Advanced **L**ogic **S**ynthesis for **E**lectronics

http://www.alse-fr.com

Tornado FPGA Board : *Perfect (also) for the Education !*

Jan. 2005, ver 1.0 **Application Note 501**

Acknowledgements

We want to thank **Altera** (both France and Corporate USA offices) for their active support and for having helped us shaping and offering the board at an affordable cost. Thanks also to our friends at **Doulos UK** for the fruitful exchanges and their kind reviews and suggestions.

Introduction

Tornado being (also) a wonderfully well suited tool for **Digital, FPGA and ASIC Design courses** shouldn't be a big surprise since it has been designed by A.L.S.E. !

At ALSE, we are specialized in FPGA and ASIC design with a special taste for complex projects, but all our designers spend at least 30% of their time teaching the Doulos™ HDL Training Courses in France. When we designed Tornado, we did stuff it with as many features as we could, in order to cover many different aspects of the industry within a single board, but we also kept in mind our experience as Instructors !

An Engineering school will find in Tornado the right tool to teach the basic **Digital Design Concepts**, and to gradually move towards very **elaborate projects**, in utter simplicity.

Thanks to its **USB interface**, Tornado is bus-powered: it **doesn't even require a power supply**, and **no (zero) programming cable** is necessary either !
Last but not least, the design software (Quartus II Web Edition) is ***Free*** and runs on any Windows 2000 or XP PC !

For Educational use, the most important might be that we deliver Tornado with **ready-to-use Labs**, **Projects** and **IPs** (Intellectual properties) so the teacher will be able to build his own set of teaching Material with very minimal efforts.

So the only investment is to purchase the Kit... and it is very affordable !

One last word before we enter more specific topics : if you have already *other* vendor's (X) platforms, *you do not have to give up anything* !
This Kit is a wonderful *complement*, offering the complete Altera flow in an easy, affordable and productive bundle.

But the Know-How and the Labs are 100% portable to any platform !

We will see now in more details how Tornado can fit in the HDL and Digital Design teaching process.

Features

The heart of Tornado is the **Altera** FPGA **Cyclone 1C6**, based on a 1.5-V, 0.13-µm, all-layer copper SRAM process, beating at 60 MHz. This device offers two PLLs so you can run it at other different frequencies if you wish.

With nearly **6,000 Logic Elements** and **20 Ram blocks**, this cost-effective device allows a very wide range of applications up to complex ones. A RISC processor core or an FFT engine will nicely fit in the device...

Around the Cyclone FPGA, we have added **lots** of peripherals :

- **EPCS1 Programming Memory** : the board can be live at power up (and may be used autonomously, without the USB connection).
  This memory is programmable through the JTag connector.
- **USB 1.1, 2.0 compliant port** used for FPGA download and for high speed bi-directional transfers (we measured a sustained average of 8 Mbits/s).
- **E²Prom** for USB enumerator.
- **JTAG** connector.
- **4 x digits LED display** 7-segments + decimal point.
- **1 x** On-board **Speaker**.
- **2 x audio** outputs (**stereo** jack).
- **1 x** multiplexed Keypad interface (12 keys **"telephone type" keyboard**).
- **1 x fast ADC input (up to 60 Ms/s)** with miniBnc input connector.
- **2 x PS/2 or I²C or CAN bus ports** with PS/2 connector (to connect a keyboard and a mouse for example) with on-board voltage conversion.
- **4 x** protected **Digital Inputs** (-5..+20V, with Schmidt triggers).
- **8 x** High Voltage **Open Collector outputs** (up to 12 V - 100 mA).
- Dual 500mA **Switching Power Supply** (3.3V + 1.5V) powered by USB or external 5V power supply if desired (autonomous use).
  The board's **operating power consumption is very low** and the board won't drain batteries in robotics applications !
- Room for Power connector accepting **external 5V** regulated power supply for use without attaching the USB connection (Stand-alone use).
- **2 x RS232** (up to 400 kbauds level converter), one DB9, one HE10-10.
- 1 x spare configurable **bi-directional SPI** connector.
- **1 x ISO 7816 Compact Smart Card** socket (dual voltage: 5V and 3.3V) !
- **2 x Lines LCD** module connector HE10-14 (LCD module not included).
- **4 x Push Buttons** (user-assigned) + "Download" Push Button.
- **4 x DIP switches**.
- **4 x Digital Servo connectors** (robotics !)
- **4 x** discrete **LEDs** + 1 x Power LED.
- **1 x** On-board **Temperature sensor** (microWire).
- **1 x User Expansion** connector.
- Generous **Wire Wrap (prototype) area** with Gnd & Power pads nearby.
- Many **Test Points** for easy access to important signals with an oscilloscope probe.

