

Chapter 2

DESIGN FLOW OVERVIEW

2.1 Design Levels

Functionality and architecture of electronic devices can be very complex. The systems may consist of analog and digital hardware together with software parts. A telecommunication system contains for example:

- An analog front-end to the physical transmission channel
- Digital hardware for coding and modulation
- General purpose or signal processors for control, user interface and transmission protocol handling

Many designers with specialization in different areas are involved in design and implementation. Several design steps are necessary to realize a system concept on silicon. The design process can be classified in several design levels as shown in Figure 2.1.

Each design level is associated with certain design tasks concerning the whole system or system parts. Starting from system level the design description becomes more and more detailed in a design step. CAD tools support the designer at each level.

The system level is the first design level beginning with an idea of the desired system. This level is also called concept engineering. The system concept and main algorithms are described at a very abstract level without information about the implementation of algorithms. For example, the coding algorithm to be used for data transmission is specified, but it is not decided to implement the coder in hardware or software.

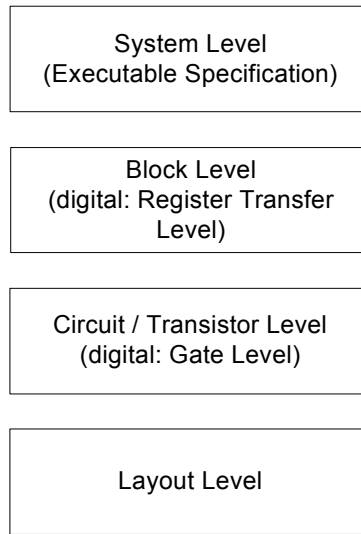


Figure 2-1. Design levels

The system specification can be developed on a sheet of paper. More powerful is an executable specification supported by system-level simulators (for example CoCentric System Studio, MATLAB, and SPW). It allows the evaluation of the selected algorithms and provides a reference model for following design steps.

The system is now partitioned into several hardware (analog or digital) and software subsystems. This design level is named Block Level or Register Transfer Level (RTL) in the digital area. The description of the subsystems at this level contains more detail about the design architecture. At this level the design consists of different blocks, for example multiplier, adder, register, A/D converter, analog filter and amplifier.

Digital and mixed-signal hardware description language (HDL) simulators support the block level design. Commonly used modeling languages in this area are VHDL-AMS and Verilog-AMS. The design of hardware/software systems is further supported by special tools, for example instruction set simulators (ISS).

The third design level is called gate level in the digital domain and circuit level in the analog domain. The blocks of the system are now represented by netlists containing gates or active and passive analog elements. Gate level models can be generated from RTL descriptions by logic synthesis. In the analog design, the circuits are still designed manually.

Gate level or circuit simulation is used to evaluate the design at block level. In the digital domain a timing analysis can be executed, and the blocks

are still described in VHDL and Verilog. Circuit simulators such as SPICE and Spectre are used in the analog domain to analyze the behavior of the designed block.

Based on the gate level or circuit netlist and data of the circuit technology the layout of the circuit is designed. The design is now represented as polygons at different layers of an integrated circuit. In the digital domain this step is well-automated. The tools will check if the design rules for a specified circuit technology are fulfilled. In the analog domain further manual optimization of layout may be necessary, for example to minimize crosstalk between signals or to achieve a symmetric design. Tools that extract parasitic effects that originate from layout also support the layout verification.

