PHYSICAL PROTOTYPING OF MICROPROCESSOR DEVICES

Frantskevich K.E.
Belarusian State University of Informatics and Radio Electronics. Minsk, Belarus
Stempitsky V. R. – Candidate of Technical Sciences, Associate Professor

Annotation. Microprocessors and microcontrollers are one of the most dynamically developing areas of modern electronic technology. The development of computer technology has led to the development of description languages for digital equipment. Hardware description languages are an integral part of CAD, especially for complex circuits such as specialized integrated circuits, microprocessors, and programmable logic devices. Modeling using HDL (Hardware Description Languages) allowed engineers to work at a higher level of abstraction than modeling at the circuit level, and therefore simplify the modeling of projects consisting of thousands of transistors and more.

The rapid growth of the integration degree and functional complexity of modern electronic devices has led to the need to improve and develop design methods for large and extra-large integrated circuits (LSI and VLSI).

As modern VLSI systems contains millions of semiconductor structures on a chip, the method of element-by-item designing of architecture of computer schemes widely used earlier on ascending methodology cannot provide defect-free designing of difficult systems in strictly established, acceptable terms. Besides, this methodology doesn't provide the possibility of describing program means and taking into account this description while designing. Errors revealed at the upper levels of the representation lead to the necessity to repeat the stages of the design route starting from the level at which the mistakes were made.

To solve these problems in recent decades, a new approach to the design of integrated circuits has appeared - an approach using a top-down design methodology that is free from the listed disadvantages. Top-down design is described on a multi-level hierarchical representation of devices, and development is carried out in accordance with the system hierarchy from a general description of the system to a detailed description of its components

Hardware designing is the process of developing technical documentation, on the basis of which a device (microcircuit, printed circuit board) can be made that meets the specified requirements (technical specifications). Requirements specify the functionality of the device, dimensions, power consumption, operating temperature range and other characteristics. The main type of requirements that make the device meaningful are functional requirements. They determine what the device should do without going into details, how it will do it, and how, by what technology, it will be manufactured. In general terms, design can be considered as a synthesis of a system with a given functionality on a specific elemental base while fulfilling the given restrictions.

The following main stages of the hardware design process are distinguished:

1. Behavioral design (architecture level).
2. Detailed design (level of register transfers).
3. Logical synthesis (level of logical gates).
4. Physical synthesis (transistor level).

The main means of behavioral design (the first stage) are high-level programming languages (C / C ++) and system design languages (SystemC, SystemVerilog). The result of this stage, in addition to clarifying the requirements and creating the appropriate documentation, is a software emulator of the developed equipment. It is important to note that such emulators are used for verification as reference models.

At the second stage, using the hardware description language (Verilog, VHDL), the logical structure and operation of the device are described with maximum accuracy. The hardware description languages are a formal record that can be used at all stages of the development of digital electronic systems. This is due to the fact that language is easily perceived by both machine and man. It can be used at the stages of description, verification, synthesis and testing of equipment, as well as transferring data about the project, its modification and maintenance. A widespread approach used in the construction of equipment is to synchronize subsystems using the so-called clock, which is transmitted to all modules of the device and controls the transfer of signals within and between the modules. This description is called a register transfer level model (RTL). The RTL level is the level of the graphic image of the circuit, quite remote from the lower level of transistors, when the circuit is depicted mainly using registers, triggers, multiplexers, decoders, digital comparators, adders and buffer elements with minimal involvement of the simplest circuits AND, OR, EXCEPT to drawing RTL diagrams.

Modern CAD software for large structured (hierarchical) projects replenishes its RTL schemes with additional blocks that hide the RTL scheme of the module younger in the project hierarchy and displayed as rectangles. In this vein, digital machines are depicted without access to their internal RTL-scheme.

In the third stage, we obtain a circuit of logic gates in a given basis, functionally equivalent to the previously developed RTL-model. Currently, this stage is automated, although manual refinement of the scheme for optimization purposes is not ruled out. The process of deriving a logic circuit from an RTL model is called logical synthesis. If we draw an analogy with software development, logical synthesis corresponds to translating a program from a high-level language into machine code.

