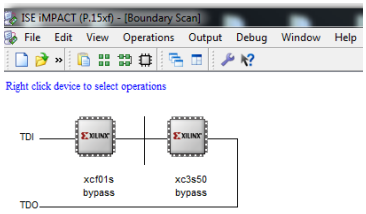
It is recommended to use program “XC3SPROG”, for programming XK2F3 kit.

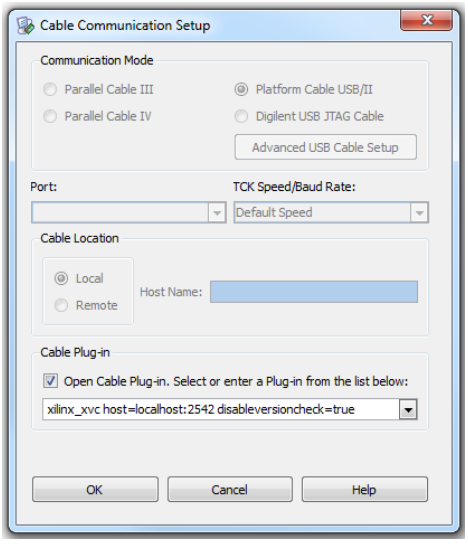
1) Download xc3sprog.exe from https://svn.code.sf.net/p/xc3sprog/code/trunk/xc3sprog.exe
2) Drop xc3sprog.exe to the folder of your ISE project, where .bit file is located.
3) Create shortcut on your desktop "C:\...\xc3sprog.exe -c ftdi -v -p 1 FILE_NAME.bit". Change "1" to "0", if you want to program FLASH instead of FPGA.
4) If something went wrong, or you want to explore more options, start "xc3sprog" from CMD.



Another way, to program XK2F3 kit, is to use XILINX default program “iMPACT”.
FTDI FT2232D USB JTAG programmer officially is not supported by XILINX iMPACT.
There is a trick, how to make it work with iMPACT. We can use FTDI JTAG as virtual cable.
For that, let's use program "mlab_xvcd.exe".

1) Download program from here: http://www.mlab.cz/PermaLink/XVC-SOFTWARE/XVC-1x/BIN
2) Start mlab_xvcd.exe, it should detect XK2F3 FPGA kit right away.
3) Run iMPACT 14.1. Press -> edit -> launch wizard -> configure devices using boundary-scan -> automatically connect.. -> "OK".
4) You should get error that no cable found. Press "ok", and go to Output-> cable setup -> "Open Cable plug-in..." ->
   Paste this : "xilinx_xvc host=localhost:2542 disableversioncheck=true". Press "ok".





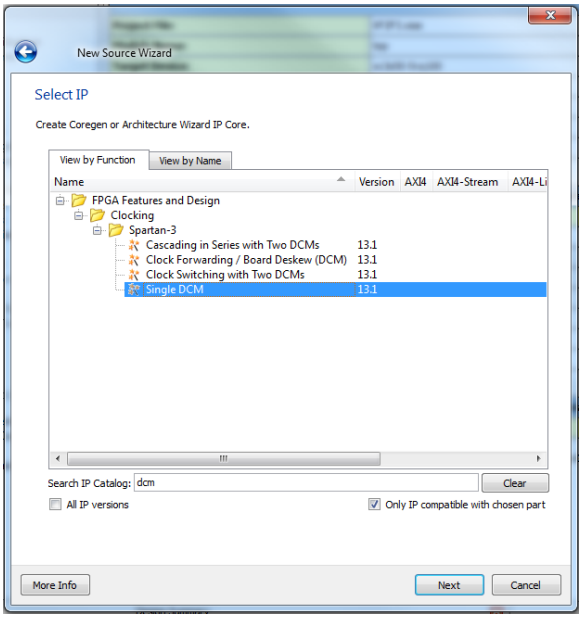
5) You should see now, in the mlab_xvcd window, that connection is established.
6) Press "initialize chain", in iMPACT. You should see your JTAG chain now. Please note, that this method is not officially supported, so something can fail to work properly.

