DOWNLOAD PDF CUSTOM IC DESIGN FLOW

Chapter 1 : VLSI Basics: VLSI design flow

Design kits include all the foundry-specific devices and models for use with Pyxis Custom IC Design solutions. The platform supports all common commercial design kit formats plus numerous customer proprietary formats.

An early successful commercial application was the gate array circuitry found in the low-end 8-bit ZX81 and ZX Spectrum personal computers , introduced in and Customization occurred by varying a metal interconnect mask. Gate arrays had complexities of up to a few thousand gates; this is now called mid-scale integration. Later versions became more generalized, with different base dies customised by both metal and polysilicon layers. Some base dies include RAM elements. Standard cell In the mids, a designer would choose an ASIC manufacturer and implement their design using the design tools available from the manufacturer. While third-party design tools were available, there was not an effective link from the third-party design tools to the layout and actual semiconductor process performance characteristics of the various ASIC manufacturers. Most designers used factory-specific tools to complete the implementation of their designs. A solution to this problem, which also yielded a much higher density device, was the implementation of standard cells. Every ASIC manufacturer could create functional blocks with known electrical characteristics, such as propagation delay , capacitance and inductance, that could also be represented in third-party tools. Standard-cell design is the utilization of these functional blocks to achieve very high gate density and good electrical performance. By the late s, logic synthesis tools became available. Such tools could compile HDL descriptions into a gate-level netlist. Standard-cell integrated circuits ICs are designed in the following conceptual stages referred to as electronics design flow , although these stages overlap significantly in practice: A team of design engineers starts with a non-formal understanding of the required functions for a new ASIC, usually derived from requirements analysis. Register-transfer level RTL design: The design team constructs a description of an ASIC to achieve these goals using a hardware description language. This process is similar to writing a computer program in a high-level language. Suitability for purpose is verified by functional verification. This may include such techniques as logic simulation through test benches , formal verification , emulation , or creating and evaluating an equivalent pure software model, as in Simics. Each verification technique has advantages and disadvantages, and most often several methods are used together for ASIC verification. Logic synthesis transforms the RTL design into a large collection called of lower-level constructs called standard cells. These constructs are taken from a standard-cell library consisting of pre-characterized collections of logic gates performing specific functions. The standard cells are typically specific to the planned manufacturer of the ASIC. The resulting collection of standard cells and the needed electrical connections between them is called a gate-level netlist. The gate-level netlist is next processed by a placement tool which places the standard cells onto a region of an integrated circuit die representing the final ASIC. The placement tool attempts to find an optimized placement of the standard cells, subject to a variety of specified constraints. An electronics routing tool takes the physical placement of the standard cells and uses the netlist to create the electrical connections between them. Placement and routing are closely interrelated and are collectively called place and route in electronics design. Given the final layout, circuit extraction computes the parasitic resistances and capacitances. In the case of a digital circuit , this will then be further mapped into delay information from which the circuit performance can be estimated, usually by static timing analysis. This, and other final tests such as design rule checking and power analysis collectively called signoff are intended to ensure that the device will function correctly over all extremes of the process, voltage and temperature. When this testing is complete the photomask information is released for chip fabrication. These steps, implemented with a level of skill common in the industry, almost always produce a final device that correctly implements the original design, unless flaws are later introduced by the physical fabrication process. Standard cells produce a design density that is cost effective, and they can also integrate IP cores and static random-access memory SRAM effectively, unlike gate arrays. Gate-array and semi-custom design[edit] Microscope photograph of a gate-array ASIC showing the predefined logic cells and custom interconnections. Gate array design is a manufacturing method in which diffused layers, each consisting transistors and other active devices

