COMPUTER AIDED DESIGN (CAD) TOOLS FOR ANALYSIS AND DESIGN OF INTEGRATED CIRCUITS

Genchev I.
Faculty of Electrical Engineering, Electronics and Automation– University of Ruse “Angel Kanchev, Bulgaria
igenchev@uni-ruse.bg

Abstract: The design of electronic integrated circuits (ICs) can be achieved at many different levels from the most detailed layout to the most abstract architectures. This survey presents an overview of recent advances in CAD tools for analog and mixed-signal ICs. Due to the increasing levels of integration available in silicon technology and the growing requirement for digital systems to communicate with the continuous-valued external world, there is a growing need for CAD tools that increase the design productivity and improve the quality of analog ICs. In today’s market, there are plenty of CAD tools, however, most of them are expensive and require high performance platforms. Selecting an appropriate CAD tool for academic use is considered as one of the key challenges in teaching Microelectronics design. In this paper, number of open-source and freeware CAD tools are presented and evaluated. Based on the objectives of the user, this paper furnishes guidelines that help in selecting the most appropriate open-source and freeware CAD tool for teaching. The paper describes the motivation and evolution of these tools and outlines progress on the various design problems involved: simulation and modeling, synthesis and optimization, layout generation and tests.

Keywords: VLSI, CAD Tools, LASI 7, LTSpice

1. Introduction

The microelectronics market are characterized with constantly increasing level of integration complexity, now featuring multimillion transistor ICs. In recent years, complete systems that previously occupied one or more boards have been integrated on a single single-chip. Examples of such systems on a chip (SoC) are the single-chip TV or the single-chip camera or new generations of integrated telecommunication systems that include analog, digital, and eventually radio-frequency (RF) sections on one chip. The technology of choice for these systems is of course CMOS, because of the good digital scaling, but also BiCMOS is used when needed for the analog or RF circuits. Although most functions in such integrated systems are implemented with digital or digital signal processing (DSP) circuitry, the analog circuits needed at the interface between the electronic system and the “real” world are also being integrated on the same die for reasons of cost and performance[1].

The greatest challenge in modern Very Large Scale Integration (VLSI) design is to manage system complexity. Modern SoC designs combine memories, processors and high-speed I/O interfaces on a single chip. The implementation must be divided among large teams of engineers and each engineer must be highly productive. Design proceeds through multiple levels of abstraction, hiding details until they become necessary. The practice of structured design, which is also used in large software projects, uses the principles of hierarchy, regularity, modularity, and locality to manage the complexity [2].

Based on the typical VLSI design work flow, a good VLSI CAD tool must support the following basic features: logical design, circuit schematic design, layout generation, and design check. In today’s market, most VLSI CAD tools are based on Unix or Linux platforms. Only few of them have the ability to run in Windows or run independently of certain emulation/simulation software. The choice of the CAD tools is very critical and concerns the users more and more. Also for educational purposes, an intensive CAD tool is fairly useful for the students to understand the design and concepts of VLSI [3]. In this paper, two open-source VLSI CAD tools base on Windows are presented.

LASI 7 (LAyout System for Individuals) is a "general purpose" layout and design system originally intended for integrated circuits. It is versatile enough that it can be used for ICs, MEMS, discrete devices, schematics, PC boards and project documentation drawings [9]. LTspice is a high performance SPICE simulator, schematic capture and waveform viewer with enhancements and models for easing the simulation of switching regulators [10].

In this work will be overviewed the integrated circuit design and the basic structures, principles and work of LASI7 and LTSpice.

2. Microelectronic Circuits and systems Design

Microelectronic circuit design layout involves the process of creating an accurate physical representation of an engineering drawing (netlist) that conforms to restrictions that are imposed by the manufacturing process, the design flow, and the performance requirements [4]. Fig. 1. [5] Shows the basic structure of ICs design process.

The major stages in the design process are as follows:

![Fig1. Microsystem Design Process](image-url)
1) Specifications and expected performance: This is typically the product conceptualization stage, where the specifications for a design are gathered and the overall product concept is developed. Careful checking of the specifications is crucial for the later success of the product in its application context. Mathematical tools such as Matlab/Simulink are often used at this stage. This stage also includes setting project management goals such as final product cost and time-to-market, project planning, and tracking.

2) Circuit topology: This is the first stage of the actual design, where the overall architecture of the system is designed and partitioned. Hardware and software parts are defined and both are specified in appropriate languages. The hardware components are described at the behavioral level, and, in addition, the interfaces have to be specified. This stage includes decisions about implementation issues, such as package selection, choice of the target technology, and general test strategy.

3) Sizing and Biasing: This stage is the high-level decomposition of the hardware part into an architecture consisting of functional blocks required to realize the specified behavioral description. This includes the partitioning between analog and digital blocks. The specifications of the various blocks that compose the design are defined, and all blocks are described in an appropriate hardware description language. The high-level architecture is then verified against the specifications using behavioral mixed-mode simulations.