Applications

All these features create a formidable potential for implementing very different applications. In the Education field, this platform can serve as a base for the simplest exercises (“see Your First VHDL or Verilog Application” later in this Manual), or for highly challenging designs like Real Time Digital Signal Processing applications, Digital Radio, GPS signal processing, Cryptography engines and so on...

Robotics adepts will be delighted by the embedded Servo outputs, analog input, protected I/Os, low power consumption, and by the possibility to use Tornado as stand-alone (without USB connection).

Moreover, students and teachers will discover that Tornado is also a wonderful Prototyping Platform !
Tornado can serve to build a functional mock-up system for an industrial application in no time, often just by using the embedded peripherals and I/Os, or by attaching simple interfaces to the user I/Os.

It is impossible indeed to list all the applications that could be implemented on this board, since many are still waiting to be invented !

We designed Tornado to be a good vector for **your** creativity.

Deliverables

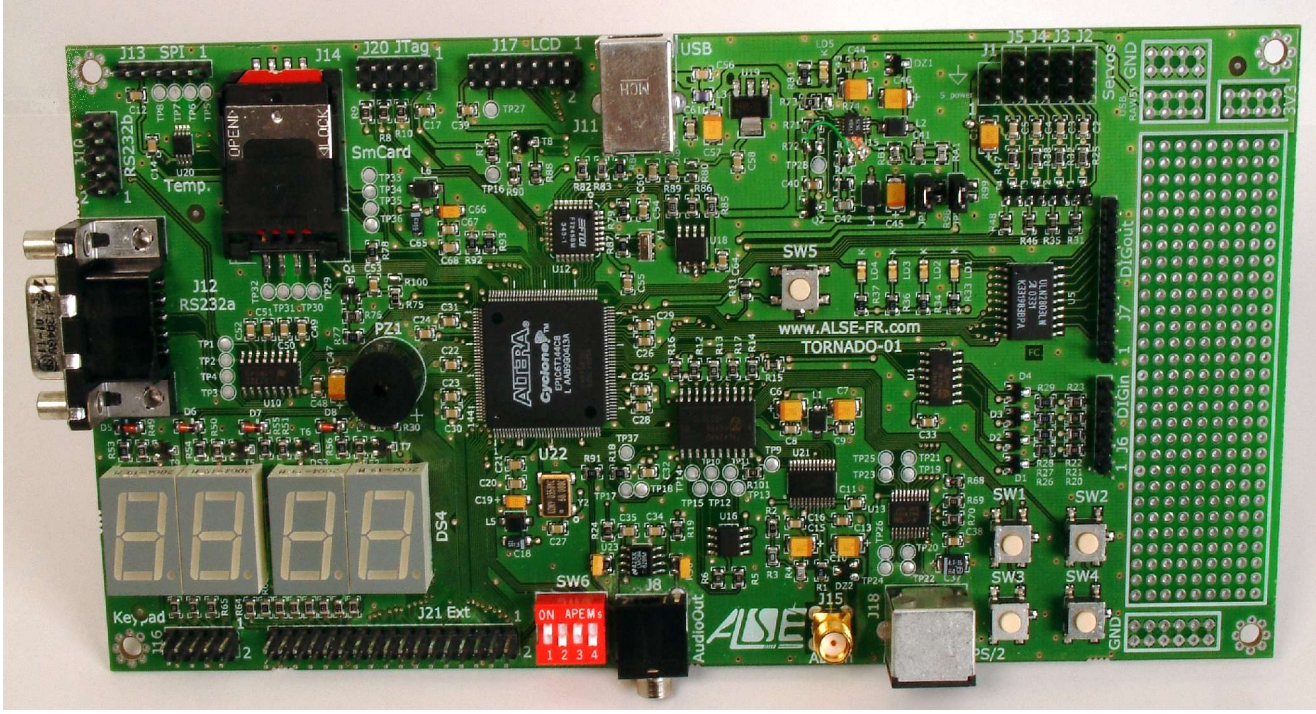
When you purchase a “**Tornado Education Kit -5**” you get :

- 5 x ready-to-use and tested Tornado Boards.
- 5 x standard USB type A-B cables aka as “printer” cables.
- One Altera Quartus II WebEdition CD-Rom.
- One Altera Documentation Library CD-Rom (optional).
- One ALSE CD-Rom including :
  - Installation Instructions,
  - Teacher's Manual, © ALSE,
  - Student's Labs Manual, © ALSE,
  - Design Resources Manual, © ALSE,
  - Sample applications,
  - Source code, Project files, bitstreams, etc...
  - USB drivers, Download utility and test bitstream,
  - Possibly other recent information : see the **readme** file.

