3ème conférence Internationale des énergies renouvelables CIER-2015
Proceedings of Engineering and Technology - PET
pp.70-75
Copyright IPCO-2016

Spartan-3E FPGA Design Guide for prototyping and production environment

Mohammed BOUDJEMA[#1], Latifa HAMAMI-MITICHE[*2]

Signal and Communications Laboratory, National polytechnic school
Algiers, Algeria
[1]Boudjema.m@gmail.com
[2]latifa.hamami@enp.edu.dz

***Abstract*—This work describes the design process from beginning to end for a multi-purpose printed circuit board (PCB) based on the FPGA (Field-Programmable Gate Arrays) Spartan-3E[1] . It will present critical items along the way, and describe how these items were able to define the design steps.**

***Keywords*— FPGA; spartan-3E; PCB; master serial; JTAG; SRAM; IC**

I. Introduction

Many students develop FPGA (Field-Programmable Gate Arrays) applications in various domains (Image processing, signal processing, renewable energy…) using FPGA development board, but after confirming their application using development board they don't know how to implement this application in customized printed circuit board using FPGA IC (Integrated Circuit). The goal of this work is to create a "reference manual" for students to use as a guide when they need to realize FPGA application with customized PCB. This guide is based on Xilinx FPGA Spartan-3E, but designers of the other FPGA types can also follow this design flow, by respecting design requirement of their FPGA manufacturer and by using its specific design tools, this is because each FPGA manufacturer has its design requirements and specific tools, design flow of FPGA-based printed circuit board such as schematic and PCB design are the same.

II. System Specification

Once you have decided to design with FPGA, there are factors which the designer should take into consideration before beginning the design process.

The first one is selecting functions dedicated to FPGA; the designer must analyze which functions should be implemented in the FPGA and which will be done with other components. This will allow the designer to have an idea about the specifications of the FPGA to be selected for the application and determine the other components needed.

The second factor is the specification for the PCB that you are planning to build; the designer must have an idea about physical board constraints, and the cost dedicated to this project.

Along the way these parameter will affect many decisions such as selection the FPGA device, selection of design tools, number of PCB layer…

III. FPGA Selection

FPGA selection is the key of the successful design; a good design will start by the selection of the appropriate and the optimum FPGA device.

As announced before this design guide is based on Spartan-3E Xilinx FPGA, but in this section we will briefly give criteria to be considered to choose the optimal FPGA device for the design:

A. FPGA Programming Technologies

The main criterion is the FPGA Programming technology, (We will introduce with more detail since it is very critical and must be understood when designing with FPGA) it will determine the next steps and the big lines of the design such the tools and annex components.

Three different major technologies are in use today for programming FPGAs: SRAM, FLASH EPROM and antifuse [1]:

- SRAM-based Technology [2]: The first one uses RAM cells to store the configuration information. The main advantage of this approach is that an FPGA can be programmed after the chip has been assembled into a system, without the need for any separate handling during manufacture. Furthermore, the system can be upgraded after delivery by storing new configuration information, rather than having to replace chips or other hardware. If the configuration is stored using volatile SRAM cells, it needs to be loaded each time power is applied to the system. Hence, the configuration needs to be stored in a separate nonvolatile memory, and additional circuits need to be included in the system to manage loading the configuration. The two main FPGA vendors, Xilinx and Altera, both use SRAM cells for their devices and provide specialized flash RAM devices for storing and configuring the FPGAs.
- EPROM-based Technology: Other vendors, such as Actel, provide FPGAs that use nonvolatile flash RAM cells for the configuration information. Such devices

[1] Spartan-3E is Xilinx FPGA

ISSN: 2356-5608

do not need the external components for storing or loading the configuration, thus reducing overall system complexity. This second form called EPROM- or FLASH-based FPGA.

- Anti-fuses Technology [2]: The third main form of FPGA uses anti-fuses to configure the device, as its name suggests, is a conductive connection that is formed during programming, as opposed to being blown. Since programming is done by forming a connection, no storage is needed, either inside the FPGA or externally. Moreover, the device is less susceptible to soft errors due to radiation. However, the device must be programmed separately before being installed in the final system. This requires additional manufacturing steps and handling, adding cost to the manufacturing process.

