Design of a Digital Audio Receiver in a VLSI Lab

Peter Blinzer

Technical University of Braunschweig
Department of Integrated Circuit Design (E.I.S.)
Postfach 3329, Gaußstraße 11, 38023 Braunschweig, GERMANY
email: blinzer@eis.cs.tu-bs.de

Abstract. This paper describes the successful implementation of a real world design task from consumer electronics in a university design lab. Within one term, students designed a circuit for digital audio decoding as a Verilog model and an FPGA implementation. This design task allowed to practice a basic chip design strategy on a nontrivial problem and was highly motivating for the students, since audible results were produced by a hardware testbed.

1 Introduction

Design labs are an important part in academic education for the design of integrated circuits. Lectures may be sufficient to supply a good theoretic base in this area, but certainly do not allow to get experience in the use of design methodology and software tools. By the combination of lecture and design lab, the essential aspects of chip design can be covered.

In a design lab, the design task has to meet several requirements. One of the most important requirements is the practical applicability of the design methodology taught. Also important is the size of the design, which must allow completion of the design task in the available time. Student motivation of a design task is another aspect not to be ignored, since it allows to unleash their creative power. Of course, there are several other things to be taken care of, but these three play a key role for a successful design lab.

The chip design course offered by the Department of Integrated Circuit Design at the Technical University of Braunschweig is also based on lectures given in parallel. In the 1996 summer term, a design task from consumer electronics was used in the FPGA design lab. Students had to design an interface circuit, which decoded a digital audio signal supplied by a standard audio CD player and which preprocessed the data for a digital to analog converter used for audio playback.

This design task proved to be well suited to practice the design of an integrated circuit by the use of HDL modeling and manual schematic entry. All participants worked with great enthusiasm and solved the given problem successfully in design groups of three students.

2 Organization of Design Lab

The design lab in the 1996 summer term had a duration of 12 weeks. Prerequisite for participation were the CAD lab exercises held in the preceding winter term, which gave the students a first insight into the CAD tools.

Every week there were two lab sessions of 90 minutes each at CAD workstations. This time turned out to be just sufficient to solve the task. An equal time at home was estimated to prepare lab sessions and to write a project documentation.

A description of the design problem was already handed out at the end of the winter term. This allowed the students to get familiar with the task and to recognize comprehension problems in the spring break time.

The lab itself was subdivided into five phases (Figure 1). At the end of each phase, certain intermediate results had to be completed. These were reviewed for correctness and completeness and discussed in a colloquium with the group members. The design phases covered certain parts of the whole design flow and therefore not only provided useful checkpoints both for supervisor and students, but also stressed out points of key decisions in the design process.

week	phase	design target
1 2 3	I	Verilog model
4 5	II	complete schematics
6 7 8	III	correct schematics
9 10 11	IV	FPGA configuration
12	V	project documentation

Figure 1: Design lab schedule

In the first lab phase spanning three weeks, an executable model in the hardware description language Verilog had to be created using register transfer level abstraction and a fully synchronous clocking scheme. A Verilog testbench was provided to test the correct function of this model. In this phase, some of the computer based design work was also done outside the lab by most students, having access to a PC and being able to use the free version of the Verilog simulator VeriWell [1]. A documentation of the Verilog model was also required, which described the modular design structure and the interfaces.

The second phase of the lab lasted two weeks. In this design phase the Verilog model had to be transferred manually into a complete schematic representation. The hierarchy had therefore to correlate with that of the Verilog model where possible and to be extended when necessary. The schematics had also to be documented with respect to hierarchy, module structure, function, and interfacing.

In the third design phase, the schematics were verified by simulation and corrected if necessary. For each nontrivial module in the design, a test for the basic functionality had to be designed and documented. As in the first phase, a testbench was provided for the test of the complete design permitting the reuse of test data for the Verilog model.

The fourth phase was used for the optimization of the already functionally correct designs. The optimization was based on the resource and timing limitations of the target FPGA, so place and route was also done in this phase using the XACT software from Xilinx. The optimization done in this phase consisted mainly of removing redundant logic, of pipelining, and of redesigning simple sequential logic such as FSMs and counters by the use of one hot encoding. The phase was concluded by a postlayout simulation of the successfully placed and routed design.

In the final week, the students compiled a consistent project documentation based on the documentations of each design phase. This was discussed in the last colloquium, where the design was also tested in a real testbed. It consisted of an audio CD player and some analogue interface circuitry for the FPGA chip to make the design success (or failure) audible.

3 The Design Task

The task was to design an interface circuit for a digital audio signal within a single Xilinx FPGA of the type 3042-70 [2]. From a previous prototype design, this device was known to be sufficient. The source of the digital audio signal was a consumer electronics audio CD player and therefore provided a signal complying with the interface standard of Sony and Philips, SPDIF [3,4].

This serial signal was amplified from its ±200mV level to TTL compatible voltage level by an input amplifier in the testbed and then fed into the FPGA. It was at this point still biphase encoded and had a bit transfer rate of 5.6Mbit/s. To receive this signal without the use of a PLL clock regenerator, a constant clock frequency of 16MHz was used to drive the FPGA, which had to sample the bitstream with approximately three samples per code bit and to extract the bitstream data based on the biphase encoding. Since this receiving process was the most complex and time critical function of the FPGA, the design project got the name digital audio receiver (DAR).

The single data bits had then to be grouped into the SPDIF data blocks of 32 bits for each stereo channel. After extraction and validation of the audio control information, the audio sample data had finally to be serially transmitted to a two channel digital to analog converter (DAC) [5] in the testbed. The outputs of this DAC were connected to an audio amplifier to which speakers could be connected, which made the output audible. The simplified system diagram of the complete testbed is shown in Figure 2.

