

WITH THE UDM, EVERYONE'S ON THE SAME PAGE.

The Universal Design Methodology— taking hardware from conception through production

ALTHOUGH STEPS of the design process may differ for different types of hardware, the basic design procedures remain the same. Rarely, however, do companies detail an entire process in a written document or a set of procedures. Most companies divide the process among departments, and the process resides sometimes only in the brains of select individuals. These conditions make planning and scheduling difficult and lead to poor communication and, thus, false expectations among engineers. The UDM (Universal Design Methodology) is an approach to hardware design with its foundation in years of experience designing ASICs, FPGAs, CPLDs, and pc boards for large and small companies in different industries. It lays out the entire process, guaranteeing that you will be able to accurately schedule the design and correctly allocate resources and preventing you from overlooking any steps.

The UDM aims to help you design hardware that is free of manufacturing defects, works reliably over the lifetime of the device, and functions correctly in the system. It helps you efficiently design this hardware—meaning in the least amount of time and using the fewest resources, including personnel. It also helps you efficiently plan the design—meaning that you create a reasonable schedule as early in the process as possible and determine all necessary resources up-front, so that you can allocate them as early in the process as possible.

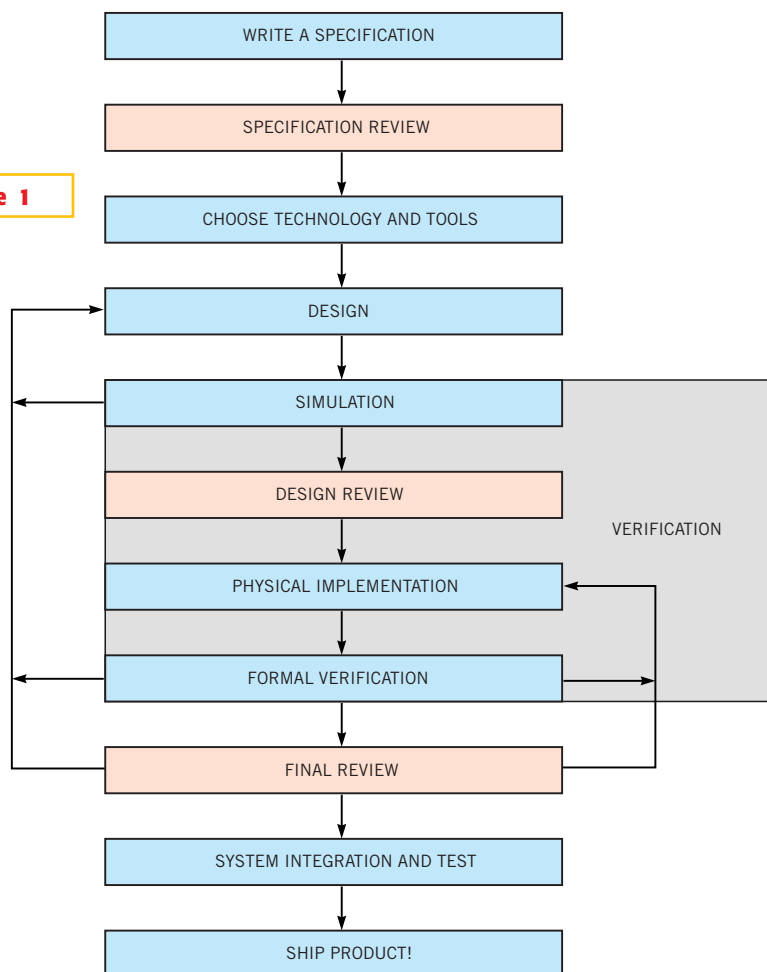
The UDM specifies steps that enable you to reach these UDM design goals (Figure 1). The specifics of each step vary with each design, but the steps essentially remain the same for all designs. You should understand each step before the project begins. Note that the entire process is not linear but consists of iterations. At various steps of the process, you discover bugs in the design and problems with the specification. You need to fix them and then backtrack in the process.

WRITING AND REVIEWING A SPECIFICATION

The importance of a specification cannot be overstated. A specification is an absolute must, especial-

ly as a guide for choosing the right technology and tools and for making your needs known to the various vendors with which you will be working. It allows each engineer to understand the entire design and his or her piece of it and to design the correct

Figure 1



Different devices require slightly different steps, but the basic methodology of the UDM remains the same.

interfaces to the rest of the system. It also saves time and, thus, cost and helps you avoid misunderstanding. There is no excuse for not having a written specification.

A specification should include:

- an external block diagram showing how the chip fits into the system;
- an internal block diagram showing each major functional section;
- a description of the I/O pins, including output-drive capability and input-threshold levels (CMOS, TTL, ECL, and others);
- timing estimates including, setup-and-hold times for input pins, propagation times for output pins, and clock-cycle time;
- logic estimates (gate or chip counts);
- physical specifications (package type, physical size, connector requirements, and others);
- a power-consumption target;
- a price target; and
- test procedures.

After writing the specification, it is important that you schedule a review. As many people as are interested should take part in reviewing the specification, because it is the basis of the entire design. The group needs to determine whether anything is wrong with or missing from the specification. Invite people from all the departments involved with the final product, including marketing, sales, software, and applications. The specification will be the guide for the rest of the project.