, are predefined and electronics wafers containing such devices are "held in stock" or unconnected prior to the metallization stage of the fabrication process. The physical design process defines the interconnections of these layers for the final device. For most ASIC manufacturers, this consists of between two to nine metal layers with each layer running perpendicular to the one below it. Non-recurring engineering costs are much lower than full custom designs, as photolithographic masks are required only for the metal layers. Production cycles are much shorter, as metallization is a comparatively quick process; thereby accelerating time to market. Often difficulties in routing the interconnect require migration onto a larger array device with consequent increase in the piece part price. These difficulties are often a result of the layout EDA software used to develop the interconnect. Pure, logic-only gate-array design is rarely implemented by circuit designers today, having been almost entirely replaced by field-programmable devices. Most prominent of such devices are field-programmable gate arrays FPGAs which can be programmed by the user and thus offer minimal tooling charges non-recurring engineering, only marginally increased piece part cost, and comparable performance. Today, gate arrays are evolving into structured ASICs that consist of a large IP core like a CPU , digital signal processor units, peripherals , standard interfaces , integrated memories , SRAM , and a block of reconfigurable , uncommitted logic. This shift is largely because ASIC devices are capable of integrating large blocks of system functionality and systems on a chip SoCs require glue logic , communications subsystems such as networks on chip , peripherals and other components rather than only functional units and basic interconnection. In their frequent usages in the field, the terms "gate array" and "semi-custom" are synonymous when referring to ASICs. Process engineers more commonly use the term "semi-custom", while "gate-array" is more commonly used by logic or gate-level designers. Full custom Microscope photograph of custom ASIC chipset showing gate-based design on top and custom circuitry on bottom By contrast, full-custom ASIC design defines all the photolithographic layers of the device. Full-custom design is used for both ASIC design and for standard product design. The benefits of full-custom design include reduced area and therefore recurring component cost , performance improvements, and also the ability to integrate analog components and other pre-designed â€”and thus fully verifiedâ€”components, such as microprocessor cores, that form a system-on-chip. The disadvantages of full-custom design can include increased manufacturing and design time, increased non-recurring engineering costs, more complexity in the computer-aided design CAD and electronic design automation systems, and a much higher skill requirement on the part of the design team. Automated layout tools are quick and easy to use and also offer the possibility to "hand-tweak" or manually optimize any performance-limiting aspect of the design. This is designed by using basic logic gates, circuits or layout specially for a design. However, the basic premise of a structured ASIC is that both manufacturing cycle time and design cycle time are reduced compared to cell-based ASIC, by virtue of there being pre-defined metal layers thus reducing manufacturing time and pre-characterization of what is on the silicon thus reducing design cycle time. Design differentiation and customization is achieved by creating custom metal layers that create custom connections between predefined lower-layer logic elements. Because only a small number of chip layers must be custom-produced, "structured ASIC" designs have much smaller non-recurring expenditures NRE than "standard-cell" or "full-custom" chips, which require that a full mask set be produced for every design. What makes a structured ASIC different is that in a gate array, the predefined metal layers serve to make manufacturing turnaround faster. In a structured ASIC, the use of predefined metallization is primarily to reduce cost of the mask sets as well as making the design cycle time significantly shorter. For example, in a cell-based or gate-array design the user must often design power, clock, and test structures themselves; these are predefined in most structured ASICs and therefore can save time and expense for the designer compared to gate-array. Likewise, the design tools used for structured ASIC can be substantially lower cost and easier faster to use than cell-based tools, because they do not have to perform all the functions that cell-based tools do. In some cases, the structured ASIC vendor requires that customized tools for their device e. Cell libraries, IP-based design, hard and soft macros[edit] Cell libraries of logical primitives are usually provided by the device manufacturer as part of the service. Although they will incur no additional cost, their release will be covered by the terms of a non-disclosure agreement NDA and they will be regarded as intellectual property by the manufacturer. Usually their physical design will be pre-defined so they

could be termed "hard macros". What most engineers understand as " intellectual property " are IP cores , designs purchased from a third-party as sub-components of a larger ASIC. Many organizations now sell such pre-designed cores â€" CPUs, Ethernet, USB or telephone interfaces â€" and larger organizations may have an entire department or division to produce cores for the rest of the organization. Indeed, the wide range of functions now available is a result of the phenomenal improvement in electronics in the late s and early s; as a core takes a lot of time and investment to create, its re-use and further development cuts product cycle times dramatically and creates better products. Additionally, organizations such as OpenCores are collecting free IP cores, paralleling the open-source software movement in hardware design. Soft macros are often process-independent i. Hard macros are process-limited and usually further design effort must be invested to migrate port to a different process or manufacturer. Multi-project wafers[edit] Some manufacturers offer multi-project wafers MPW as a method of obtaining low cost prototypes. Often called shuttles, these MPW, containing several designs, run at regular, scheduled intervals on a "cut and go" basis, usually with very little liability on the part of the manufacturer. The contract involves the assembly and packaging of a handful of devices. The service usually involves the supply of a physical design database i. The manufacturer is often referred to as a "silicon foundry" due to the low involvement it has in the process. Application-specific standard product[edit] An application specific standard product or ASSP is an integrated circuit that implements a specific function that appeals to a wide market. ASSPs are used in all industries, from automotive to communications. Both of these examples are specific to an application which is typical of an ASIC but are sold to many different system vendors which is typical of standard parts.

DOWNLOAD PDF CUSTOM IC DESIGN FLOW

Chapter 2 : Cadence Tutorial

Custom IC / Analog / RF Design Custom IC / Analog/ RF Design Overview Cadence Â® custom, analog, and RF design solutions can help you save time by automating many routine tasks, from block-level and mixed-signal simulation to routing and library characterization.