2.2 Top-down System Design

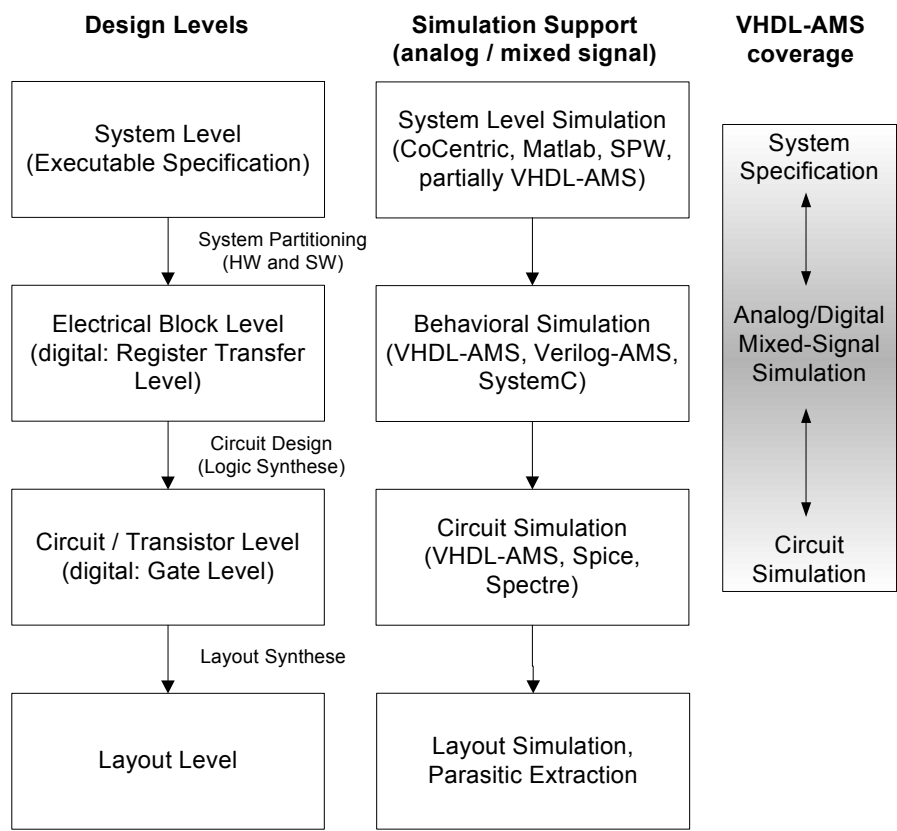


Figure 2-2. Top-down design and simulation support

Top-down design is a method of designing an electronic system that starts with the complete system concept and then breaks it down into smaller and smaller components (see Figure 2-2).

The first design level at which top down design starts is the system level. For telecommunication systems it is here that is specified which algorithms are used to transmit data from the signal source at point A to a sink at point B. Algorithms which are specified at this level may be for example:

- data structure and protocol
- forward error correction techniques (FEC)
- modulation techniques (QPSK, QAM, GMSK, OFDM)
- channel equalization and synchronization

The system level design is supported by system level simulation. Efficient simulation techniques (for example event driven or data stream driven simulation) allow the simulation of the complete transmission system. The simulation also includes a model of the transmission channel (additive white Gaussian noise, AWGN, or mobile channels with fading). The goal of the system design is an overall system specification. If a system level simulation model exists, it can be used as an "executable specification" (see Figure 2-3).

If the system level specification was successfully verified within a system level simulation the system is partitioned. The algorithms of the system can be implemented in different ways:

- analog hardware
- digital hardware
- software

The second design level is named Block Level or in the digital area Register Transfer Level. The system is now partitioned into components and subsystems. Now parameters of the components can be specified.

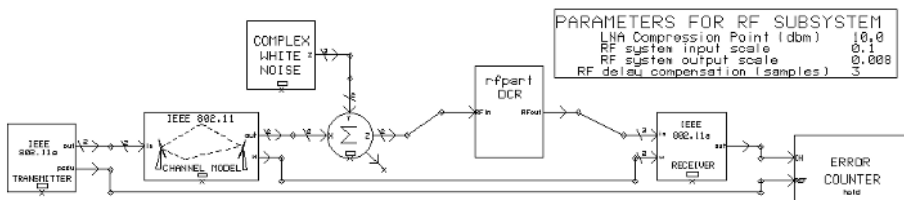


Figure 2-3. Top level schematic of a WLAN system simulation model (SPW)

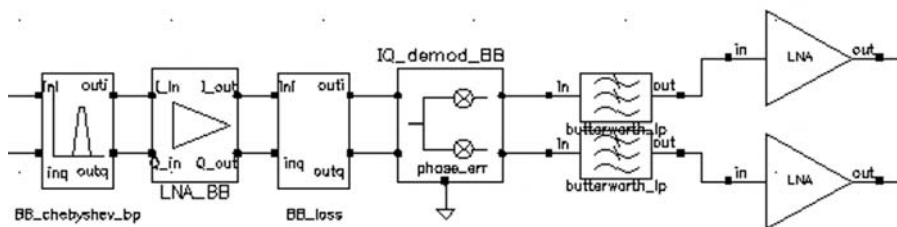


Figure 2-4. Schematic of the RF subsystem (direct conversion receiver)

