Education Software for Courses on Theory of Information

Karel Vlček
Dept of Measurement and Control
FEI, VŠB Technical University of Ostrava
Ostrava-Poruba, Czech Republic
E-mail: karel.vlcek@vsb.cz

Key Words: Information Theory, Source Coding, Channel Coding, VHDL Modelling, Noise-Channel Model Simulation.

Annotation:

A new ways in Information Theory education are practised at the VŠB Technical University of Ostrava, Czech Republic. The courses have a conception of the both theoretical lectures and practical exercises. The simulations of digital systems bring a new look into theory and practical application of the information theory as a classical discipline of discrete mathematics. The both textbooks [6,7] arouses on the basis of the university courses which started four years ago for the students of Communication and Computer Science. The courses are oriented to foundations of discrete mathematics and its applications, as well as to communication and testing of electronic circuits and systems.

INTRODUCTION

The aim of courses on Information Theory is to introduce the methods and procedures, how to describe and evaluate the properties of processing and transmission of information. On the other hand, the author makes an effort to give a practical view to design of circuits for encoding and decoding. The text is written for designers of electronic circuits, who must build the coding and decoding equipment and for the communication system designers who must incorporate circuits into a system for message transmission. Texts uses only basic mathematics relations.

The classical mathematical presentations in theorem/proof are avoided. In spite of that: coding is extremely mathematical subject and it was an impossible task to present the matter without mathematics at all. The practical designers are mostly interested in using mathematics but not in constructive rigorous proofs. The applications of digital technique in computers as well as in memory medium become decisive matter of course and a condition of growth of computational power. Digital data transmission within computer systems are intolerant of even very low error rates usually, because a single error can destroy a computer programme. Error-control coding is becoming important in these applications.

DESIGN METHODOLOGY

Some parts of the courses are dedicated to description of circuit implementations of encoders and decoders. To be technology independent, the descriptions are done in VHSIC Hardware Description Language (VHDL), which becomes the universal description mean of digital circuits. The hierarchical structure of VHDL is ideally suited for description of extensive electronic circuits and systems, which are needed with the requirement of high speed of communication. Without exaggeration it is characteristic, that coding is necessary to

functioning of satellite communication links, optical memories of computers, and for example mobile communication. Other aspects of applications, which are not visible: the support of testing of circuits and systems.

Courses on VHDL are lectured at the Technical University of Ostrava from 1993. The giving of lectures started from the co-operation with the Technique High-School of Gent in Belgium in the frame of TEMPUS Project in 1993. The courses are prepared for students in the 2nd period of pre-graduate studies. The fundamental lectures are divided into three specific areas: i) the state of the art in circuit and system design, ii) the VHDL like a programming language, and iii) the BSDL programming in design for test and self-test systems.

The content of this three specific areas can be illustrated in some sentences. The first explanation is devoted to the reasons which give rise of the VHDL (Very High Speed Integrated Circuits Hardware Description Language). The very complicated situation was in circuit design, implementation of VLSI (Very Large Scale Integration) and its technology innovations required radical changes in verification and testing of designed circuits and systems in the middle of 80th. The complex process of design must deal with the test designing.

DESIGN SUPPORT MEANS

The content of the three specific areas: i) Circuit design, ii) VHDL Description, and iii) Design for Test can be illustrated in some sentences. The first explanation is devoted to the reasons which give rise of the VHDL. The very complicated situation was in circuit design, implementation of VLSI (Very Large Scale Integration) and its technology innovations required radical changes in verification and testing of designed circuits and systems in the middle of 80th.

The complex process of design must deal with the test designing. The high level design methodology is applied thanks to System GALILEO, which is used in exercises in Workstation Lab. New designs can be captured quickly, and existing designs can be quickly re-targeted to a new FPGA architecture. The time explorer gives very detailed view to all time responses, and allow to analyse the quality of the design. A lot of students' designs was done in the four years of giving the lectures. More then ten of the designs was implemented in the frame of diploma projects by ANTI-FUSE FPGA of Actel with co-operation of the enterprise PHOBOS in Frenštát pod Radhoštěm.

