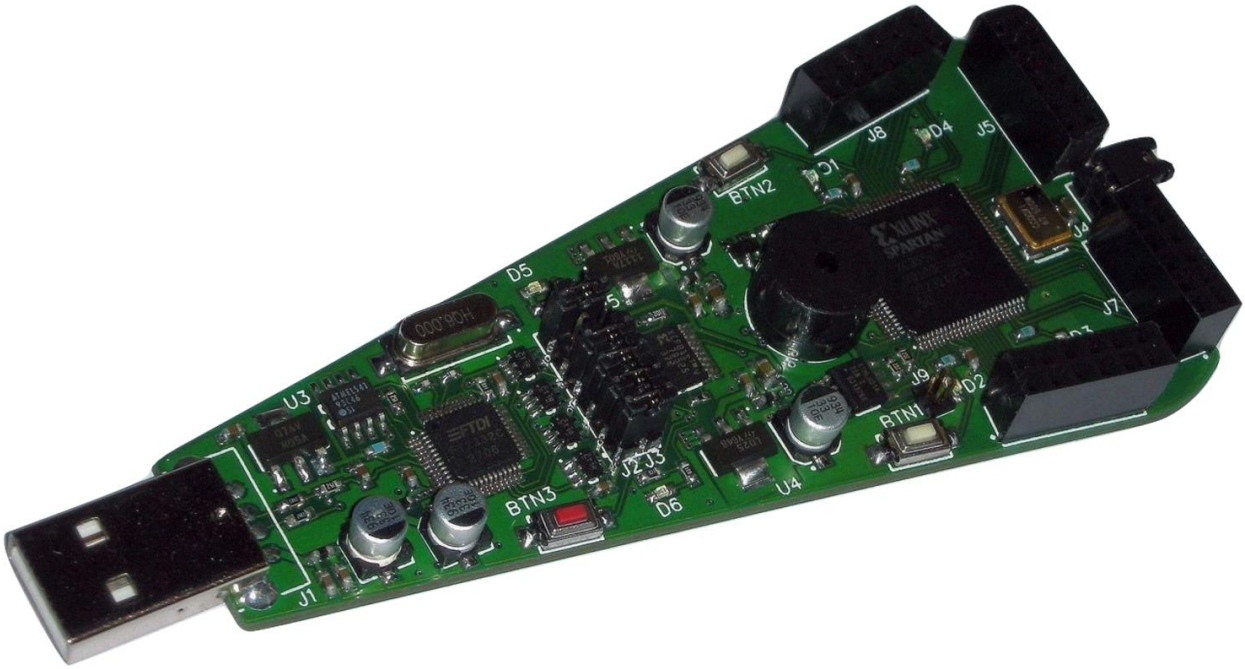
XILINX SPARTAN-3 FPGA KIT USER's GUIDE

MODEL: XK2F3-2 Revision C



Introduction

Xilinx SPARTAN-3 XK2F3-2 FPGA kit mainly oriented for FPGA beginners and students. Handy board design allows to overcome problems, what have 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. Kit also can be used as USB programmer, for programming any other external JTAG device. Documentation includes step by step instructions, and demo project.

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

2) XILINX XCF02S VO20C FLASH
- 2 Mbit Platform Flash PROM
- IEEE Standard 1149.1/1532 Boundary-Scan (JTAG) support for programming
- Endurance of 20000 Program/Erase Cycles

3) FTDI FT2232 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)
- 1024Kbit EEPROM

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 IO connectors
- 3.3V IO voltage
- I²C header

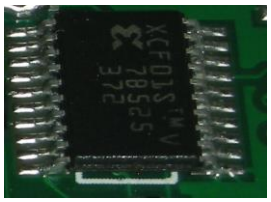
8) Small and handy PCB layout

Instructions

1) The Spartan-3 family is a alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.This 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 XCF02S VO20C is 2 Mbit platform Flash PROM with JTAG interface. There are 2,097,152 configuration bits in XCF02S Flash. XC3S50 FPGA requires 439,264. Demo project, for testing this board, will be stored in flash.