Configuration for FT2232D is stored in 93c46 EEPROM. If something went terribly wrong, and onboard programmer is not working anymore, use FTDI's program "FT_Prog", to restore EEPROM configuration. You can find "FT.xml" file, in zip archive for this kit, it is XK2F3 EEPROM content.

- FT2232D datasheet:
  http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT2232D.pdf
- FTDI's program "FT prog":
  http://www.ftdichip.com/Support/Utilities/FT_Prog_v2.8.2.0.zip
- More info about FT2232D:
  http://www.ftdichip.com/Products/ICs/FT2232D.htm
- More info about XC3SPROG:
  http://xc3sprog.sourceforge.net/
- More info about mlab_xvcd:
  http://www.mlab.cz/

4) Board has few simple peripherals onboard: 4 yellow LEDs (p71,p5,p21,p55), 2 momentary active-low push buttons with pull up resistors (p4,p65), 16 ohm passive speaker (p72). XK2F3 kit is equipped with 62.500 MHz CMOS crystal oscillator. Your project may require frequency other than 62.5 MHz. Here are two ways, how 62.5MHz can be converted inside of FPGA:

First method is to use FPGA digital clock manager (DCM). This method allows dividing and multiplying of clock frequency.



1) Open ISE 14.1. Right click in "Hierarchy" window -> "New Source"
2) Select "IP", type file name, and press "Next"
3) In the search field type "dcm", find "Single DCM", select it, press "next", press "finish"
4) In next window, don't change any settings and press "OK"
5) In the field "input clock frequency" type "62.5".
6) Uncheck "RST" and "locked", check "CLKFX", press "next".
7) Don't change any settings and press "next".
8) Type desired output frequency and press "next", press "finish"
9) In "Hierarchy" window, left click on your DCM name.
10) Now, to use your DCM, double click "View HDL Instantiation Template". Template will be generated. Copy - paste it to your project.
11) If your project is based on schematic top module, then skip step 10, and double click on "Create Schematic Symbol", in the "Process" window. DCM schematic symbol will appear in schematic symbol library.

Another way, to convert clock frequency, is to use custom counter. This method only allows dividing of clock frequency:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
entity clkdiv is
Port ( clk : in STD_LOGIC;
rst : in STD_LOGIC;
clkout : out STD_LOGIC);
end clkdiv;
architecture Behavioral of clkdiv is
constant cntendval : STD_LOGIC_VECTOR(25 downto 0) := "11101110011010110010100000";
signal cntval : STD_LOGIC_VECTOR (25 downto 0);
begin
process (clk, rst)
begin
if rst = '1' then cntval <= "00000000000000000000000000";
elsif (clk'event and clk='1') then
if (cntval = cntendval) then cntval <="00000000000000000000000000";
else cntval <= cntval + 1;
end if;
end if;
end process;
clkout <= cntval(25);
end Behavioral;
```

Change "constant cntendval" to value on which clk should be divided. Constant in this example is set to 62500000, to get 1Hz on output. Use DEC to BIN converter, to get your value.
Use this template, to fit above example in your VHDL top module:

```
component clkdiv
port (   clk              : in  std_logic;
         rst              : in  std_logic;
         clkout           : out std_logic);
END component;
begin
         clkd: clkdiv
         port map(clk => clk,
                  rst => '0',
                  clkout  => spkr);
```

Also, remember, that additional clock signals can be connected from outside of the board.

- More info about DCM:
  http://www.xilinx.com/support/documentation/application_notes/xapp462.pdf

5) Reset button BTN3 is connected to FPGA PROG_B signal, when it is preset – RESET is active.

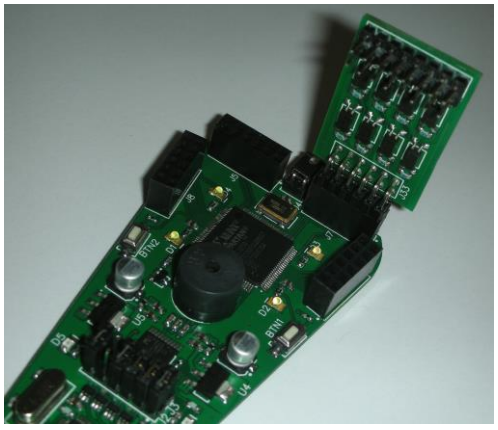
As described in step 3, of this guide, XK2F3 kit can be programmed by onboard USB programmer or by external JTAG programmer. D6 is FPGA DONE LED. It turns on after FPGA has finished configuration process. J4 is FPGA mode select jumper.

When jumper is connected, all tree FPGA mode pins are tied to GND. This is Master Serial mode (loading from flash). When Jumper is removed, M0 and M2 pins are tied to VCC AUX by internal FPGA resistors, this is JTAG mode (waiting configuration from JTAG, not loading from the flash).

- More information about configuration modes, SPARTAN-3 datasheet, page 118: http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf

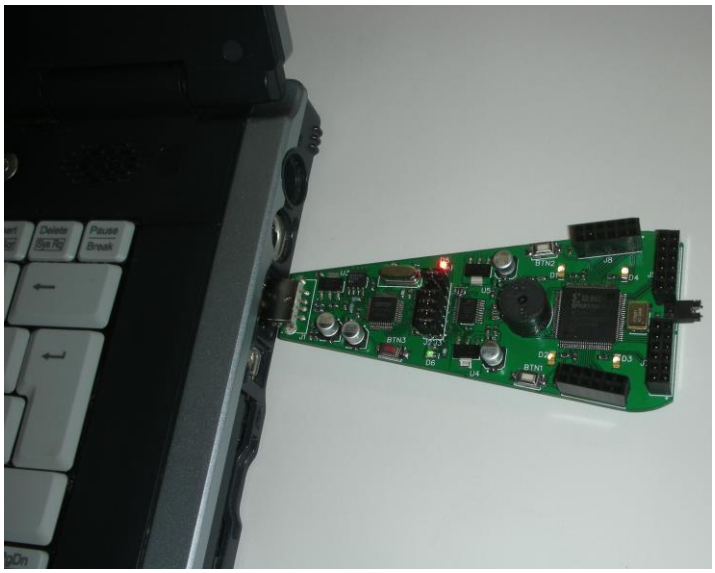
6) XK2F3 kit is designed to receive power from USB port. Kit has onboard power regulators and filters, to serve all needs. Filtered 5V is used by USB controller and EEPROM. 3.3V is used by FPGA IO, by onboard peripherals, and also supplied to IO connectors. 2.5V is used as FPGA VCCAUX and JTAG voltage. 1.2V is used as FPGA core voltage. D5 is a POWER-ON LED, when it lit - it means that there is input DC voltage connected.

7) There are total 32 I/O pins, what are routed to the connectors, and can be used in your design. Please refer to the board schematics, to get information about connections. All I/O are independent and not crossing with any kit peripherals. J5, J6, J7 and J8 are 6x2 female connectors. Each connector has 8 IO, 2 VCC 3.3V and 2 GND. These female connectors are designed specially to support peripheral modules. Peripheral modules, with compatible male connectors, can be ordered from our web site, or some other suppliers. Please note that all FPGA banks are powered from 3.3V, it means that you must use I/O standard “LVCMOS33” or “LVTTL” in your designs, to achieve better results. Please note that HSWAP_EN pin is connected to VCCAUX, this means that when FPGA is not configured, all IO pull-up resistors are disabled.



When demo design is loaded to FPGA, IO connectors can be tested. Use fly wire, to connect IO pins one by one to VCC 3.3V. Test is passed if speaker beeps, every time when one of IO pins is connected to VCC. If there is missing contact or short circuit between IO, speaker will not beep. Be careful, and don’t make short circuit between VCC and GND.

8) Board size is small and handy. It is designed to be plugged directly in your computers USB port. If you are using desktop pc, and your USB port is far from your desk, you can use USB extender cable, or external USB hub.



Quick start guide, for the first time user

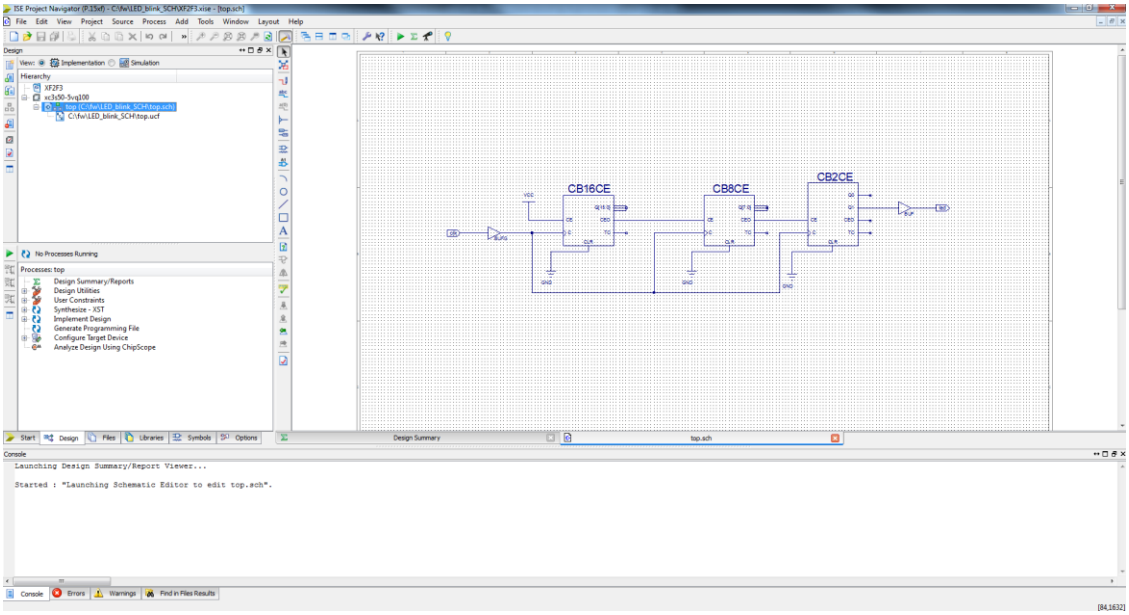