The specific design language properties of VHDL allow the designer to use the routines results in average two times improvement in the logic capacity of the programmable devices. The CPLDs and FPGAs architectures assume the technology specific optimisation techniques, including algorithms for state machine and glue logic, and module generation for data path and arithmetic logic, to take maximum advantage of unique architectures for significant speed and area reductions. The variety statements is given in the VHDL behavioural level for optimal compilation, which is different for so called "fine grain" or "coarse grain" FPGAs. The truth table constructs, which are optimal for description of CPLDs architectures can be arbitrary combined with the behavioural description of other parts of architecture by the entity declaration. The global as well as local variables can be used for hierarchical description [2].

An arbitrary modelling level can be used interchangeably. It is possible to present a system at virtually any level of abstraction [6]. The VHDL provides a very broad of data abstraction capabilities, such as user defined types, enumerative types, and composite types. The simulation uses a behavioural model of the major functional elements of the architecture,

combining many parts of the circuit into logical elements that perform some overall function. The RTL-level modelling is one step closer to hardware implementation. The advantage of this step is the close continuity to gate array library. The disadvantage is the limitations in re-targeting of model to new FPGA types.

The design methods must be versatile and technology independent for of CPLDs (Complex Programmable Logic Devices), FPGAs (Field Programmable Gate Arrays), and CMOS ASIC design. Students as well as professional designers can quickly efficiency, and economically consolidate multiple designs into one larger design, retarget a design, and use VHDL to accomplish their designs. The programme packages can optimise the designs for area and/or speed. The VHDL accepts designs described as equations, truth table descriptions or interconnection descriptions.

THIRD COMPANIES ROLE

A lot of the third companies products can synthesise VHDL register-transfer-level (RTL) descriptions into FPGAs. New designs can be captured quickly, and existing designs can be quickly re-targeted to a new FPGA architecture. High level design allows one to use the same design methodology regardless of target technology. It generates the specific FPGA netlist. This ensures present and future compatibility with design implementation systems from FPGA suppliers. The designer can describe a top level design by top level schematic. Behavioural descriptions have two basic forms: algorithmic and data flow. The difference between these two forms can be illustrated by means of examples. The both examples have the same entity [3]:

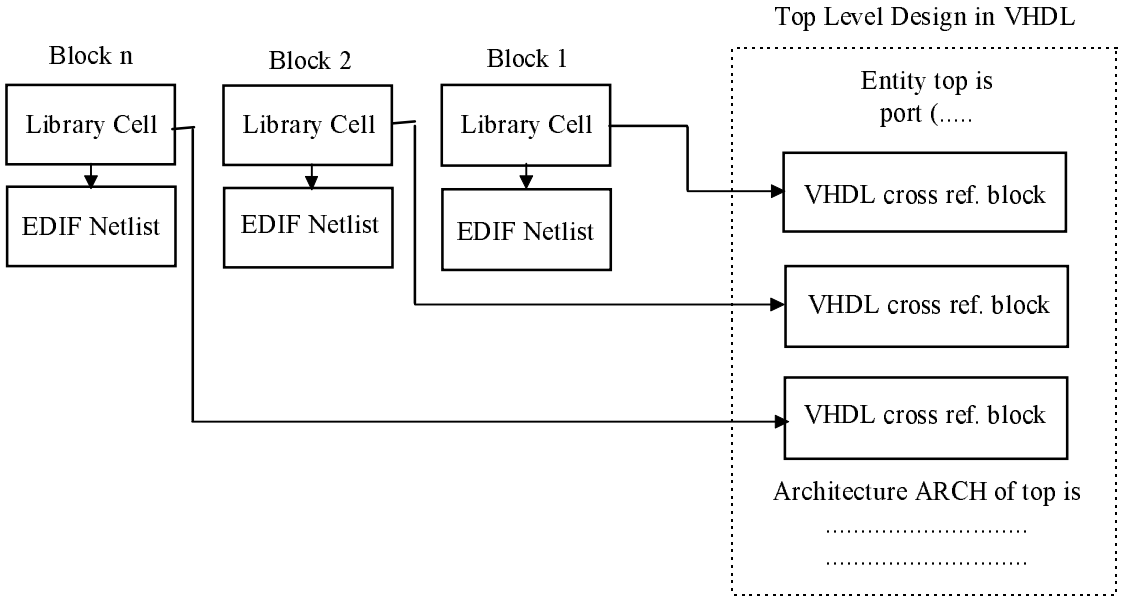

Figure 1: *The VHDL Top-Down Design Methodology*

Determining of FPGA performance requires knowledge of the functionality, branching of signal wires, and specific routing [1]. Exact timing is done by implementation of the logical function in the gate array. The top level design can be done by VHDL source file or a top level schematic. When the VHDL file is used as the top level design source, one can reference hierarchical design blocks in addition to VHDL architectural description blocks. The description blocks can be mapped incorporating the information from these external sources

into the top level design netlist [2]. The VHDL top level design with hierarchical blocks represents the design information from the "external" input sources.