Note that the specification includes the testing procedures. Too often, designers make the mistake of leaving testing for late in the design process. You should understand and specify the test procedures early. In this way, you can build testing structures into the design. You can also develop any software necessary for testing in parallel with the hardware development, and you can fix any testing problems that arise before you finalize the hardware.

It is important to understand that a specification is a living document. Many sections include only best guesses, but these guesses become more accurate as you design the hardware. Engineering problems arise that require compromises in function. Market changes may require changes to the specification. All de-

UDM DOES NOT LIMIT OR DICTATE THE DESIGN PRACTICES THAT YOU USE. IT IS IMPORTANT THAT YOU USE RELIABLE, ACCEPTED DESIGN PRACTICES DURING THIS STEP OF THE PROCESS.

isions about functionality must use the specification as an up-to-date reference, and you must enter all subsequent changes into the specification. With a complete and accurate specification, you immediately know the repercussions of any changes. Make sure that you immediately distribute the revised specification to the entire design team.

CHOOSING TECHNOLOGY AND TOOLS

Once you have written a specification, you can use it to find the vendors with the technology and pricing that best meet your requirements. You can then choose tools that work well together and with the device and technology that you have chosen.

If your project is an ASIC, you use the specification to decide which vendors, semiconductor processes (CMOS, TTL, and others), architectures (gate array or standard cell), and process sizes (1 micron, 0.8 micron, and others) meet your requirements. If your project is a programmable device, you use the specification to determine the vendor, the type of device (CPLD or FPGA, for example), and the technology (antifuse, SRAM, or flash, for example). If your project is a pc board, you use the specification to determine the board size, number of layers, materials, routing widths, and vendor. If you are designing a system, the specification determines all of its components and the vendors that manufacture them.

The design step of UDM depends entirely on the kind of device you are designing. Digital hardware differs from analog hardware; ASIC design differs from pc-board design. UDM does not limit or dictate the design practices that you use. It is important that you use re-

liable, accepted design practices during this step of the process.

VERIFICATION

Verification is a “super step” because it comprises several other steps of the UDM. The exact steps that make up verification are open to argument and vary according to the type of device that you are designing, but you can generally break verification into the following steps, each of which is essential to the entire process:

- simulation,
- design review,
- synthesis,
- physical implementation, and
- formal verification.

Simulation is an ongoing process throughout the design. You should simulate small sections of the design separately before hooking them up to larger sections. Obtaining the correct functionality requires much iteration of design and simulation. Simulation must include a functional simulation but may also include simulation of power consumption, timing, and other parameters.

Once you finish the design and simulation, you need to schedule another design review. It is important to include other engineers—even those who are not involved in the design—to look over the simulations and make sure that no one missed anything or made any improper assumptions. This review is one of the most important, because only with correct and complete simulation can you know that your chip will work correctly in your system. Outside engineers can raise issues that you and your team may have missed, challenging your assumptions and discovering corner cases that were not already simulated and often reveal problems in the design. Although this exercise can be confrontational, if you manage it correctly, it can uncover many potential problems.

Note that other design reviews must occur throughout the design process. Some must be somewhat formal, with documents distributed to people and a meeting to discuss issues and implement changes. Other reviews can be less formal and may take place in small groups.

For an ASIC, physical implementation involves synthesis and placement and routing, resulting in a physical layout and a set of masks for production. For CPLDs

and FPGAs, this step also involves synthesis and placement and routing, but it results in a pattern of bits used to program the device. For a pc board, this stage involves creating a netlist and producing a layout of the layers of the board.

During formal verification, you establish whether the physical implementation is functionally equivalent to the fully simulated original design. In some cases, you perform formal verification by resimulating the physical implementation and comparing the results with the original simulation. Some formal verification tools compare the physical implementation with the original design and mathematically determine whether they are equivalent. It is important at this stage to verify power consumption, timing, and any other critical parameters against the specification.

The final review of the design should be a formality. If you follow all of the other steps and conduct all of the other reviews, final review is a simple sign-off

that verifies that the design has been created, simulated, physically implemented, and formally verified and that it is now ready to be manufactured.

SYSTEM INTEGRATION AND TEST

At the step for system integration and test, you are responsible for determining that the entire system, including the device that you have designed, works correctly. If you have followed UDM procedure up to this point, chances are good that your system will perform correctly with only minor problems. These minor problems often require only slight modifications to the system or changes to the system software. You need to test and document these problems so that you can fix them on the next revision of the device. System integration and testing is necessary to ensure that all parts of the system work correctly together.

When the device enters production, you need to perform a burn-in test of your system that continually tests your system

over a long period of time, preferably at the extreme low and high temperatures that it might encounter in the field. A correctly designed device will fail only because of the marginal physical defects that this kind of stress testing usually reveals.

The end of the UDM is the time to ship your product and take that long-awaited vacation. Let the salespeople worry about the rest. □

AUTHOR'S BIOGRAPHY

*Bob Zeidman is president of Zeidman Consulting (www.zeidmanconsulting.com), an EDA company that offers contract design of hardware and software. He is also the author of *Designing with FPGAs and CPLDs, which offers information about the UDM and programmable devices*; *Verilog Designer's Library*; and *Introduction to Verilog*. He holds an MSEE degree from Stanford University (Stanford, CA) and a BSEE and a BA in physics from Cornell University (Ithaca, NY). You can reach him at bob@zeidmanconsulting.com.*