The table I briefly summarizes the key points associated with the various programming technologies described above [1]:

TABLE I. SUMMARY OF PROGRAMMING TECHNOLOGIES

Feature	**FPGA Programming Technology**		
	SRAM	***Antifuse***	***Flash***
Reprogramming?	Yes (in-system)	No	Yes (in-system or offline
Reprogramming speed (including erasure)	Fast	Not applicable	3 x SRAM
Volatile ?	Yes	No	No (but can if required)
Extenal configuration file ?	Yes	No	No
Good for prototyping ?	Yes	No	Yes
Instant-on ?	No	Yes	Yes
IP security	Poor	Very good	Very good
Size of configuration cell	Large (six transistors)	Very small	Small (two transistors)
Power consumption	High	Low	Medium
Radiation hardness ?	No	Yes	No

B. The number of CLBs (Configurable logic blocks) in the device

This will determine how much logic the device can hold and how easily your design will fit into it. Although most FPGAs have similar logic blocks, there are differences, for example, in the number of flip-flops and the width of the lookup tables. Try to find a CLB architecture that fits your design. If your design has wide combinatorial functions, choose an FPGA using CLBs with large numbers of inputs. If your design has many pipelined stages, you will prefer CLBs with several flip-flops.

C. The number and type of I/O (Input/Output) pins

Obviously, the FPGA will need to support the number of I/O pins in your design. Also, determine how many of these are general-purpose I/O and how many are reserved for special functions such as clock input, master reset, etc.

D. The number of clock input pins

Clock signals can be driven only into particular pins. If your design has several clock domains (i.e., sections driven by separate clocks), you will need an FPGA that has that many clock input pins.

E. Embedded devices

Does your design interface with devices such as a microcontroller or a PLL? Many FPGAs now incorporate specialized functions like these, which will make your job much easier and allow you to integrate more devices into a single FPGA.

Those are the most important criteria that designer should take in consideration when choosing an FPGA.

Good understanding of the FPGA architecture and all available features will allow the designer to take in consideration all criteria, and thereafter a best selection. Other criteria and more details can be found in [3].

The Spartan-3E is Xilinx SRAM-Based FPGA. It is Logic Optimized; destined for applications where logic densities matter more than I/O count and Ideal for logic integration, DSP co-processing and embedded control which require significant processing and narrow or few interfaces. This model contains 500,000 gates, 158 input/outputs (65 differential), 1164 logic blocks, and a 2.7ns delay time [4].

IV. CONFIGURATION AND PROGRAMMING MODE

At this stage we should select the configuration and/or the programming mode for the selected FPGA.

The process of downloading configuration data into an FPGA using an external data source, such as a PROM is the Configuration. Programming is the process of loading the configuration data or program data into a PROM.

For selecting the configuration or the programming method we need to consider whether the system being developed is for a prototyping environment, a production environment, or both. A prototyping environment requires easy access to the device for multiple reconfigurations. For a production environment, reconfiguration is not as high a priority as fast programming times and robust configuration methods that retain configuration data.

For Spartan-3E, the production environment will be achieved by Master-Serial Mode and the prototyping environment with JTAG mode. The combination of the Master-Serial mode with the Boundary-Scan (JTAG) mode gives us a versatile setup which allows for easy debug and multiple configuration methods, Master Serial mode provides a simple and robust configuration mode for production, and the Boundary-Scan (JTAG) mode provides in-system programming support for flexibility when configuring a chain in the prototyping stage.

A. *Master-Serial Mode*

The Master-Serial mode is the simplest configuration method for FPGAs. The FPGA loads configuration data from a serial PROM. Using the FPGA to provide the clock, it virtually loads itself and utilizes its internal oscillator, which drives the configuration clock. The FPGA provides all the control logic. In this mode, data is loaded at one bit per CCLK [5].

B. *JTAG or Boundary-Scan Mode*

JTAG or Boundary-Scan mode is an industry standard (IEEE 1149.1, or 1532) serial programming mode. External logic from a cable, microprocessor, or other device is used to drive the JTAG specific pins, Test Data In (TDI), Test Mode Select (TMS), and Test Clock (TCK). This mode has gained popularity due to its standardization and ability to program FPGAs, and PROMs through the same four JTAG pins. The data in this mode is loaded at one bit per TCK [5].

C. *Xilinx PROM*

There are many different kinds of PROM chips. PROM stands for Programmable Read-Only Memory. Because this type of memory is meant to be read quickly on a regular basis, writes typically take much longer than reads. PROM chips exist for many different types of devices. Fortunately, Xilinx makes a PROM chip that is designed to work specifically with many members of the Xilinx FPGA line. For this application, the Xilinx XCF04S was chosen. The XCF04S (04: it means 4MB) features programmability through the JTAG interface. This will allow us to arrange the PROM and the FPGA in a JTAG chain such that we will be able to program the devices independently through a single interface.

There are three general steps necessary to configure or program a Xilinx programmable logic device described below.

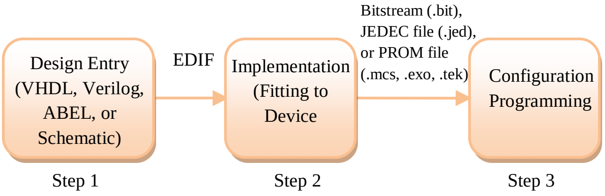
Design Entry (VHDL, Verilog, ABEL, or Schematic)
EDIF
Implementation (Fitting to Device
Bitstream (.bit), JEDEC file (.jed), or PROM file (.mcs, .exo, .tek)
Configuration Programming
Step 1
Step 2
Step 3

Fig. 1. Flow Overview to Configuration and Programming Xilinx Devices [5].