IMPLEMENTATION TARGET

The design methods must be versatile and technology independent for of CPLDs (Complex Programmable Logic Devices), FPGAs (Field Programmable Gate Arrays), and CMOS ASIC design. Students and designers can quickly efficiency, and economically consolidate multiple designs into one larger design, retarget a design, and use VHDL to accomplish their designs. The programme packages can optimise the designs for area and/or speed. The VHDL accepts designs described as equations, truth table descriptions or interconnection descriptions.

The specific design language properties of VHDL allow the designer to use the routines results in average two times improvement in the logic capacity of the programmable devices. The CPLDs and FPGAs architectures assume the technology specific optimisation techniques, including algorithms for state machine and glue logic, and module generation for data path and arithmetic logic, to take maximum advantage of unique architectures for significant speed and area reductions.

The variety statements is given in the VHDL behavioural level for optimal compilation, which is different for so called "fine grain" or "coarse grain" FPGAs. The truth table constructs, which are optimal for description of CPLDs architectures can be arbitrary combined with the behavioural description of other parts of architecture by the entity declaration. The global as well as local variables can be used for hierarchical description.

ERROR-CONTROL LIBRARY

The construction of Meggit decoder is based on the property of the cyclic codes: We can concentrate on the last position of each received word w, which we correct or not, according this syndrome.

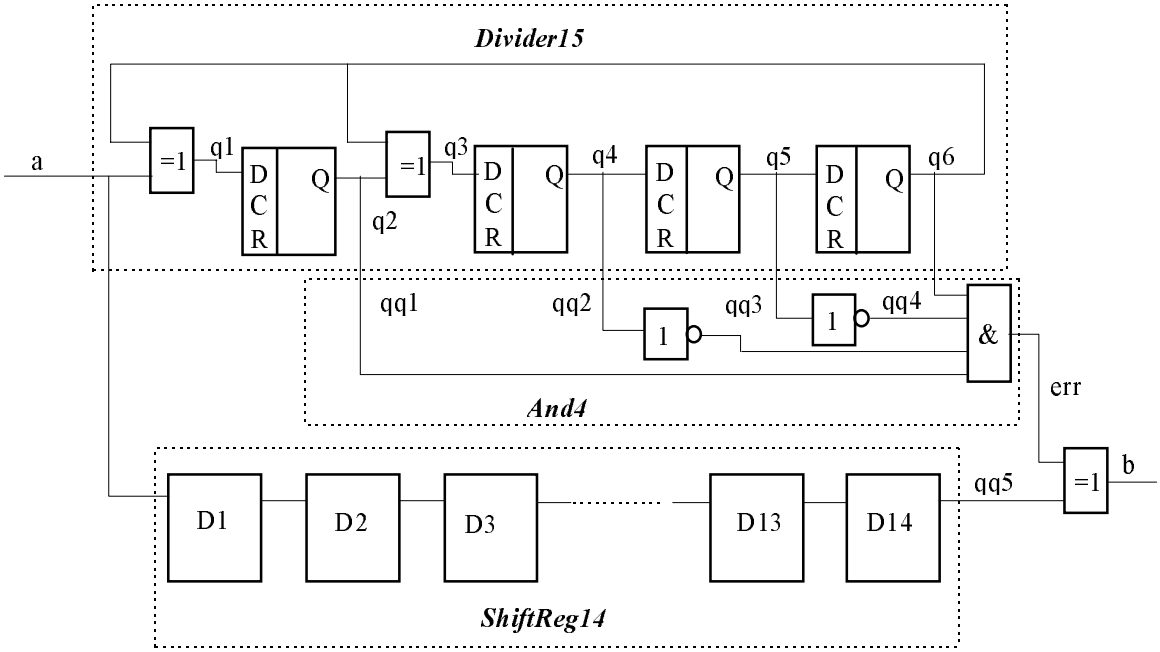

Figure 2: Block diagram of (15,11)-code decoder circuit

Then we make a cyclic shift of code word *w*, and again study the last position, etc. After *n* cyclic shifts, all positions will have been corrected. For the generating polynomial $g(x) = x^4 + x + 1$ we can compose syndrome computing circuit for the systematic cyclic Hamming (15,11)-code.