The three manuals above are delivered in OpenOffice (.sxw), MSWord (.doc) and Acrobat (.pdf) formats. The Teacher is granted the rights to cut and paste from these documents to assemble his own printed materials. However, the source code's headers can **not** be modified.

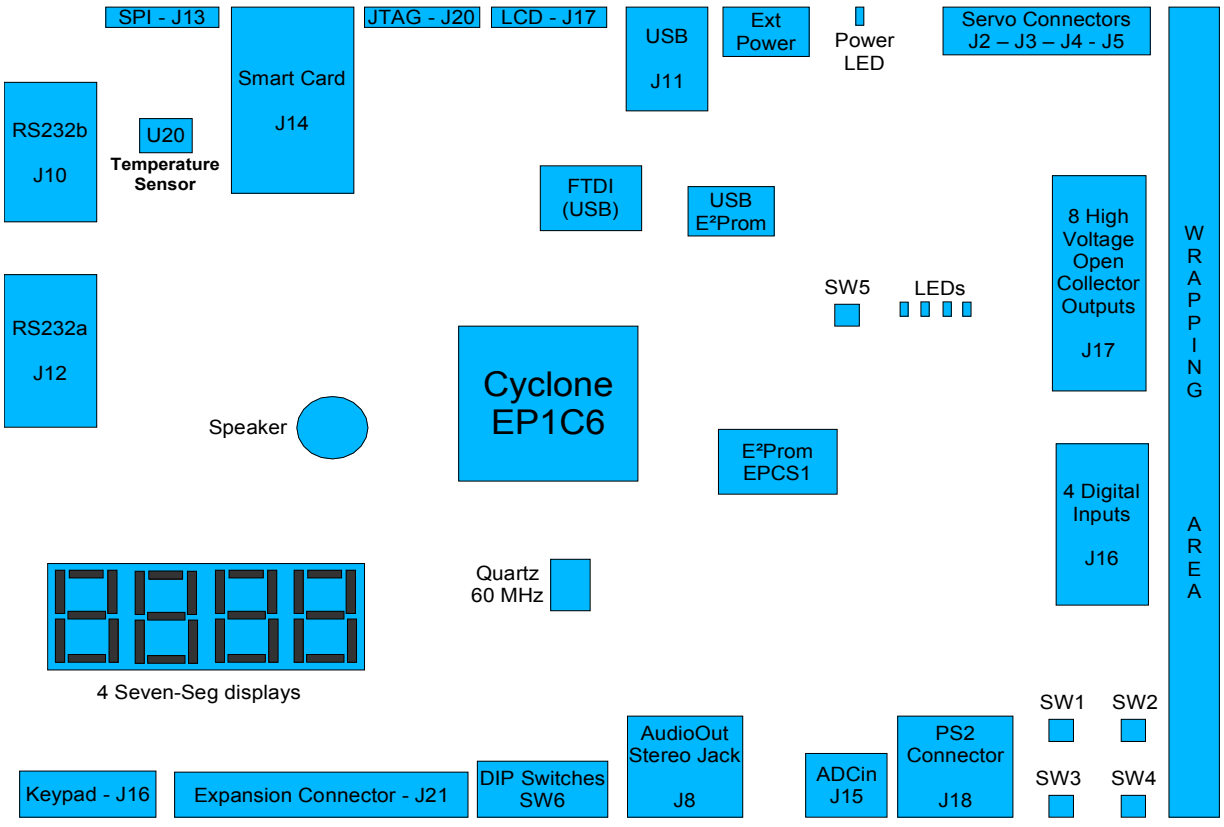
Tornado's Picture

The picture below was taken from our first prototype, but very few changes or enhancements were necessary for the final version (the main change was a different buffer chip to take care of the bi-directional logic level conversion with the I²C-PS/2).



Tornado's Block Diagram

The diagram below does illustrate the main functional blocks available in Tornado.



Digital Design Labs © ALSE

To help teachers and students in getting up to speed in no time, we have designed a set of **five (5) ready-to-use HDL Design Labs**, complete with solutions and optional guidance material.

