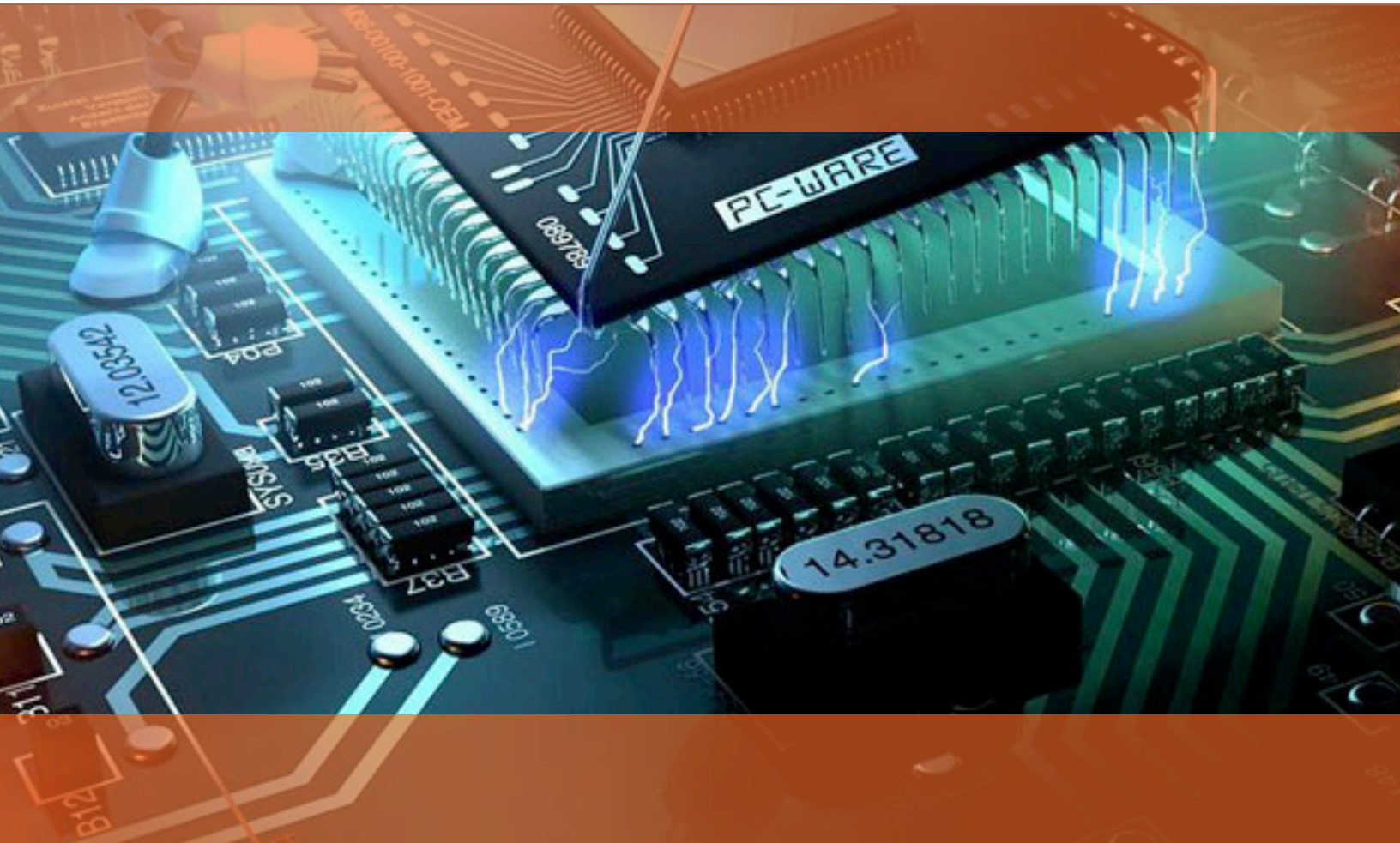
AGNISYS
ACCELERATING DESIGN VERIFICATION

20 Advantages For Automating Your ASIC Or FPGA Transition From Specification To Design



Introduction

The transition from specification to hardware is a challenging and time-consuming task in the design of complex ASIC and FPGA systems. Registers and memory-maps being an essential component of ASIC/FPGA designs, need to be managed and configured diligently, in order to avoid the potential errors, thereby improving the design quality and reducing the development time and cost.

