

QuteIP: An IP Qualification Framework for System on Chip

Hsing-Chih Hung, Ting-Hao Lin, and Chung-Yang (Ric) Huang
Graduate Institute of Electronics Engineering,
National Taiwan University

ABSTRACT

The robustness and reusability of Intelligent Properties (IPs) are the keys to the success of the modern System on Chip (SoC) designs. Therefore, it is very important to implement a rigid IP qualification (IPQ) framework to ensure the quality of IPs in the SoC design flow. In this paper, we carefully examine the advancements and limitations of previous IPQ tools and propose several novel concepts and methodologies to construct a new IPQ framework. We make the framework easily configurable so that it can be applied to different IP vendors with different IP users' needs. We design the user interface from the practicality point of view so that it is simple, yet complete enough to support various usages in the IP qualification process. Most important of all, our implementation does not include any proprietary information so that the readers should be able to reconstruct a similar IPQ framework on their own. We have deployed our framework to several IP design teams and their feedbacks showed that our IPQ framework can help them reduce the time in design and greatly improve the quality of their IPs.

1. Introduction

With the increasing complexity in modern System on Chip (SoC) designs, Intelligent Property (IP)-based design flow is becoming an inevitable trend. Consequently, the success of a SoC design does not depend on how efficiently a group of engineers can build the chip from scratch, but relies on how smoothly they can leverage the individual IPs from IP vendors. Therefore, it is critically important for the IP vendors to provide robust and ready-to-use IPs, and guarantee a short turn-around time when problems arise.

IP Qualification (IPQ) process, as its name suggests, is a rigid process by the IP vendors to qualify the IPs and thus fulfill the SoC design requirements. Its goal is to make sure that all aspects of the design constraints are met and the IP delivery documents are well prepared. In order to have a quantitative measurement, IP vendors usually use a "check list" of IPQ rules to evaluate the completeness of the qualification process. However, different vendors usually have different sets of rules and tool flows, which make it difficult for the SoC design house to integrate and configure IPs from varied IP vendors. Therefore, it is necessary to define a standard for the IP qualification and integration.

The Virtual Socket Interface Alliance (VSIA) is an open organization developing SoC, IP and reuse standards to enhance the productivity of SoC design [1]. It has created and adopted (e.g. [2][3]) more than 20 documents, specs and standards. The most widely adopted one, the QIP Metric, offers an executable, self-scoring spreadsheet that can help IP vendors to qualify their IPs with a wide range of IPQ rules. On the other hand, the SPIRIT Consortium,

feeling that "...*(VSIA) has not managed to focus, however, on bringing things to market*" [4], announced the IP-XACT specifications to provide a standard to describe behavior of IPs and also the design environment. Its IP meta-data description is an XML schema for a common and language-neutral way of IP description, and the tool integration API provides a standard to exchange design architectural data.

While the organizations like VSIA and SPIRIT offer a comprehensive set of documents in standardizing the IP qualification process, there are still great efforts remained for the EDA companies and/or IP vendors to construct an IPQ framework for the actual IP evaluation and integration. The IPQ project in [5] developed EDA methods for quality assurance in IP design and integration. In [6], the authors implemented an automatic qualification framework with several qualification methods, like completeness check, coding guidelines and design rule checks, verification and synthesis qualification, etc. The work in [7] also developed a methodology for automated and design flow-oriented IP quality checks and the authors in [8] proposed an automatic qualification flow for technology independent soft IP.

The above researches have been implemented in several IP design houses and tested with IP users, and have achieved great improvement in the IP qualification process. However, there is still room for improvements. First, the automated IP rule checking capability, although very helpful in reducing the validation effort, may be vulnerable to the design flow and tools updates. Secondly, it takes great manual efforts to build up the IPQ process even with the highly-automated IPQ tools. When we move from one design project to another, there are usually high similarities in the design flow and IPQ process. This leaves room for us to improve the reusability of the IP qualification flow. Lastly, different IP vendors have different flows and IPQ processes. Even with the standard IP rules, the IPQ tools should make it easy to customize the design flows and IPQ rules.