4) Schematic capture: For the analog blocks, this is the detailed implementation of the different blocks for the given specifications and in the selected technology process, resulting in a fully sized device-level circuit schematic. The stage encompasses both a selection of the proper circuit topology and a dedicated sizing of the circuit parameters. Throughout this process, more complex analog blocks will be further decomposed into a set of subblocks. Manufacturability considerations (tolerances and mismatches) are taken into account in order to guarantee a high yield and/or robustness. The resulting circuit design is then verified against the specifications using SPICE-type circuit simulations.

5) Physical layout: This stage is the translation of the electrical schematic of the different analog blocks into a geometrical representation in the form of a multilayer layout. This stage involves area optimization to generate layouts that occupy a minimum amount of chip real-estate. The layout is followed by extraction of layout parasitics and detailed circuit-level simulations of the extracted circuit in order to ensure that the performance characteristics do not deviate on account of layout parasitics.

6) Layout vs Schematic: The generation of the system-level layout of an IC not only includes system-level block place and route, but also power-grid routing. Crosstalk and substrate coupling analysis are important in mixed-signal ICs, and proper measures such as shielding or guarding must also be included. Also, the proper test structures are inserted to make the IC testable. Interconnect parasitics are extracted and detailed verification (e.g., timing analysis) is performed. Finally, the system is verified by cosimulating the hardware part with the embedded software.

7) Fabrication and Testing: This is the processing stage where the masks are generated and the ICs fabricated. Testing is performed during and after fabrication in order to reject defective devices.

Note that any of the many simulation and verification stages throughout this design cycle may detect potential problems with the design failing to meet the target requirements. In that case, backtracking or redesign will be needed, as indicated by the upward arrows of Fig. 1.

3. VLSI CAD Tools

As its mention before in today’s market there is a lot of software tools for creating and testing of Microelectronic circuits and systems. Some of the most popular are Cadence - electronic design automation (EDA) software and engineering services company, founded in 1988 by the merger of SDA Systems and ECAD, Inc. For years it had been the largest company in the EDA industry producing software for designing chips and printed circuit boards [11]. Tanner - EDA provides a complete line of software solutions for the design, layout and verification of analog and mixed-signal (A/MS) ICs and MEMS [12]. Verilog - hardware description language (HDL) used to model electronic systems. Verilog HDL is most commonly used in the design, verification, and implementation of digital logic chips at the register-transfer level of abstraction. It is also used in the verification of analog and mixed-signal circuits [13]. All of this software programs are licensed and Linux based. For educational purposes is better to use a free licensed and Windows based CAD tools such as LASI 7 and LTSpice [8].

LASI 7: LASI is shortening for Layout System for Individuals. LASI is a PC-based CAD program used for design of the physical layout of integrated circuits. LASI is used to create and arrange polygons which correspond to the regions (i.e., drain, metal interconnect, etc.) that compose each device (i.e., MOST, resistor, etc.) in IC. LASI is basically a two-dimensional drawing program. The drawing created is called a layout [6].

The program window (fig. 2) is organized in the following manner: (i) Program commands along the top: these provide system functions like "save", "help", "sys", etc. or functions that control the view of the layout, (ii) Drawing commands (or buttons) on the right: for creating and editing the patterns, (iii) Program information along the bottom, (iv) Drawing area in the centre.

![Fig.2. Organization of LASI 7](image)

The construction of a circuit layout is done through a hierarchy of cells. Designers create a number of cells and each cell is assigned a rank between 1 and 15. One cell can contain other cells of lower rank, but cannot contain cells of equal or higher rank. For example, suppose that the goal is to construct OP AMP. One way to proceed would be to start by defining two rank 1 cells: a PMOS and an NMOS transistor. Then you could define a rank 2 cell — a current mirror that would be built up using the PMOS and NMOS transistor cells. The next logical progression would be a rank 3 cell, an OP AMP that would be built up of a collection of rank 1 & 2 cells. The basic idea is to construct a set of building blocks that can be used over and over again in the process of putting together the circuit. This is how digital designers are able to build such huge ICs.

Each of the fab processes has design rules that go along with it. The design rules specify the minimum that can be used for some of the features or the minimum separations between features and is
Creating an NMOS cells with LASI

1) When LASI is started the program will present a dialog window where can be named a new cell or open an existing one. Since NMOS transistor will be drawn, the cell will be named "nMOS_4_8x1_6" (transistor has a W/L ratio of 4.8μm/1.6μm) and set the rank to 1, then set the λ to be 0.8μm.