These Labs, including all the source and simulation files, scripts, bitstreams, Tornado schematics, etc... as well as the teaching material in source form with right to modify are **free** to customers purchasing a package of at least five (5) Tornado boards (*Tornado Education Kit*).

Lab #1 : **4x Digits BCD counter**.

This is the simplest Lab. The student must write the HDL source for a synchronous 4 x digits BCD counter with enable (~40 lines of code).
This module should then be unitary-tested by simulation with a test bench written by the student.
Once this module is tested, the integration in Tornado is very simple since we provide a ready-to-use top-level file with a complete FPGA compilation script. The decoding & display of the BCD vector is taken care of in the top-level so the student doesn't have to do this.

If the prepared material is provided to a beginner, Tornado should display a fast counting counter within 2 hours, including the verification by simulation.

Indeed the teacher can adapt the Lab to different audiences, and decide to not provide the template files that we deliver...

Lab #2 : **Simple PWM**.

The purpose is to create “Beating Heart” blinking LED (*gradually* turns on and off). The simplest technique to create an “analog” output is to adopt a Pulse Width Modulation scheme. This PWM coder is driven by a slow up/down/up counter, creating the desired visual effect.

This Lab, though very simple (the solution is 30 lines of HDL code), is a good introduction to PWM techniques, and represents a small HDL programing challenge. Careless designers will easily create buggy code which will be well debugged by HDL simulation.

If the prepared material is provided to a beginner, Tornado should display a “heart beat LED” within 2 hours, including the verification by simulation.

Lab #3: Digital Thermometer.

This design should **display** in real time the current ambient **Temperature** in Celsius degrees on the 4 digits located on the board : “24°C” (eg).

Tornado includes a Temperature sensor chip with a microWire (SPI-like) interface. The student must therefore design the µWire interface, and a simple binary to BCD conversion.

This specification can lead to several different implementations, each with pros and cons (usually ease of initial writing vs ease of debugging and implementation efficiency).

The best implementation uses a Finite State Machine to interface with the µWire device, an extremely powerful and versatile style.
A simple behavioral model of the Temperature sensor is provided, thus helping the student in debugging his µWire interface.

This lab does propose as an option to design a serial Binary to BCD converter thus taking advantage of the inherent bit serialization.
The simpler-to-understand solution is a “brute-force” combinational decoder which, in this case is still small and will work nicely.
Another option would be to convert the Celsius degrees into Fahrenheit (“brute force” but viable solutions -look up tables- exist).

If the complete ALSE material is provided for guidance, Tornado should display the temperature within 4 hours, including the verification by simulation.

This Lab is extremely useful, since the SPI interface is very common.

Lab #4: **Dual Tone MultiFrequency (DTMF) Generator with Keyboard**.

This design decodes a 12 keys telephone type multiplexed keyboard, displays the Key code on the LED, and generates the proper DTMF signals on the audio output. Plugging speakers should dial any phone number on a telephone set from Tornado.

The skills involved include : scanning and decoding the matrixed keyboard, generating accurate sine waves, and creating the analog output through a one bit DAC. All these techniques are very useful in many different situations.

We provide a “how to” guide which explains the algorithms involved which aren't very complex. The array of hexadecimal values for the half sine wave is included in the document. If the student has enough time, this table wouldn't need to be provided. The implementation of these algorithms is also quite straightforward. It wouldn't take a skilled designer more than a couple of hours. With the guidance material that we provide, a student who went through the previous Labs could reach the solution within 4 hours.

Optionally, the input could come from a PC keyboard (PS2) (a read-only PS2 interface is freely available on ALSE's website) with the advantage of not requiring a telephone-type matrix keyboard (not included with Tornado).

Lab #5: Designing an RS232 UART and Serial line data Scrambler (56 bits DES)

This Lab is a more serious challenge. But if the student has successfully gone through the previous Labs, he should have no problem designing a working UART. Then, the DES encrypter (delivered as a working/tested ready-to-use IP) must be fitted in the design through an ad-hoc control logic, and last it should be clear that the system also needs some buffering, so we add a Fifo and its control scheme.