Although the digital design flow is a bit different for ASIC and FPGA, they both include four main steps, namely: specification to RTL description, verification, RTL synthesis and floor planning/place-and-route. The generated code produced from each step of the design, has to be simulated, optimized and verified against the target specification.

A successful FPGA/ASIC design entails an effective communication between all the engineers involved in the design steps including architects, hardware designers, verification engineers, software developers, firmware engineers, etc.

Very often, even after rigorously converting the specifications to a hardware description, the resulted hand-code RTL (used to configure the FPGA or to layout the ASIC) is inevitably error-prone, functionally incorrect and may be non-synthesizable. The issue is exacerbated for more complex designs that require more configuration bits. The high complexity level makes the task of error detection and debug quite frustrating and time consuming. The most common errors incurred during the transition from specification to hardware include register address overlaps, non-traceable register names, inconsistent data, illegal access, etc.

The flow of ASIC/FPGA designs starts with a database of specifications that are typically given to engineers in a textual format to be converted to a hardware description. The specification document is often a lengthy document containing unstructured representation of memory map information and register structure, using standard text processors (such as Microsoft Word). Note that standard text processors are particularly appealing to system architects in view of their ease of use and the fact that no special software training is needed to create and maintain the specifications.

However, the unstructured specification database does not provide a reliable way of extracting the required data from the document. Furthermore, it is not easy to maintain the consistency among the documents that are derived from such a database. The unstructured textual specification database is, therefore, a potential source of design and verification errors.

There are also more structured database formats such as IP-XACT and SystemRDL, which facilitate the generation of the required design views and documentation.

1-855-VERIFYY

Transition from Specification to Design

For a given set of design specification, a standard method for implementation of addressable registers is non-automated register development, in which the engineer manually creates the register model. Manual configuration of registers requires an extensive effort and is inevitably error-prone.

A more time-efficient alternative to deal with this problem is to use “in-house” tools in a semi-automatic fashion. More precisely, the engineer receives the structured database or describes the registers as a structured database (e.g., extracting an XML file from an excel file), and then uses a script (e.g., Python, Perl, etc.) to generate the register models and other derived documents (such as VHDL packages, C headers, HTML documentation, etc.). An example of an open source Python-based HDL register model generator is called HDLRegs. The file generator receives the register specification in a text-file formatted according to the JavaScript Object Notation (JSON) and uses a Python script to generate the output files. The in-house tool option requires a tedious job to develop and maintain the tool (script). More importantly, this approach is limited to intra-organizational use. That is; the tool is developed for a specific design and cannot be used (or has to be modified substantially) if the specifications are changed for future designs or new functionality is added to the same design during the development process.

Both the non-automated and semi-automated methods mentioned above fail to address the critical issue, namely, managing the synchronization and consistency between all the views (description, design, documentation, verification and software). The significance of synchronization is further highlighted because the register information is needed at all levels of design, implementation and verification. More specifically, although it is a design team that implements the registers, the generated file has to be accessible to the verification team, the software team, firmware team and finally the customers (both internal or external).

The non-automated implementation of registers (manually performed by the design or verification engineer), as well as being extremely time consuming