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

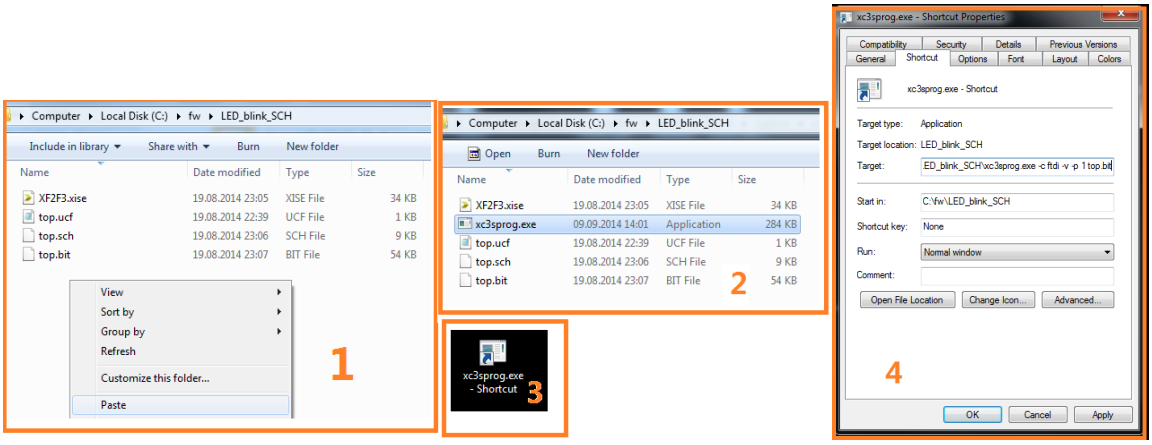
3) XK2F3-2 FPGA kit has FTDI FT2232 USB controller onboard. FT2232 provides RESET supervision for FPGA IC. PORT B of FT2232 is configured as UART interface. RX and TX signals are connected to FPGA pins 75 and 74. Demo design, stored in FLASH memory, demonstrates VHDL UART communication. Use these settings to establish UART connection: 115200-8-n-1. PORT A of FT2232 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.

Other side of buffer IC's is 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 program XK2F3-2 kit with external programmer.

It is recommended to use program “XC3SPROG”, for programming XK2F3-2 kit.

1) Download xc3sprog.exe from https://svn.code.sf.net/p/xc3sprog/code/trunk/xc3sprog.exe
2) Put xc3sprog.exe to the folder of your ISE project, where .bit file is located.
3) Create shortcut on 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 you want to explore more options, start "xc3sprog" from CMD.



Configuration for FT2232 is stored in 93c46 EEPROM. If needed, use FTDI’s program “FT_Prog”, and “xk2f3-2_93c46.xml” file, to restore EEPROM configuration.

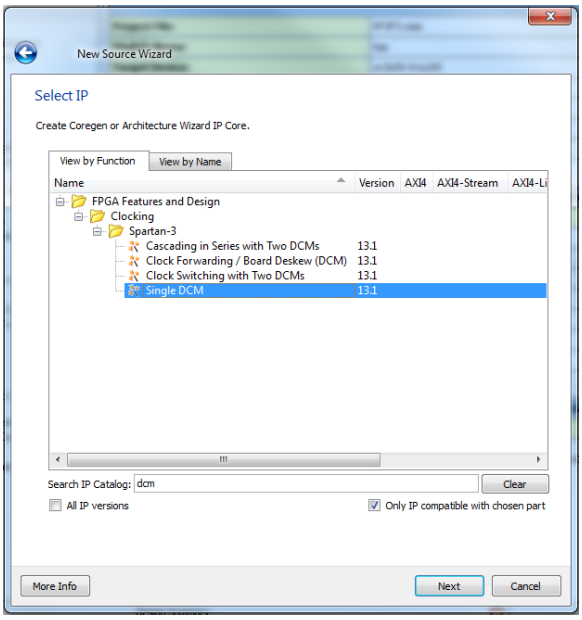
.

- FT2232 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 XC3SPROG: http://xc3sprog.sourceforge.net/

4) Board has few simple peripherals onboard: 4 yellow LEDs (p71,p5,p21,p55), 2 momentary active-low push buttons (p4,p65), 16 ohm passive speaker (p72), and 1024Kbit I²C EEPROM. All these peripherals can be tested when VHDL demo project is loaded to FPGA. VHDL demo project includes “reprogramming mode”. In this mode, EEPROM can be reprogrammed. To program AT24C1024, connect external programmer to J9. To restore EEPROM factory settings, use “xk2f3-2_24c1024.bin”. J9 connector also can be used to connect logic analyzer, to debug your projects.

XK2F3-2 kit is equipped with 62.500 MHz CMOS crystal oscillator. Your project may require different frequency. Here are two ways, how 62.5MHz frequency 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 14.7. 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 insert above example into 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, when needed, 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

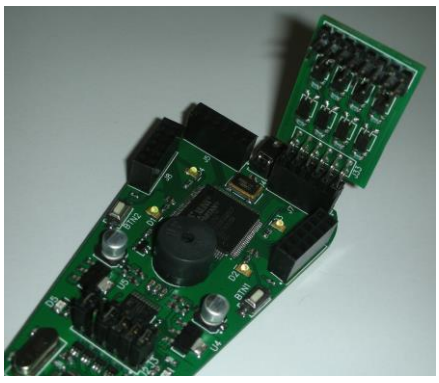
5) Reset button BTN3 is connected to FPGA PROG_B signal, when it is pressed – RESET is active.

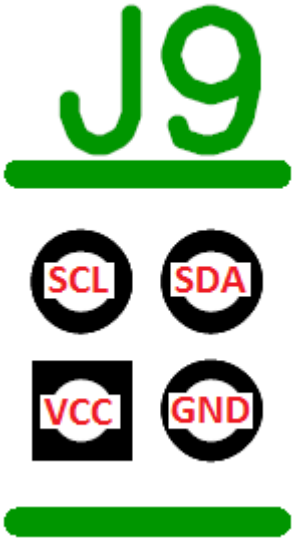
As described in step 3, of this guide, XK2F3-2 kit can be programmed by onboard USB programmer or by external JTAG programmer. D6 is FPGA DONE LED. It turns on after FPGA has finished configuration process. J4 is FPGA mode select jumper.