In this paper, we propose several concepts and methodologies in constructing an IP qualification framework to answer the above improvement possibilities. Highlights of our IPQ framework include: (1) Semi-automated rule checking by using "plug-ins", (2) IPQ project managements with re-loadable project file and template, (3) Complete and yet easily extensible IPQ rule compilation, (4) On-the-fly monitoring of the IPQ process, and (5) Customizable IP design flow. The details of our framework will be described in the next Section.

2. QuteIP Overview and Implementation

The qualification flow of our IPQ framework, QuteIP is as shown in Figure 1. First, the IP vendor and the IP user should have an agreement on the IPQ rules (step ①). These rules can be adopted from the predefined standard (e.g. VSIA's QIP metric) and then customized based on the

actual needs. The design manager then works with the IP designers on setting up the design flow, including the EDA tool environments, the plug-ins for the automatic rule check, and the detailed requirements on the IPQ rules, etc (②). Then during the IP design process, the IP designers can use our framework to follow the IPQ guidance, put comments on the individual rules, and monitor the progress constantly (③). After the IP qualification achieves a satisfactory level, our framework can help generate an IPQ report, package the IP, and then connect to the IP sales channel (e.g. IP mall) (④-⑥). The IPQ process can also be saved as a project template so that IP designers can easily reuse it in the future (⑦).

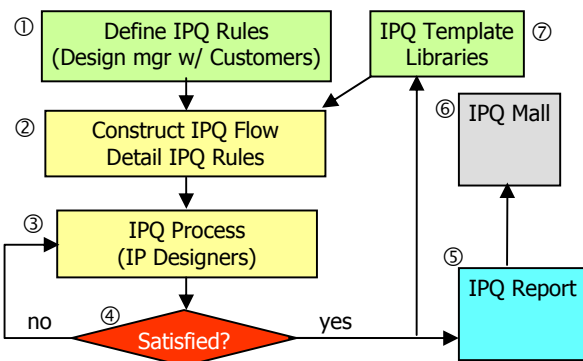


Figure 1. QuteIP IPQ flow

2.1 Creation of IP Qualification Rules

As we mentioned above, most of the IP qualification tools provide a “check list” of IPQ rules to evaluate the completeness of the qualification process. In general, the design of the IPQ rule checker needs to meet two sides of requirements — (1) It needs to cover a wide range of common rules so that the IP users can compare the quality of IPs acquired from different vendors, and (2) It should be easily extensible for customized rules in order to accommodate different types of designs and flows.

Therefore, as in [7], we divide the IPQ rules into two categories — “predefined” and “user-defined”. The predefined rules are the ones that are loaded in the beginning and cannot be added, removed or modified during the execution. They are represented in an XML file and classified as fixed groups of rules.

Fig. 2 shows the graphical user interface (GUI) for the predefined rules window. The illustrated rules are adopted from the VSIA Soft IP development metric 2.0.3 release. It defines the IPQ rules in two groups: “IP Ease of Reuse” and “Design and Verification Quality”. The column “Rank” describes the severity levels of the rules. The characters ‘I’, ‘R’, ‘G’, ‘O’, and ‘M’ stand for “Imperative”, “Rule”, “Guideline”, “Optional”, and “Mitigable”. They correspond to different scores in the rule grading system and will affect the IPQ progress report during the IPQ rule checking process. We also support the predefined IPQ rule sets from ITRI [9].

The user-defined rules, on the other hand, can be modified during the IP qualification process. We provide a

friendly GUI so that the users can freely add and remove rule groups and edit the rules accordingly. However, we usually recommend the design manager to set up the user-defined rules in the beginning with the designers based on the design spec. Therefore, in the later IP implementation phase, the designers can follow the rules to make sure the design meets the spec.

We also recommend grouping the user-defined rules based on the design flow and/or EDA tools. This can act as an alternative way to document the IP design flow. When these groups of rules are saved as an IPQ project template in an XML file (described later), the IP designers can reuse it for the future designs and easily follow the previous design setup and environment.