To correct an error we will use the syndrome value 0001. To be the received word corrected it must be delayed 15 steps. This delay is realised by the shift register with 15 flip-flops. For generating an correcting bit it is used the circuit completed by two inverters and one four input gate with AND function and concatenated gate with EX-OR function.

MODEL OF DECODER CIRCUIT

The modelling is defined by description of behavioural models of elementary parts of circuit. There are defined entities of flip-flop, EX-OR gate, inverter, and AND-gate:

```
entity hamdec15o1 is
   port (a, cl, rr: in bit;
              b : out bit);
end hamdec15o1;

architecture hd15a of hamdec15o1 is
   component Ex_or
    port ( In1, In2 : in bit;
              Out1 : out bit);
   end component;
   component ShiftReg14
    port (in1 : in bit;
          cl,r: in bit;
          out1: out bit);
   end component;
   component And4
    port ( in1, in2, in3, in4: in bit;
          out1 : out bit);
   end component;
   component Divider15
    port (a, cl, rr: in bit;
          m1,m2,m3,m4: out bit);
   end component;
   signal qq1,qq2,qq3,qq4,qq5 : bit;
   signal err: bit;

begin
   n2: Divider15  port map (a,cl,rr,qq1,qq2,qq3,qq4);
   n5: And4       port map (qq1,qq2,qq3,qq4,err);
   n6: ShiftReg14 port map (a,cl,rr,qq5)
   n7: Ex_or      port map (err,qq5,b);
end hd15a;
--------------------------------
```

The example of Meggit decoder is composed as the interconnection model. The partial models of interconnection are described in the library.

CONCLUSION

The Meggit decoder is realised as an structural model composed from behavioural models of components. Some components "`Divider15`", "`And4`" and "`ShiftReg14`" are modelled by structural models too. The model was verified by functional simulation using the software suite V-System. Decoder corrects one error when two errors occurs decoder detect error but it cannot be corrected.

Support for GACR project "Research and Development of Built-in Diagnostics Means of Integrated Circuits" (No. 102/98/1003) is gratefully acknowledged.

References:

1. Adámek, J.: Kódování. SNTL Praha (1989) (In Czech)
2. Adámek, J.: Kódování a teorie informace. ČVUT Praha (1991), ISBN 80-01-00661-1 (In Czech)
3. Blahut, R.E.: Theory and Practice of Error Control Codes. Addison-Wesley (1983)
4. Dewey, A.: Design Automation. IBM Enterprise Systems CTU Prague (8-11 March, 1993).
5. IMEC: VHDL: Intensive Course. Leuven, Belgium, (August 1993).
6. Coelho, D. R.: The VHDL Handbook. Kluwer Acad. Publish. (1989).
7. Cohen, B.: VHDL Coding Styles and Methodologies. Kluwer Acad. Publish. (1995).
8. Cohen, B.: VHDL Answers to Frequently Asked Questions. Kluwer Acad. Publish. (1998).
9. Vlček, K.: Theory of Information and Coding. VSB-TU (1998), ISBN 80-7078-494-6 (In Czech)
10. Vlček, K., Miklik, D.: The VHDL Model of DEC-TED Memory Checker. Proc. of Ninth Europ. Workshop on Dependable Computing. (May 14-16, 1998, Gdansk, Poland), pp. 54-58.
11. Vlček, K.: The VHDL Model of Wzner-Ash Channel Coding for Medical Applications. Proc. of DECS, Szczyrk, (2-4 Sept., 1998), pp. 145-151, ISBN 83-908409-6-0
12. Vlček, K.: Theory of Information, Coding and Cryptography. VSB-TU (1999), ISBN 80-7078-614-0 (In Czech)