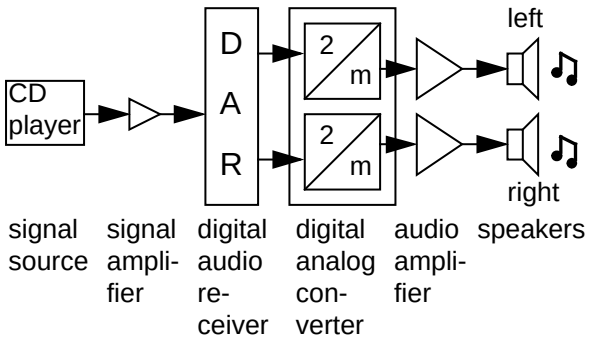

Figure 2: System diagram of the DAR testbed

One of the basic requirements for the successful completion of this task was to understand the SPDIF standard for digital audio transmission. SPDIF specifies a unidirectional self-clocking digital connection for serial transmission via a single coaxial cable or an optical fiber link.

The self-clocking of the signal is performed by a special encoding of the transferred data bits, the biphase encoding. In this code, every data bit is represented by two code bits, such that a logical zero yields two equal code bits (00 or 11), a logical one yields two different code bits (01 or 10) and a signal edge does occur between two encoded data bits. The decoding of data and clock is possible by locating the signal edges and measuring the time between them. The start of data blocks is marked by special preamble codes. These contain certain violations of the biphase encoding scheme and enable synchronization of both blocks and bits.

Some hints were given for a possible solution approach based on a prototype design, including a diagram of its coarse structure (Figure 3).

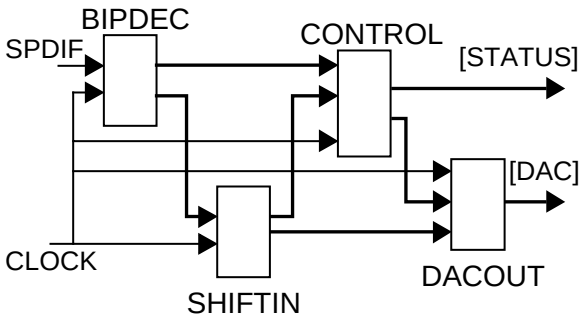

Figure 3: Coarse structure of the DAR

The function of the modules was characterized on a relatively abstract level to guide the design process, but not to steer it.

The module BIPDEC was therefore suggested to receive SPDIF signals by determining the bit synchronization, decoding the biphase bitstream and finding the preamble codes of the SPDIF data blocks.

Composition of 32 bit data blocks from the incoming data bits was devoted to module SHIFTIN, being nothing more than a serial input parallel output shift register enabled by the bit synchronization. The module CONTROL gathered the status information of the signal, checked it for validity and enabled the output module DACOUT accordingly.

4 Special Design Aspects

The design task had some special aspects, making it both challenging and valuable as a chip design exercise.

The restriction of logic resources and circuit speed by the given target FPGA type (3042-70) made it necessary to take care not only of timing issues, but also of logic resources. The students therefore had to think thoroughly about where to use serial and parallel processing and where to use techniques such as one hot encoding and pipelining. These restrictions also required all groups to actually do limited redesigns in the optimization phase of the lab.

Since there was no restriction on the timing relation between signal input and audio output and the internal FPGA structure supported pipeline optimization by almost no additional resources, there were a lot more optimizations possible than really needed. This made design teams aware, that a design flow may have cycles and that there is more than one solution.

Despite of a final test in the hardware testbed (Figure 4), all intermediate tests for the digital audio receiver were done by simulation. The students therefore had to develop sets of test patterns for their designs to check functionality and to identify design errors. Due to this test strategy, the design flow was comparable to that of mask programmable ASICs.

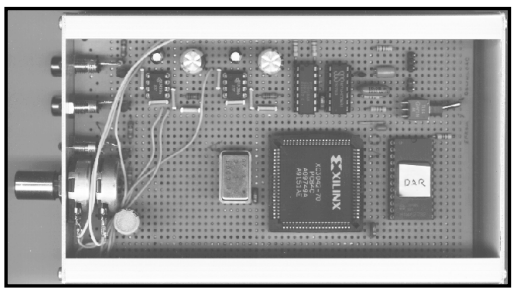

Figure 4: Hardware testbed of the DAR

The design experience gained is therefore a good base for the solution of design problems requiring a first time right approach due to size and cost.

5 Results

In the 1996 summer term, there were 15 participants in the FPGA design lab, which were working in five groups of three students. All five groups solved the design task successfully by passing the post layout simulation. Three of these designs worked without any problem in the hardware testbed, while two produced a slightly distorted sound. The distortions were caused by data loss due to problems in bit resynchronization, which was not uncovered by the test data.

In a concluding survey students noted, that they would recommend the lab, although it had required a lot of work. The kind of design task was found very motivating, because a widely used audio transmission technique by a digital connection was made transparent, and an audible result was produced.

6 Conclusion

The design task of the digital audio receiver was successfully applied in our FPGA design lab despite its complexity. It met the requirements to practice a design methodology and to be motivating for the students. It allowed also to check design alternatives and to prove functionality of the designs within a real time and real world environment.

References

[1] Wellspring Solutions Inc, VeriWell^{tm} Users Guide 2.0, June, 1995, (available at http://www.wellspring.com)

[2] Xilinx, The Programmable Logic Data Book, 1994

[3] International standard IEC958, digital audio interface, First edition 1989-03

[4] Francis Rumsey, John Watkinson, The Digital Interface Handbook, Focal Press 1993

[5] Analog Devices, AD8522 Data Sheet, 1994