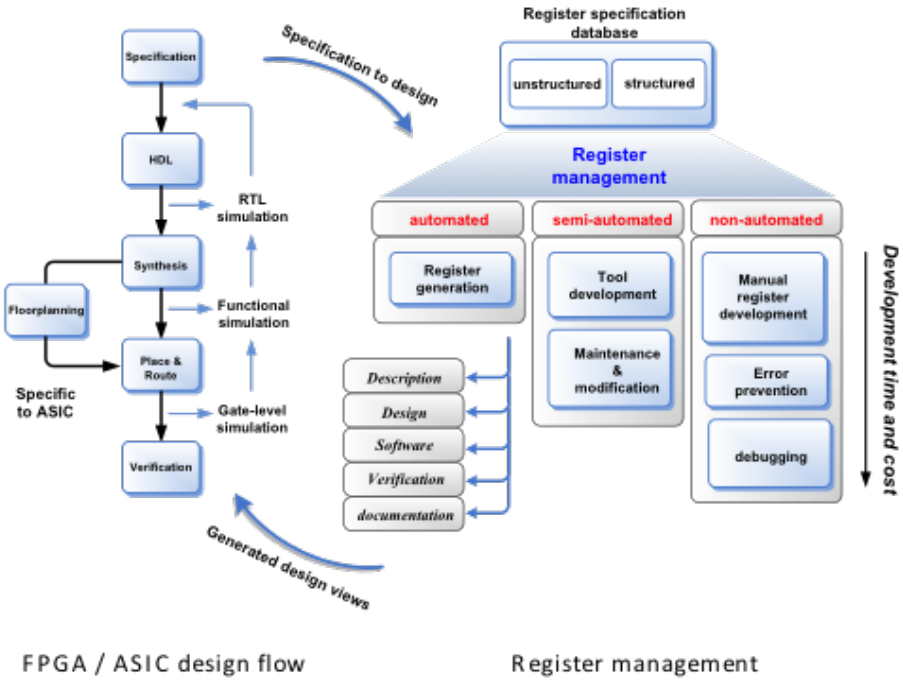


Figure 1. FPGA / ASIC design flow and register management scenarios

and error prone, does not provide an efficient means of communication with other members of the design group (for simulation and verification purposes). Note that the manually generated file is normally not accompanied with other useful derived documents and design views to be used by other members of the team. The documents and design specification can change at any stage of the design. Failing properly to reflect and distribute these changes to other documents and views can lead to serious consistency problems, which prolong the development time and raise the cost.

The semi-automatic (hand scripting) method significantly saves the development time, particularly when there is a multitude of registers and associated bits to be dealt with. However, aside from the hassle of writing and maintaining the script, the developed tool does not naturally lend itself to real-time error checks (such as conflicts, overlaps and missing values). Moreover, such tools are typically designed to be compatible with a specific input format and produce a particular output (design view or document) unless a huge amount of design work is invested in the script.

Automatic Register Management Method

An efficient solution to accelerate the development time while eliminating the potential errors, preventing the miscommunications between the members of the design team and maintaining the consistency among the different design views and documents, is to manage the register model generation, and memory mapped addresses in an automated fashion. The automatic register model generators are offered as commercial tools. These tools support multiple standard formats for the specification database. These standard formats include IP-XACTTM, Verilog, VHDL, Excel, Microsoft WordTM, OpenOfficeTM, Frame MakerTM, etc. The commercial tools may or may not support customer-specific input formats.

From the imported input file, the commercial tools then generate different design views and documents for design, verification, software development and documentation. The possible output formats include, but are not limited to Verilog, VHDL, SystemVerilog for design, OVM and UVM for verification, C++ header for software development and HTML, PDF for documentation. Also, these tools typically allow creation of user defined outputs using scripting languages such as Python, Perl, Tcl, etc.

These register automation tools are offered as standalone applications, within the platform or plug-in for editors. They provide a correct-by-construction GUI for creation and management of control registers. More precisely, they perform real-time error checking, and produce the desired outputs such that any modification that is made on the register specifications is automatically reflected into each output.

There are a few commercial products available for automatic creation and management of register specification and memory maps. Examples of these commercial tools include:

- **Bitwise SOCRATES™ by Duolog Technologies.** Bitwise is an application within the SOCRATES platform, which provides a single specification source (IP-XACT, Excel, Verilog / VHDL or other user-defined formats) and automatically creates the design views for design, verification, software and integration teams. It also

allows the creation of customer-specific outputs by using standard scripting languages. ARM Holdings PLC recently acquired Duolog Technologies.

- **Blueprint™ by Denali Software.** Blueprint is a compiler that imports the textual specification source (SystemRDL) and provides an interface from specification to output generators (for design, verification, documentation, software development, post-silicon debug and SystemC™). SystemRDL is viewed by many as an out of date specification because it does not contain enough information to support modern designs. Denali Software was acquired by Cadence Design Systems, Inc., who has discontinued support for this product

- **SpectaReg by PDTi™.** SpectaReg is a register map automation tool, which supports IP-XACT XML format as the single-source and auto-generates the desired views and documentation such as VHDL, Verilog, SystemVerilog, C/C++, DITA, HTML, and RALF.

- **SpiritEd:** SpiritEd is a FrameMaker™ based environment providing a graphical user interface to realize an automated specification flow for register description. The register data is captured in an XML database using IP-XACT. An interface is provided for SpiritEd, which enables users to write generation engines (scripts) for specific outputs. SpiritEd facilitates an automatic synchronization between the register database and the derived documents.