Figure 2-4 shows for example the block level schematic of the RF subsystem of the WLAN receiver. At system level the RF subsystem was specified either with ideal parameters or with parameters like noise level, gain and linearity. Now it is broken down into its components (filter, amplifier and mixers) which must be parameterized.

At block level we use behavioral models for the simulation of the subsystems. For the analog and mixed-signal area, models can be written in VHDL-AMS and Verilog-AMS. For pure analog simulation, additional languages (for example SpectreHDL) are provided with the simulation tools. The simulation at block level is used to verify whether the block level realization of the subsystem meets the system level requirements.

After the blocks are specified, the circuit design can start. In the digital area, gate level designs can be generated automatically from behavioral models. However for analog blocks there are still no synthesis tools available. So the analog designers must create the transistor level implementation of the components manually. This is supported by transistor level simulation. The block level simulation models can be reused as testbench or reference models if the circuit level simulator supports behavioral modeling languages. Verilog-AMS and VHDL-AMS simulators often support the simulation of SPICE netlists; therefore they can also be used for verification of the transistor level design.

If the transistor level design was verified by simulation the layout can be developed. With the layout level the top down design flow is finished. The layout design is not within the scope of this book. It is possible to extract parasitic effects from layout level simulation which can be used to improve the accuracy of transistor level simulation.

2.3 Bottom-up Verification

The amount of information and number of parameters increases during the top-down design process from the system concept to its implementation.

At the beginning of the design, the system is described with some algorithms. After implementation the system may consist of a large number of transistors. Concept verification is needed to check that the implementation meets the requirements of the system.

In the “V” diagram (Figure 2-5) the verification starts from the layout level (bottom) and then proceeds up to the block and system levels.

After layout, simulation parasitic effects can be back-annotated into the circuit netlist. The circuit simulation with the extracted netlist is used to verify the circuit design. The designed circuits can now be combined into functional blocks, which are checked against their specification in a block level simulation. Finally the designed blocks can be connected to the system. System level simulation verifies that the blocks fit into the system environment.

It is recommended to start verification before the design is completed at layout level. After each design step simulation can be used to verify the design or component against the specification.

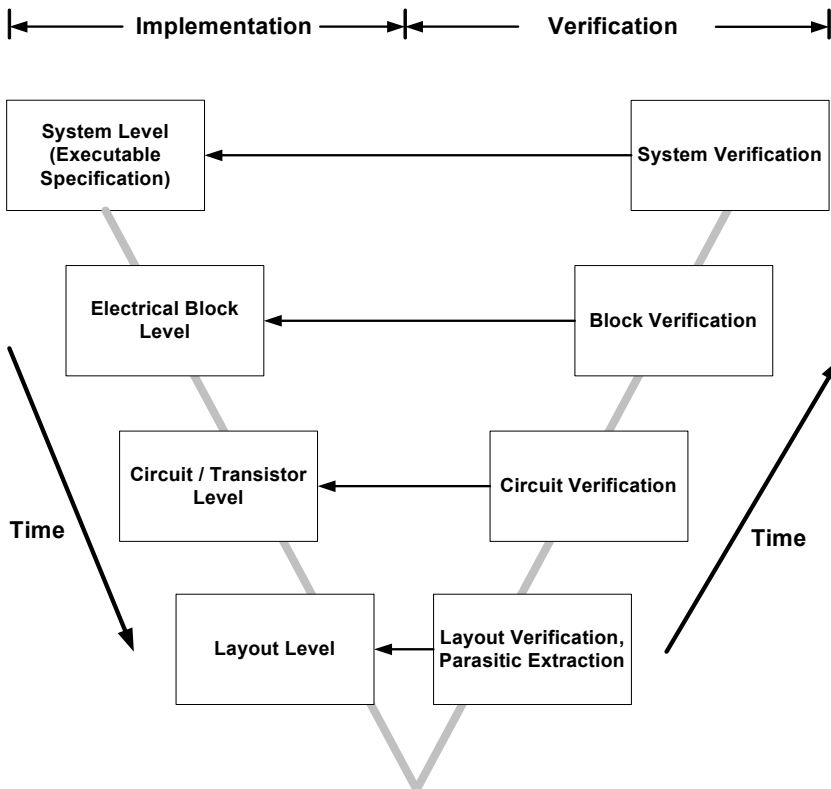


Figure 2-5. Top -down design and bottom-up verification (V diagram)