1) Download and install Xilinx ISE (http://www.xilinx.com/support/download.html). Register, and get free license. Step by step instructions, in this guide, are referenced to ISE 14.1. If your version of ISE is different, some minor mismatches can be found. Also, demo projects for this kit, are created in ISE 14.1. It should be possible to open them in newer ISE version.

2) Plug into USB port your XK2F3 kit. Windows should find and install driver automatically, without any problems. If this did not happen, or you are not using windows, then, download drivers for FT2232D chip, from FTDI website (http://www.ftdichip.com/Products/ICs/FT2232D.htm). During driver installation, please note the COM port number, what was assigned to the kit. If you have missed that information, please find this information in computers device manager.

3) Now, first of all, let's test that our hardware is working as expected. After you plugged in your XK2F3 kit, red LED D5 will turn on, indicating presence of supply voltage. If you did not removed jumper, from J4 connector, then, demo design, what is stored in the flash memory, will be loaded to FPGA. Green LED D6 will turn on, indicating that FPGA configuration is completed. You can notice barely visible delay between D5 and D6. This delay is time, what is needed for FPGA to load bitstream from flash. This delay is also extended by RESET delay, from FTDI chip. If you will remove jumper from J4 connector, and plug in XK2F3 to USB, nothing, except red LED, will lit. Now, when kit is plugged to USB without J4 jumper, place J4 jumper, without removing kit from USB. Nothing happens. Yes, because FPGA samples mode pins only when reset signal is active. Press BTN3 red button, and bitstream from flash will be loaded to FPGA. You can also press reset button again, to reload flash bitstream, if it was updated, or just for fun. FPGA remembers configuration only till next reset event. It is automatically updated, if you send another configuration, to configured FPGA. When factory demo design is loaded to FPGA, speaker will start beeping. And yellow IO LEDs will start blinking. Circling in one direction. Press BTN1, beeping will become louder, and LEDs will start to circle in another direction. Press BTN2, and they will return to previous state. Please read step 7, of "instructions" guide, to get instructions, how to test IO connectors.

4) Let's test UART. For that, as mentioned before, we need to know what COM port number was assigned to XK2F3 kit. Also, we need a special program, UART terminal. If you are still using WINDOWS XP, then, just start "hyper terminal", located in your "Accessories" program folder. In most other cases, unless you already have it, you will need to download, from somewhere, UART terminal program. We recommend to use program "TeraTerm". Open your UART terminal, and establish connection to COM port, what was assigned to XK2F3 kit. Set terminal settings to 9600-8-n-1. If you can't select assigned port number, in the UART terminal, go to device manager, and change assigned port number to something, what you have in UART terminal, even if it says that port is in use. Please note that XK2F3 kit must be plugged in USB, before you make attempt to establish UART connection. Demo project is designed so, that UART received data is transmitted back as echo. Received data is monitored, XK2F3 kit will respond with unique sound for symbols 0,1,2,3,4,5,6,7,8 and 9. XK2F3 kit will not make any sound for other symbols, but you can change it, by editing VHDL code.