- **US 8316330 B2:** The editor described in US 8316330 B2 uses a word processor (IP-XACT®, CSV, EXCEL®, etc., input to an XML processor and coupled to the editor) for register data entry and dynamically detects and modifies the errors during the data entry process. Data consistency is checked during data entry and feedbacks are created for the users in a visual way. This tool is capable of generating general-purpose (output of the editor coupled via XSL transformer to VHDL, Verilog, C/C++, IP_XACT®, OVM, VVM, etc.) or user-defined documents. Agnisys, Inc. owns this patent.

- **IDesignSpec™** by AGNISYS: IDesignSpec is an advance register generator that is added as a plugin to conventional text editors (Microsoft WordTM, Excel, OpenOffice® and FrameMaker). Various outputs such as UVM, OVM, RALF, SystemRDL, IP-XACT, etc., are generated from the input source. IDesignSpec also supports user defined outputs produced using Tcl or XSLT scripts.

Compared to manual and semi-automatic register generation methods, the automatic approach brings crucial advantages to the design team by reducing the development time and cost. Incorporation of an automatic scheme becomes increasingly relevant because the ASIC/FPGA designs are continually growing in terms of size and functionality and, therefore, need more configuration bits. The automatic register management approach makes the logic designer more productive. It also improves the quality of the design as it allows more time to be spent on the verification rather than the detailed implementation. Moreover, such automated designs are more readable that simplifies their use by other members of the design team.

The 20 Advantages of Automating Your ASIC or FPGA Transition from Specification to Design

Reducing the development time and cost through automation of the laborious tasks:

1. Straightforward, intuitive and easy to use
2. Graphical user interface
3. Customizable input/output formats
4. Scalable with the design size
5. Non-design-specific
6. Low maintenance effort
7. Updatable

Reducing the risk of human errors:

8. Automatic error detection
9. Automatic error modification
10. Automatic verification

Providing synchronization and consistency between engineering teams:

11. Automatic creation of various derived documents
12. Automatic distribution of modifications to all of the outputs
13. Interoperability with other EDA tools
14. Applicable at different stages of the design
15. Appropriate for both inter-organizational and intra-organizational uses

Improving the quality of the design:

16. Enhanced engineer productivity
17. More efficient teamwork
18. More accurate verification
19. High quality documentation
20. Adjustable customer-specific requirements

AGNISYS
ACCELERATING DESIGN VERIFICATION

1-855-VERIFYY

IDesignSpec The Preferred Option

As mentioned above, intuitiveness and ease of use is a highly desired attribute of automated register management tools. IDesignSpecTM is an engineering tool, which provides a high level of automation in the creation of register map specification and at the same time, maintains simplicity of use. IDesignSpecTM uses a single source for the specification document. The tool is added as a plugin to standard text processors such as Microsoft WordTM, and Excel®, therefore, enabling the generation of register specification without needing to learn a new software or language. The added plugin performs error checking in the document editor based on design rules and generate error flags if the rules are violated. Once the errors are corrected, the tool creates usable codes and views from the specification files. This tool provides individual access to each slice of the register for the purpose of description and verification. IDesignSpec minimizes the level of human intervention and, therefore, reduces the risk of accidental errors in generating the output codes and, as a result, improves the quality of the design and reduces the development time and cost.

Input formats supported by IDesignSpec include:

- MS Word,
- MS Excel,
- IP-XACT,
- SystemRDL,
- generic XML,
- CSV and plain text (require a simple transformation)

Output formats supported by IDesignSpec include:

- Synthesizable RTL: Verilog, VHDL, SystemVerilog.
- Verification: UVM, OVM, VMM (RALF), eRM (vr_ad).
- Firmware: C header files, C++ Classes.
- Industry Standards: IP-XACT 1.5(1685-2009), IP-XACT 1.4, SystemRDL.
- Documentation: HTML, PDF, XML, SVG.
- User-specific outputs (using Tcl, XML, Perl scripts)

Implement an automated register management tool and gain the 20 advantages of automating your ASIC or FPGA transition from specification to design. Reach out to Agnisys to learn how.

AGNISYS ACCELERATING DESIGN VERIFICATION