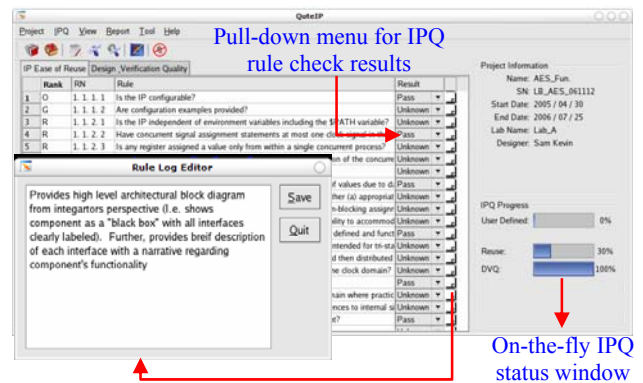


Figure 2. IPQ Rule Checker (Main) Window

2.2 IPQ Rule Checking

The IPQ rule checking determines the completeness of the IPQ process. We propose three novel ideas in making the IPQ rule checking more flexible and easy to use: (1) semi-auto rule checking using “plug-ins”, (2) comment editor for the individual rules, and (3) on-the-fly status checking and report.

Ideally, it is plausible if we can parse the output log files of the EDA tools and then automatically retrieve the necessary information to determine the satisfaction of the IPQ rules (e.g. [5]-[8]). However, many of the IPQ rules cannot be determined this way. Moreover, the automatic rule checking mechanism is also very vulnerable to the updates of the EDA tool flow and versions.

Therefore, we propose a *semi-auto* rule checking mechanism using “plug-ins”. The “plug-ins” are external programs in any executable format, as long as they can evoke a call-back function of our tool to set the status of the individual rules. The users can associate these plug-ins to the rules for the automatic checks. They can apply these plug-ins whenever they feel ready to update the status of the rules. For the rest of the rules that can not be automated, we provide a pull down menu for the users to manually determine the status of the rules (Fig. 2).

Another nice innovation of our IPQ rule checker is the “comment editor” for the IPQ rules (Fig. 2). It is worthwhile to note that the passes/fails of the rules can only provide a “quantitative” measurement of the IP robustness. They say nothing about the detailed result of each rule. Therefore, the

users can utilize the “comment editor” to record the information and provide insights for the later design review.

We also support an on-the-fly monitor for the IPQ checking progress. This idea, similar to the scoring spreadsheet in the VSIA metrics, can not only serve as an instant feedback to both the design managers and engineers on the design schedule, but also facilitate the communication between IP vendors and users about the completeness of the IPQ process.

2.3 IPQ Project Management

An IP design in our qualification framework is called an “IPQ project”. For each IPQ project, we create a project file in XML format to record the necessary information of the IP qualification process. The project file can then be saved, reloaded, and restored at any point of time. We use it to manage the IPQ project as follows:

Project information: As shown in Fig. 2, users can specify the project name, series number, project execution period, affiliation (lab/department), and designers’ names in the IPQ main window.

Design environment: In the IP design process, designers usually need to apply various EDA tools in different stages, with different scripts, files and directory structures. Therefore, how to create and maintain the IP design environment is indeed a tedious and time-consuming job. On the other hand, it requires a complicated user interface from the tool to record the setups of the design environment. Therefore, it may not at all ease the task.

We take a compromise approach by associating executable scripts with EDA tools as shown in Fig. 3. The detailed settings are hid in the scripts and should be edited by the designers off-line. We provide a table to clearly list the scripts with descriptions so that the users can easily apply them in the IPQ process.

IPQ rule checking information: We also record the IPQ rules and their corresponding checking results with comments in the XML file.

IPQ project template: In addition to the IPQ project file, users can also save the project as a “template”. All the user-defined rules, plug-ins, and design environment information will be kept in the template file. On the contrary, the rule checking results and project information will be reset. Therefore, when another new design is created, the users can open a new IPQ project with the template file and thus save a lot of efforts in re-constructing the IPQ environment.