Software design entry tools are used to create a design in VHDL, Verilog, ABEL, or Schematic. Software implementation tools are used to fit the design netlist into the desired Xilinx architecture and produce a configuration bitstream or JEDEC file.

Xilinx are developed these software and include in one software package, which will be used in our case; Xilinx ISE (Integrated Software Environment).

D. *Hardware configuration*

The hardware connections to implement the Master Serial/Boundary-Scan (JTAG) Mode Combination are shown in “Fig. 2” [6].

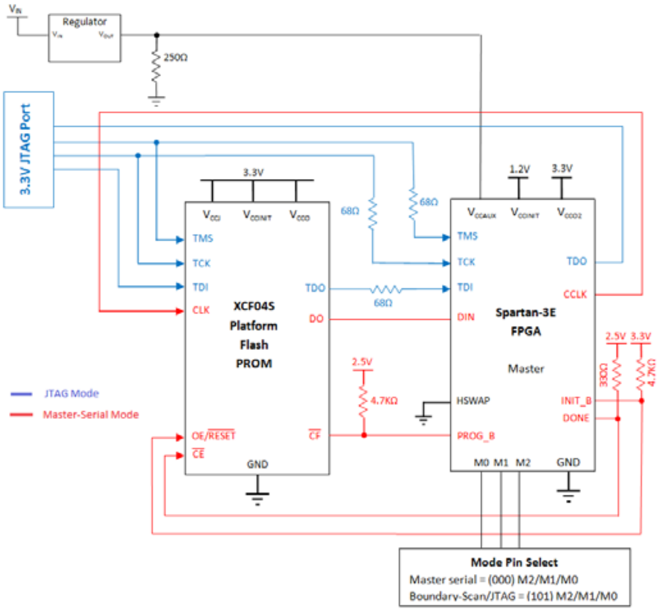

Fig. 2. Hardware connection for Master Serial/Boundary-Scan (JTAG) Mode combination.

V. SCHEMATICS AND PCB

When the configuration mode is established, our schematic editing can be started, for this we need to add other parts of the system (Table II Show the basic board requirements other than FPGA and PROM), we must have a complete and accurate schematic diagram before beginning to layout the PCB.

TABLE II. BASIC BOARD REQUIREMENTS.

Required Component	***Details***
Clock	50 MHz speed
DDR-SDRAM	about 64 MB for memory
Serial port	DTE (DCE optional)
Analog-Digital conversion	at least 2 analog inputs
Digital-Analog conversion	at least 2 analog outputs
Switches/ Buttons/ LEDs	assorted I/O for physical interaction with user

Almost modern PCB software tools provide a host of tools that allow designers to edit schematics in order to produce PCB. This allows editing the schematic with the PCB design in mind to avoid mistakes when translating. A simple and free tool “PCB Artist” will allow that with replying to our PCB design requirements [7].

A. *Design Considerations for schematic:*

We start schematic creation process by determining how the interfaces between the FPGA and the other components will be implemented (IO standards, clocking requirements, timing specifications, etc...). For this we need to refer to data sheet for each component.

After, we should determine the power supplies for the board (for both the FPGA and the components). The data sheets for the FPGA and components will tell us what voltages and decoupling are required. For the FPGA, the most important consideration is VCCO banking rules since every bank is powered individually and must be configured to operate at the voltage it is powered at. We will use the XPE (Xilinx Power Estimator) tool to determine power supply sizing for the FPGA and then we will add 50-100% for margin.

The schematic creation shall be start with the most complex components and interfaces and leave the simple easy items (switches and LEDs) for the end.

If the schematic is neat, logical and clearly laid out, then it really does make your PCB design job a lot easier.

B. *Design Considerations for PCB Layout [8]:*

Component selection is perhaps one of the most important steps in the PCB creation process. The primary thing to look for in a part is its ability to satisfy or exceed the requirements originally established. The second thing is its size. Small sized parts are great when it is necessary to fit many parts onto a small board. However, this may not be ideal when considering how the board is to be assembled. For example, 0805 sized part for Resistor and capacitor will be used since will be soldered by hand[2].

Spartan-3E chip (XC3S500E) comes in several different sizes and packages. Some of these have more input/output pins than others, mostly because of size constraints. For the size and scale of this particular process, the PQ208 package was chosen. This chip is 28mm by 28mm, and has 208 pins with a 0.5mm pitch [4]. These specifications are suitable for this project because they allow for an acceptable number of input/outputs while maintaining the ability to be soldered by hand. Same consideration for XCF04S (Xilinx's PROM) that comes in a VO20 package which is easily hand-soldered.