2) Now will be defined the source and drain regions. Select the Layr command on Menu 1, and from the layer table check the active layer ACTV and hit OK. The n+ active region is drown in the drawing area. Usually, the patterns will be in the form of rectangular boxes. The ACTV layer should be green in color and with size of 14.4μm (18λ) on x-direction and 4.8μm (4λ) on y-dimension, fig.3 A).

3) Now will be define the gate for the NMOS. Again, select the Layr command, but this time choose POL1 from the layer table. (Poly stands for polysilicon, which is the material that modern MOSFETs use for the gate). The POL1 patterns region can have a minimum dimension of 2λ, as its mention before meaning that the minimum drawn gate length L for the transistors will be 2λ. Also, the poly gate must overlap the sides of the active region by at least 2λ, and must be at least 3λ from ends of the active region. Here make the gate a narrow vertical strip centered in the active region, with the narrow dimension of L = 2λ = 1.6 mm, the POL1 layer is red in color, fig.3 B).

4) Next will be define the active layer to be n+ not p+. Choosing Layr and pick the n-select layer NSEL. Then Add the NSEL box around the outside of the active region, spaced at a distance of 2λ, everywhere. The NSEL layer in unfilled green pattern. The select region must overlap the active region by ≥ 2λ, fig.3 C).

5) Now will be define contact regions. The first thing to do is to cut contact holes through the oxide that covers the source, drain, and gate. Selecting the Layr command, and this time choose the contact layer CONT from the pop-up menu. Drawing the contact holes within the source and drain regions on each side of the gate, and one contact hole below the gate by 1λ, as shown. The contact holes must be exactly 2λ x 2λ. The CONT regions are solid yellow, fig.3 D).

6) Next will be added metal over the contact holes. Using the Layr command and choose the metal 1 layer MET1 from the layer table. Then draw metal 1 over the gate, drain, and source contacts. Metal 1 should completely overlap the contact holes by at least 1λ on each side. The minimum width of a metal line is 3λ, and there must be at least 3λ of separation between metal lines. The metal 1 is cross-hatched blue in color, fig.3 E).

7) Now will be added a body contact (i.e., the substrate terminal) for the NMOST. In order to make a good electrical contact to the p-type substrate, first will be create a small p+ region into the substrate. We use the p-select layer PSEL to tell LASI that the active region will be p+. And then the following is done:

Layr -> ACTV, Add the active region shown to the left of the n-select region.
Layr -> PSEL, Add the p-select region around the active region. The PSEL is a yellow open rectangle.
Layr -> CONT, Add the contact for the body connection.
Layr -> MET1, Add, the metal 1 region over the contact.

The body contacts must follow the same design rules as the contacts to the active regions. Note that is possible to butt the PSEL region up against the NSEL region, as long as the active regions enclosed by each are at least 4λ apart, Fig.3 F).

At this point, the cell is done, and it can be saved.

LTSpice: Spice is used extensively in education and research to simulate analog circuits. This powerful tool can help to avoid assembling circuits which have very little hope of operating in practice through prior computer simulation. The circuits are described using a simple circuit description language which is composed of components with terminals attached to particular nodes. These groups of components attached to nodes are called netlists [7].

A Spice netlist is usually organized into different parts. The very first line is ignored by the Spice simulator and becomes the title of the simulation. The rest of the lines can be somewhat scattered assuming the correct conventions are used. For commands, each line must start with a '-' (Period). For components, each line must start with a letter which represents the component type (eg. ‘M’ for MOSFET). When a command or component description is continued on multiple lines, a '+' (plus) begins each following line so that Spice knows it belongs to whatever is on the previous line.
Any line to be ignored is either left blank, or starts with an asterisk (*).

Here is a Spice netlist of the transistor that I discuss above, connected with a power supply input voltage and resistor ready for simulation:

```
Spice Simulation 1-1
*** MODEL Descriptions ***
.model nm nMOS_4_8x1_6 level=2 VT0=0.7
KP=80e-6 LAMBDA=0.01
*** NETLIST Description ***
M1 vdd ng 0 0 nm W=4.8u L=1.6u
R1 in ng 50
Vin in 0 2.5
*** SIMULATION Commands ***
.dc Vdd 0 10 0.2 Vin 1 5 1
.end
```

In Fig. 5 is shown the schematic of the transistor connected with resistor and supply voltage along with the A/V output characteristic.

![Schematic and A/V output characteristic](image)

4. Conclusions

VLSI design has become an important course at most of the electrical and computer engineering programs. However, buying licenses for commercial VLSI CAD is usually costly and requires high performance workstations which many academic institutes may not be able to afford. This paper provided some insight on some of the most popular open-source CAD tools that can be used in the academic field.

In this work was presented a brief overview of the basics steps of the integrated circuit design process. It was done layout design of the integrated transistor using free-based CAD tool LASI 7. It was created a Net list of the transistor and it was made simulations using LTSpice simulation tool.

REFERENCES