Each and every step of the ASIC design flow has a dedicated EDA tool that covers all the aspects related to the specific task perfectly. And most importantly, all the EDA tools can import and export the different file types to help making a flexible ASIC design flow that uses multiple tools from different vendors. ASIC design flow is not exactly a push button process. To succeed in the ASIC design flow process, one must have: This article covers the ASIC design flow in very high level. We will provide a more detailed articles in the future explaining more about the activities within each phase. In ASIC system design phase, the entire chip functionality is broken down to small pieces with clear understanding about the block implementation. Some other large blocks need to be divided into subsystems and the relationship between the various blocks has to be defined. In this phase the working environment is documentation. In addition to the digital implementation, a functional verification is performed to ensure the RTL design is done according to the specifications. When all the blocks are implemented and verified the RTL is then converted into a gate level netlist. Synthesis In this phase the hardware description RTL is converted to a gate level netlist. This process is performed by a synthesis tool that takes a standard cell library, constraints and the RTL code and produces an gate-level netlist. Synthesis tools are running different implementations to provide best gate level netlist that meets the constraints. It takes into account power, speed, size and therefore the results can vary much from each other. To verify whether the synthesis tool has correctly generated the gate-level netlist a verification should be done. Layout In this stage, the gate level netlist is converted to a complete physical geometric representation. Then placement of physical elements within each block and integration of analog blocks or external IP cores is performed. When all the elements are placed, a global and detailed routing is running to connect all the elements together. Also after this phase a complete simulation is required to ensure the layout phase is properly done. The layout should be done according the silicon foundry design rules.

Chapter 3 : Application-specific integrated circuit - Wikipedia

Custom IC Design Blogs. KomalJohar This feature brings in efficiency to your design process flow by ensuring that your design data is aligned to the selected MPT.

A typical design cycle may be represented by the flow chart shown in Figure. Our emphasis is on the physical design step of the VLSI design cycle. However, to gain a global perspective, we briefly outline all the steps of the VLSI design cycle. The first step of any design process is to lay down the specifications of the system. System specification is a high level representation of the system. The factors to be considered in this process include: The fabrication technology and design techniques are also considered. The specification of a system is a compromise between market requirements, technology and economical viability. The end results are specifications for the size, speed, power, and functionality of the VLSI system. The basic architecture of the system is designed in this step. While MAS is a textual English like description, architects can accurately predict the performance, power and die size of the design based on such a description. Behavioral or Functional Design: In this step, main functional units of the system are identified. This also identifies the interconnect requirements between the units. The area, power, and other parameters of each unit are estimated. The behavioral aspects of the system are considered without implementation specific information. For example, it may specify that a multiplication is required, but exactly in which mode such multiplication may be executed is not specified. We may use a variety of multiplication hardware depending on the speed and word size requirements. The key idea is to specify behavior, in terms of input, output and timing of each unit, without specifying its internal structure. The outcome of functional design is usually a timing diagram or other relationships between units. This information leads to improvement of the overall design process and reduction of the complexity of subsequent phases. Functional or behavioral design provides quick emulation of the system and allows fast debugging of the full system. Behavioral design is largely a manual step with little or no automation help available. In this step the control flow, word widths, register allocation, arithmetic operations, and logic operations of the design that represent the functional design are derived and tested. This description can be used in simulation and verification. This description consists of Boolean expressions and timing information. The Boolean expressions are minimized to achieve the smallest logic design which conforms to the functional design. This logic design of the system is simulated and tested to verify its correctness. In some special cases, logic design can be automated using high level synthesis tools. These tools produce a RTL description from a behavioral description of the design. The purpose of circuit design is to develop a circuit representation based on the logic design. The Boolean expressions are converted into a circuit representation by taking into consideration the speed and power requirements of the original design. Circuit Simulation is used to verify the correctness and timing of each component. The circuit design is usually expressed in a detailed circuit diagram. This diagram shows the circuit elements cells, macros, gates, transistors and interconnection between these elements. This representation is also called a netlist. Tools used to manually enter such description are called schematic capture tools. In many cases, a netlist can be created automatically from logic RTL description by using logic synthesis tools. In this step the circuit representation or netlist is converted into a geometric representation. As stated earlier, this geometric representation of a circuit is called a layout. Layout is created by converting each logic component cells, macros, gates, transistors into a geometric representation specific shapes in multiple layers , which perform the intended logic function of the corresponding component. The exact details of the layout also depend on design rules, which are guidelines based on the limitations of the fabrication process and the electrical properties of the fabrication materials. Physical design is a very complex process and therefore it is usually broken down into various sub-steps. Various verification and validation checks are performed on the layout during physical design. In many cases, physical design can be completely or partially automated and layout can be generated directly from netlist by Layout Synthesis tools. Layout synthesis tools, while fast, do have an area and performance penalty, which limit their use to some designs. Manual layout, while slow and manually intensive, does have better area and performance as compared to synthesized layout. However this advantage may dissipate as