5) Now, let's finally start to learn FPGA programming. I assume that you have installed ISE, by now. Zip archive, for this kit, has folder "fw". In that folder, there are three folders with FPGA projects, and one file, "FT.xml". XML file consist configuration data for XK2F3 EEPROM. Normally, you don't need it. Folder "demo" has demo design, what is stored to XK2F3 flash. It is easy project, but still, can be difficult for first time user. So let's start from other two projects. These two projects are alike, but programmed differently. There are three ways, how to make a code, for XILINX FPGA. First way is to make schematics. Yes, you can just connect some logic components in visual editor, and program this scheme to FPGA. Second way is to use VHDL language, to describe a logic scheme, instead of drawing it. And third way is to use Verilog language, to describe a logic scheme. Folder "LED_blink_SCH" has project with scheme, for blinking a led. Folder "LED_blink_VHDL" has project with VHDL code, for blinking a led. Verilog language will be not used in this guide, but yes, you can use Verilog with XK2F3 kit. Let's start with schematics. Go to "LED_blink_SCH" folder and open "XF2F3.xise". ISE project navigator will be opened. In the "Hierarchy" window, you can see files of the project. Below "Hierarchy" window, is "process" window. In "process" window, you can see what you can do with files from "Hierarchy" window. Double click on the file "top.sch" in "Hierarchy" window.



This will open file for editing. You can also edit file "top.ucf". This file describes, to what pins our schematic will be connected. When you have finished editing, save changes, go to "process" window, right click on "Generate Programming File", and select "run". Two previous steps, "synthesize" and "implement design" will be run automatically. If there were no critical errors, then, you can find your generated programming file in "LED_blink_SCH" folder. In this example, it will have name "top.bit". Please read step 3 of "instructions" guide, to get information about how to program your BIT file in to XK2F3 kit. Follow same steps, when working with "LED_blink_VHDL" project. Instead of top.sch, open top.vhd, for editing VHDL code. When you are ready to try something else, besides LED blinking, you can check project "demo". When writing VHDL or Verilog code, always remember, that those languages describe electronic schematics. Know that there are restrictions for file names. Also, you can get error, if your folder path is too long, or it has spaces, or non latin symbols. If you have background for programming microcontrollers, then, might be good news for you, that it is possible to fit soft microcontroller, like "PicoBlaze", to FPGA. You can get more information about FPGA programming in books, educational videos, forums, and your university teacher.