At the fourth stage, the topology of the circuit on the chip is placed and traced for a given set of technological limitations (relative position of circuit elements, crystal area, minimum distance between

conductors, conductor size, etc.). The stage of physical design, as well as the preceding stage of logical design, is automated by means of modern CAD systems. The process by which the above tasks are solved is called physical synthesis.

As a result, the efficiency realized using HDL is increased. High level schemes. Management and decision making schemes are often presented in diagram form. The process of writing an HDL description depends on the nature of the chain and the designer's preferences for style coding. HDL is simply "language capture", often using a high-level algorithmic description, such as a C ++ mathematical model. Developers often use a scripting language such as Perl to automatically generate patterns of repeating structures in HDL. Special text editors provide automatic backoff functions that are syntactically dependent on color and macro- on the architecture / declaration of the object / signal.

Verilog Hardware Description Language was developed by Gateway Design Automation in 1984, after the takeover by Cadence, the latter, the language became widespread among developers and became no less popular than VHDL.

Unlike VHDL, the structure and syntax of which resemble such "complex" languages like ADA or ALGOL, Verilog provides more concise and readable syntax typical of the C language, which is very popular among programmers and developers of embedded systems. Verilog allows you to efficiently perform the description and carry out modeling and synthesis of digital circuits due to the presence of developed means for describing devices, the use of built-in primitives and user primitives, time monitoring tools, simulation of propagation delay from input to output, the ability to set external test signals,

HDL Verilog was originally intended for modeling digital systems and began to be used as a means of describing synthesized projects since 1987. Subsequently, this language began to be used for the design of analog circuits. Currently, leading software synthesis system packages based on programmable logic integrated circuits (FPGAs) from companies such as Synopsis, Cadence, Mentor Graphics, support synthesis with descriptions in the Verilog language.

Thus, using HDL can provide several benefits. The description on the HDL can be used as the terms of reference for the project. The advantage of using a formal language, such as Verilog, to describe is that such a description is complete and unambiguous. The description in the formal language is "soft" compared to the "hard" description of the scheme. Representation on HDL allows you to easily process text on any word processor, and database schemes usually require a graphical editor.

The second purpose of using HDL is modeling. Simulation can lead to errors that can only be detected after the manufacture of the equipment. Modeling can be driven by several levels. The system is described using high-level designs. At a logical level, a system is described hierarchically when there are basic blocks at the bottom of the hierarchy. This level may include information about time delays.

A third purpose of using HDL is logic synthesis. There are synthesis tools that can, according to the description of development on HDL, generate an implementation at the gate level from library elements. These tools optimize the project in terms of delay, circuit size, or other objective function. Existing synthesis tools have some limitations, for example, they use only a subset of this language and synthesized schemes cannot be as effective as those created by an experienced developer. However, the synthesis of even parts of the circuit can save time and money, providing the opportunity to develop a simplified version and a preliminary estimate of the speed / area of the crystal.

Moreover, a hardware description language is the best way to document a project. A well-commented HDL description may be better and shorter than a gate-level diagram.

Hereby, the major benefit of the hardware description language is fast design and better verification. The Top-down design and hierarchical design method allows the design time; design cost and design errors to be reduced. Another major advantage is related to complex designs, which can be managed and verified easily. HDL provides the timing information and allows the design to be described in gate level and register transfer level. Reusability of resources is one of the other advantage.

References:

1. Faggin, F. How we made the microprocessor. Nat Electron *1, p. 88, 2018*

2. Digital Systems Design Using Verilog Charles H. Roth, Jr., Lizy Kurian John, and Byeong Kil Lee , Cengage Learning, p. 594, 2014

3. Intel 8080 Microcomputer Systems User's Manual / Intel Corporation. – Santa Clara, Ca, 1975.

4. Yamin Li, Computer Principles and Design in Verilog HDL, John Wiley & Sons, Ltd, p. 572, 2015

5. Steshenko VB. FPGA of Altera firm: element base, design system and hardware description languages. - M .: DMK PRESS, 2015 .-- 573 p.

6. Polyakov A. K. Languages VHDL and Verilog in the design of digital equipment on FPGAs: Uch. pos / A.K. Polyakov, - M ,: Publishing House MPEI, 2012. - 220 p.