larger and larger designs may undermine human capability to comprehend and obtain globally optimized solutions. After layout and verification, the design is ready for fabrication. Since layout data is typically sent to fabrication on a tape, the event of release of data is called Tape Out. Layout data is converted or fractured into photo-lithographic masks, one for each layer. Masks identify spaces on the wafer, where certain materials need to be deposited, diffused or even removed. Silicon crystals are grown and sliced to produce wafers. Extremely small dimensions of VLSI devices require that the wafers be polished to near perfection. The fabrication process consists of several steps involving deposition, and diffusion of various materials on the wafer. During each step one mask is used. Several dozen masks may be used to complete the fabrication process. A large wafer is 20 cm 8 inch in diameter and can be used to produce hundreds of chips, depending of the size of the chip. Before the chip is mass produced, a prototype is made and tested. Industry is rapidly moving towards a 30 cm 12 inch wafer allowing even more chips per wafer leading to lower cost per chip. Packaging, Testing and Debugging: Finally, the wafer is fabricated and diced into individual chips in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly.

DOWNLOAD PDF CUSTOM IC DESIGN FLOW

Chapter 4 : Full custom - Wikipedia

The Single Schematic Flow in Virtuoso System Design Platform simplifies the product usage. It allows you to generate a hierarchical schematic that can be used for simulations as well as for driving the layout of the package.

Comments This is going to be a series of step-by-step explanation of physical design flow for the novice. I am going to list out the stages from Netlist-GDS in this session. Of course some say synthesis should also be part of physical design, but we will skip that for now. Next comes the physical design part of it;making your design into a representation of the actual geometries you will manufacture. You will do a bunch of stuff here, like floorplanning, placement, CTS, routing, timing closure, physical verification, formal verification etc. The major stages are explained below. Netlist In The first stage in physical design flow is reading in the netlist and the constraints to your tool of choice. Let us see what kinds of files we are dealing with here. I have used both Cadence and Synopsys tools extensively, so those are what I will base my examples on. However, every tool uses pretty much the same flow and even the same format files. Gate Level Netlist Once you choose a process and a library, a synthesis tool will translate your RTL into a collection of interconnected logic gates that define the logic. The most common format is verilog. Standard Cell Library In digital design, you have a ready made standard cell library which will be used for synthesis and subsequent layouts. Your netlist will have instantiation of these cells. For digital layout, you need layout and timing abstracts for these cells. Layout Model â€" An abstract model of the standard cell layout is used instead of the complete layout. This will have PINs defined, so as to facilitate automatic routing by the tool as per your netlist. LEF is an ascii file, so go ahead and have a read. Timing Model â€" Tools also need a timing model in the form of a. This liberty format file will have timing numbers for the various arcs in a cell, generally in a look up model. Technology File The rules pertaining to the process you have selected should also be given to the PnR tool. This includes metal widths, spacing, via definitions etc. Timing Constraints SDC files define the timing constraints of your design. You will have the clock definitions, false paths, any input and output delay constraints etc. These inputs once read in, will get you started with your database.

Chapter 5 : ASIC-System on Chip-VLSI Design: Full Custom IC Design

shows the design flow this tutorial will be implementing. In the first three parts of this manual you will design and simulate a CMOS inverter using Custom DesignerSE in conjunction with.

Chapter 6 : PPT - VLSI Design Full-custom IC Design Flow PowerPoint Presentation - ID

3 Introduction IC designers have two options to implement a circuit block: Synthesis / Auto place and route (ASIC) Custom circuit design / Custom Layout (Full Custom).

Chapter 7 : Symica IC Design Flow | SYMICA custom IC design toolkit

Symica IC Design Flow Symica provides a suite of EDA tools for schematic entry, SPICE simulation, mixed-signal simulation, simulation result analyzing and layout drawing. All Symica tools are integrated into and can be run from Symica Design Environment, except SymLayout, which is working as a separate application.

Chapter 8 : ASIC Design Flow - an Overview

Full custom design flow. The design stage is the column on the left, while the column on the right is the tool available at ARL through the Cadence Design Systems, Inc. licensed software.

Chapter 9 : PDKs for Analog/Mixed-Signal/RF Design Houses

The process steps consist of exposing the silicon to UV light under control of a mask, and the washing away the remnants using chemical or mechanical polishing. The layers will typically be N implant / P implant to make transistors, poly silicon t.