System level or block level simulation is used to verify large systems or circuits. Often a transistor level model of a system cannot be simulated because its complexity (number of transistors or gates) is much too large. Therefore it is necessary to use behavioral models.

Figure 2-6 shows the application of behavioral models during block level and system level verification. It is assumed that behavioral models were already used during the top-down design. In the verification phase it is now necessary to calibrate these models as follows:

- Parasitic extraction and back annotation into the circuit netlist improves the accuracy of the circuit model (extracted circuit model)
- Simulation with the extracted circuit model is used to gain the circuit characteristic and parameters
- Extracted circuit parameters are used to calibrate the behavioral model of this component
- Calibrated behavioral models are used on block and system levels for verification

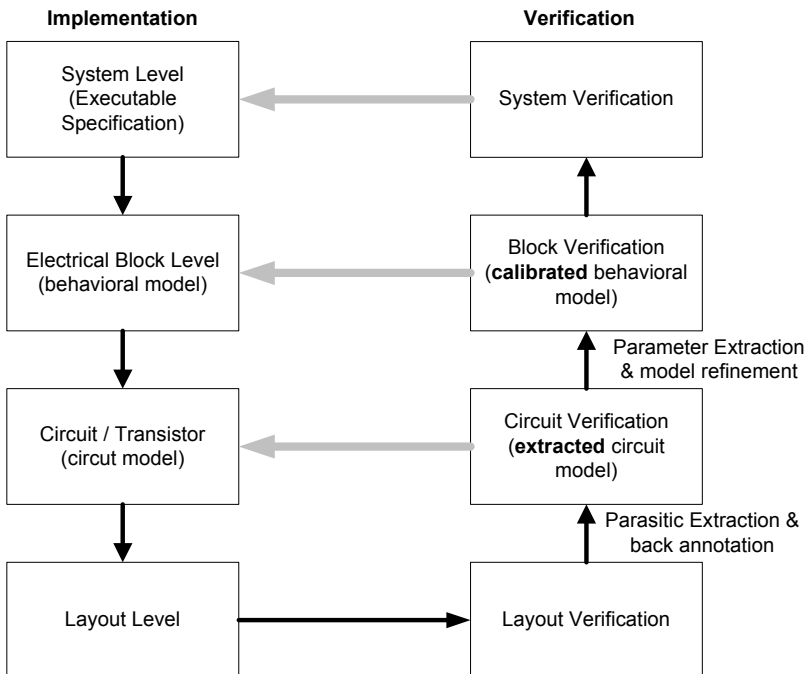


Figure 2-6. Refinement of models during bottom-up verification

The main advantage of using (calibrated) behavioral models is the simulation speedup which enables the simulation of large systems or subsystems.

Different behavioral modeling languages exist. Most of them are specific to a particular simulator. To allow the reuse of models it is suggested to use standardized languages like VHDL-AMS and Verilog-AMS.

A characterization environment can support model calibration. Characterization is the calculation of component or subsystem characteristics and parameters from measured or simulated data. A characterization run contains a set of simulation and postprocessing commands that allow the determination of significant circuit characteristics. The behavior of the circuit description and behavioral model can be compared. If the model is inaccurate, the model parameters or algorithms are modified. Characterization also supports model and circuit documentation. Chapter 11 contains more information about characterization environments.

Modeling and Simulation for RF System Design

Frevert, R.; Haase, J.; Jancke, R.; Knochel, U.; Schwarz,
P.; Kakerow, R.; Darianian, M.

2005, XI, 291 p., Hardcover

ISBN: 978-0-387-27584-0