The recommanded steps are :

- Creating the UART-receiver section, and test it. We provide a behavioral model for the UART transmitter which facilitates the verification.
- Creating and testing the UART-transmitter section. This part is easier than the receiver and can be tested with it.
- Understanding how the D.E.S. module works. Though it is delivered with all the commented source code, there is no need to understand how it works *internally*. The DES module must be treated as an “IP” or as a “black box”. A specific test bench shows the DES module in action.
- Create and verify a D.E.S controller. There is a small challenge here due to the fact that the D.E.S handles 64 bits data while the serial line's data path is 8-bits... The unitary simulation of this block should use file I/O in order to create files with crypted data. Since the encrypted data is no longer “text” but “binary”, this creates some issues.
- Assemble the blocks in the top level.
- Verify the complete system with a test bench. You should notice that there is an issue with the data flow integrity if the characters to be encoded flow without interruption. Think of possible workarounds and strategies.
- You can test the system on Tornado : chances is high that you'll have missing characters when sending a file from the PC.
- You may find out that adding a kind of “elastic buffer” could help. Add a 512 words Fifo, verify by simulation, and test on Tornado.

Testing the system on Tornado is easy : simply attach a serial cable between a PC (with Hyperterminal e.g.) and Tornado, and start typing to see the encrypted output. A sample “mysterious” text file provided, sent to Tornado should be echoed (returned) as a clear (decoded) text.

This assignment should require no more than 6 to 10 hours of personal work plus 2 to 4 hours hours with the hardware platform from the Specifications (and including the guidance documents that we provide) to a working system.

This is not The End !

Obviously, there is no need to stop here !

Most of the board's features are still unexplored at this stage, and we encourage the Teachers and Students in visiting the section of our WebSite which is dedicated to Tornado :

http://www.alse-fr.com/tornado

We encourage all the Tornado users in sharing their suggestions, their designs and applications. Plans are to open a Discussions Forum.

If there is demand, we may design a series of more advanced Labs,

Summary

As we've seen throughout this document, the “Tornado for Education” kit is a great opportunity for organizations who want to teach HDLs seriously and make sure their students end up with practical, efficient, and solid **Digital Design know-how**.

The design techniques involved in these Labs suggested in the Guidance material, and used in the proposed solutions, are State-of-the-Art, reliable, and highly productive. They take advantage of modern tools, HDL languages and methodologies.

Why and how did we come out with these Labs ?
At ALSE, we've been teaching Digital Design and HDLs for over twelve (12) years ! We've helped customers with malfunctioning hardware and witnessed a lot of despair... Browsing the source code available on the Internet or even examples delivered in many tools shows that there is still room for a lot of improvement in RTL coding styles !
On the other hand, we have designed tens of systems, sometimes extremely complex ones) with real ease, by applying the very concepts that we teach.

Therefore, we've put in this Kit the Know-How that we think is essential.

We hope that Students will discover a new realm of creativity, and how easy it can be to design complex systems with Altera Programmable Devices and Design Tools. This Kit and the Labs prove that applying a sound methodology (with simulation, synchronous design, state machines...) is quickly paying off.

HDL Course Contents Suggestions

We assume that the basics of Digital Design has been taught, with proper emphasis on known issues, strong incentives towards synchronous design and proper caveats about the bad practices (latches, combinational feedback, combinational logic and glitches, etc.).

In the VHDL or Verilog course, we suggest the following coverage :

- Digital Design Reminders : Combinational logic, Logic reduction, FlipFlops, synchronous design, dynamic performance, static timing analysis, Finite State Machines principles, Binary arithmetic, Memories and Fifos, FPGA backgrounder, MacroFunctions...
- HDL Basic syntax, Entities / Modules
- Types & Signals/Regs
- Continuous assignements and simple combinational logic
- Instanciation and structural HDL.
- Processes/Always & Variables
- Test Benches / Test Fixtures.
- Synchronous process /always
- Combinational process / always, caveats latches & sensitivity lists.
- Arithmetic
- Inference vs Instanciation
- Subprograms
- Advanced Test benches / fixtures, applyring vectors and capturing outputs in a timely manner, File I/O, Self-testing test bencges, using Golden models, On-the-fly compartison.
- Complete Design flow, post layout timing simulation.
- Using Advanced FPGA features : PLLs, DP-Rams, FIFOs, embeded Arithmetic blocks, ...
- Advanced topics, Assertion-Based Verification, PSL, using the IEEE Mathematic libraries, stochastic & advanced instructions, Asic design flow, formal tools, C to VHDL, Digital Signal processing & Hardware Implementation...

Tornado and the proposed Labs will fit very nicely within a similar cursus, allowing for practical implementation and test of all the concepts taught.

It goes without saying that we teach precisely these Courses !

If you are interested by our Training Courses (Essential Digital Design, VHDL, Verilog, Expert VHDL, Expert Verilog...), please :

- Contact ALSE for France: http://www.alse-fr.com,

- Contact Doulos for other countries: http://www.doulos.com.