Once all of the parts have been selected[3], there are factors which a designer should take into consideration before proceeding. The most important of these is the pin mapping of the FPGA especially and other components, because once a PCB is fabricated, it is very difficult to re-route connections which are incompatible with the pins they are attached to. A pin planning tool of PlanAhead Xilinx software package will be used for.

In addition to pin planning, physical board constraints must be decided; the size, number of layers which will define the cost. Designer can adjust these constraints according to project cost and design requirements.

For this project the board will be 6" by 5", and 6 layers. The table below shows an example for allocation of layers.

TABLE III. LAYERS ALLOCATION.

Layer Number	***Signal/Power***
1	Signal
2	GND/Signal
3	VCCINT/VCCAUX
4	GND
5	VCCO/Signal
6	Signal

In addition, we need to consider general PCB layout design rules such as:

- Separate the power grounds, signal grounds, analogue grounds, digital grounds, control grounds.
- Keep inputs and outputs separated and isolate to prevent oscillations.
- Oscillation can happen from the inverting input and non-inverting input of op-amps, coupling between parallel signal traces.
- Place capacitors that bypass supply voltages or decouple very close to the IC chip pins.
- Use VIAS for signal grounds.
- Keep power and ground track/traces running close proximity.
- Need proper terminations of unused Op Amp pins or sections.
- Give good spacing and clearance between tracks and pads.
- Put reference designators in both schematic and PCB silkscreen to help troubleshooting.
- Put a lot of test points on the PCB for each stage/section to break down into blocks to help troubleshooting.
- Put a lot of board cut/jumpers to isolate sections & stages from the power supply, to each stage/section on the circuit to isolate them if needed to troubleshoot or separate the power supply if blown.
- Put connections on the edges of the PCB.
- Keep the Clock signal separate/isolated from analogue signals or amplification inputs.

[2] Automatic soldering for production, or when the size's parts is very small, for this prototyping work we will soldered by hand.

[3] When selecting components we should to ensure with distributors that are in stock.

VI. Assembly And Test

When the PCB design complete, we start purchasing all of the components that we selected so that they will be on hand for assembly, we print the PCB, only Gerber file can be sent to PCB manufacturer (The "PCB Artist" software have an automatic submitting by internet to the "Advanced Circuit" company for PCB fabrication). It is good practice to include the electric test when ordering PCB to verify net continuity and the absence of shorts on the board.

When receiving the board, start the assembling respecting the rules of the art. Do a visual test before turning on. If all happen normally after turning on ensure that ICs are not very hot or smoke doesn't come out.

If all is OK, choose a program that include all components functionality testing (DDRAM, Converters, switches, LEDs ...) or build a simple one. Charging the design in the board allows us to test the configuration mode components JTAG and EPROM functionality.

If all components functionality is OK, you can configure the FPGA with the final program.

VII. CONCLUSION

Through the design steps of this guide a PCB board based on Spartan-3E FPGA was realized for prototyping and production environment. Production environment was achieved by the Master-Serial Mode and prototyping environment with the JTAG mode.

The design of an FPGA is based primarily on its technology (EPROM, Anti-fuse, SRAM), and secondly on the configuration mode suitable for the application, these two parameters define the main steps of the design, and determine the basic components of the board.

Design steps of this guide are valid not only for Spartan-3E but also for other FPGA if we follow the configuration methods described by each FPGA Company and fulfil design instructions for each type. Generally each company has its own development tools and application notes.

Some design steps are common for every board design, not only for FPGA (Schematics design, PCB layout and routing), this is why we have not detailed, and we focused only on those considered specific for FPGAs

REFERENCES

[1] C. Maxfield, “FPGAs World Class Designs,” Elsevier, 2009.

[2] Peter J. Ashenden, “Digital design An Embedded Systems Approach Using Verilog,” Elsevier, 2008.

[3] C. Maxfield, “FPGAs: Instant Access,” Elsevier, 2008.

[4] Xilinx, “Spartan-3E FPGA Family: Data Sheet,” DS312 (v3.8), Xilinx, August 2009.

[5] S. Tapp, “Configuration Quick Start Guidelines,” XAPP501 (v1.5), Xilinx, October 2007.

[6] Xilinx, “The 3.3V Configuration of Spartan-3 FPGAs,” XAPP453, Xilinx, (v1.1.1), June 2008.

[7] R. S. Hoover, “Design tutorial and comparative analysis of printed circuit board production softwares for microcontroller & FPGA-based systems,” Fall, 2009.

[8] Xilinx, “Four -and Six- Layer, High-Speed PCB Design for the Spartan-3E FT256 BGA Package,” XAPP489 (v1.0), Xilinx, October 2006.