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).

6) XK2F3-2 kit is designed to receive power from USB port. Board has onboard power regulators and filters. Filtered 5V is used by USB controller and 93c46 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 as 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 from 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. 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.





J9 is I²C header, connected in parallel with I²C bus.
J9 can be used to reprogram AT24C1024 EEPROM, or to connect logic analyzer. J9 has 1.27mm pitch.

8) Board is small and handy.
It is designed to be inserted directly into 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.

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 14.7. If your version of ISE is different, some minor mismatches can be found.
   Also, demo project for this kit, is created in ISE 14.7.

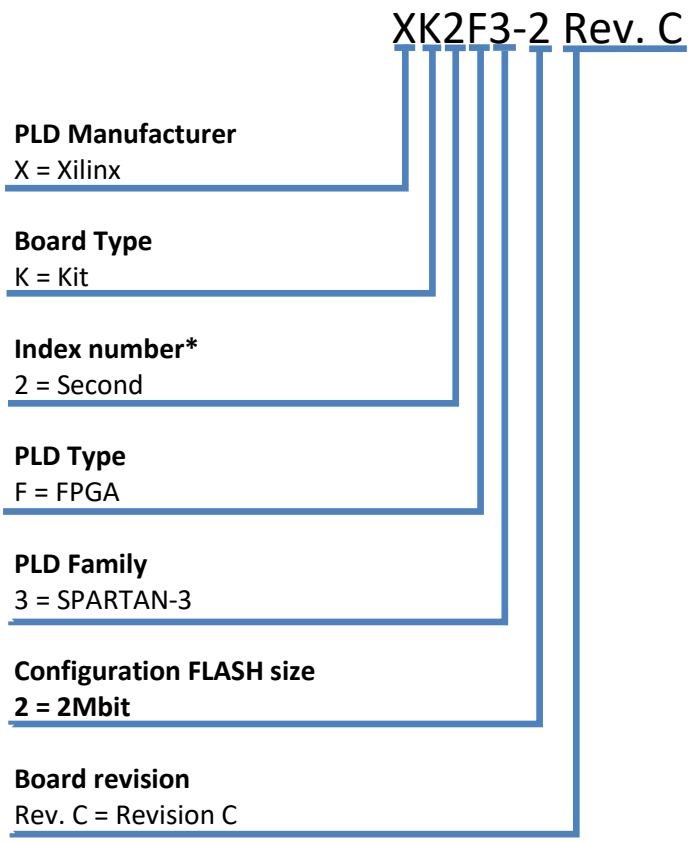
2) Plug into USB port XK2F3-2 kit. Windows should find and install driver automatically. If this did not happened, or your system is not windows, then, download drivers for FT2232 chip, from FTDI website (http://www.ftdichip.com/Products/ICs/FT2232D.htm). During driver installation, please note the COM port number, what was assigned to the kit. If you have missed that information, please find this information in computers device manager.

3) After you plugged in your XK2F3-2 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 XK2F3-2 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. FPGA remembers configuration only until next reset event. It is automatically updated, if you send another configuration, to configured FPGA. If you will reprogram flash, but won't initiate reset event, FPGA will still hold old configuration. When factory demo design is loaded to FPGA, yellow IO LEDs will be blinking.

4) Let's make UART connection. For that, as mentioned before, we need to know what COM port number was assigned to XK2F3-2 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, you will need to download, UART terminal program from internet. We recommend to use program "TeraTerm". Open your UART terminal, and establish connection to XK2F3-2. Set terminal settings to 115200-8-n-1. XK2F3-2 will send welcome note to UART, if any key, except "1","2" or "3" is pressed.
   If button "1" is pressed, XK2F3-2 will perform quick speaker test.
   If button "2" is pressed, XK2F3-2 will enter reprogramming mode. In this mode, AT24C1024 EEPROM can be reprogrammed. Connect external programmer to J9 connector.
   If button "3" is pressed, XK2F3-2 will enter IO test mode. J5, J6, J7, J8 and BTN2 can be tested. Press BTN2 or connect IO pins to VCC 3.3V one by one. If one IO pin is connected to VCC, or BTN2 is pressed, then all LEDs will turn ON. If none or several pins connected at the same time, then LEDs will stay OFF. When LEDs turn on, beep sound is produced.

5) Zip archive, for this kit, has demo design, what is stored to XK2F3-2 flash. Go to “demo” 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 these files. Double click on the file “top.vhd” in “Hierarchy” window. This will open file for editing. You can also edit file “top.ucf”. This file describes, to what pins our system 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 “demo” folder. In this example, it will have name “top.bit”. Please read page number 4 of this guide, to get information about how to program your BIT file into XK2F3-2 kit.

Model name chart



*Index number is used to distinguish from similar existing model