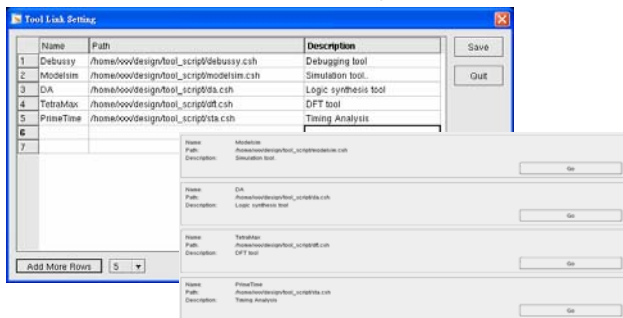


Figure 3. Setting up EDA tool links

2.4 IPQ Report, Packaging, and Delivery

The report of the IP qualification can be displayed on the fly, or be outputted as a HTML file for viewing with an external web browser.

We also implement a file browser so that the users can easily select the files she/he wants to include in the final IP package. Please note that we recommend selecting the files incrementally during the IPQ process, instead of picking the files all at once after the IP is completed. This can help prevent the erroneous selections when the number of files we need to deal with is huge.

What we have not done well with respect to the IP delivery are (1) IP encryption, and (2) more sophisticated links with IP mail. For (1), we will refer to the IP encryption standard in VSIA [3]. For (2), we will work with the local IP vendors to define the protocols for the IP upload and download.

3. Case Study and Experimental Results

In order to testify the usefulness of our IPQ framework, we deployed our framework to several local design teams in order to acquire feedbacks. The designs they are working on include multimedia (H.264, mpeg 4, GPU, etc) and communication IPs (LDPC, AES, etc).

3.1 Original IPQ Process

In comparison, we first describe their original IP qualification process without using our framework.

IP Interface: Most of their designs are based on well-established standards. However, the so-called well-defined specifications may still have different interpretations between the design teams (IP vendors) and their customers (IP users). When it happens, usually the design teams are reluctant to make the change for the specific customers.

Design flow: The design teams have their own design flow, and very often it is different from the IP users’. However, they do not discuss in advance the difference and possible issues with the customers. If any problem arises, primarily it will be the IP users’ job to fix it.

Functional verification: They perform linting, simulation, equivalence checking, and FPGA prototyping for the functional verification. The verification reports are scattered in different IP deliverable documents. They do not have a unified document to summarize the verification efforts and results.

IP packaging: Surprisingly, this is one of the most time-consuming steps in their IP design process. After the IP is ready to release, they manually collect all the necessary documents along with the design itself. Since they do not have a mechanism to force the preparation of the IP packaging in advance, they need to spend excessive amount of time at the end to figure out the list of relevant files, and recall the details for the IP deliverable documents.

3.2 IPQ with Our QuteIP Framework

We acquire their feedbacks on using our IP qualification framework. Although it is difficult to have a quantitative

conclusion about their experiences, it can be clearly shown that they can both reduce the time in designing and improving the quality of their IPs. We will show some of the IPQ results at the end.

IPQ rule creation: In the beginning, they find this a somewhat painful process — it is not practical for them to just import an open IPQ metric (e.g. VSIA QIP). They need to go through the document one by one rule to make sure it fits their needs. In addition, they supplement the predefined rules with the original IP design requirements, many of which are too specific to be covered in the common IPQ metric. They also create several user-defined rules based on the IP specification, mainly to capture the design constraints. Although this IP rule creation process takes them a great amount of time, they find it very rewarding because it forces them to talk to IP users and triggers many internal discussions even before the design starts.

IPQ project template: They use the template not only for creating a new IPQ project, but also for enriching the IPQ rules. For example, they did not have a well defined document for the requirements on the Design for Test (DfT) process. They found that the other team had begun to use our IPQ framework and had created a nice one. Therefore, they just asked for the template file from the other team, copied the DfT rules in the XML file, and appended it to their user-defined rules.

IPQ rule checking: Since they are the early adopters of our IPQ framework, they have very few plug-ins for the automatic rule checkers. Therefore, they need to manually check most of the rules on their own. However, they do not think this an overhead because they can tell the rule status very promptly. What is more interesting is that they even think the manual checking is more reliable than the automatic one because the automated approach is more likely to result in potential false positive and/or negative.

IP packaging: They also find that our proposed IP packing mechanism very useful. Whenever they finish a major step in the design process, they will arrange the files and scripts and select them in the IP packaging file browser window. This makes the IP packaging a much easier task.

Verification qualify: They have now a much better confidence about their designs. After going through a robust IP qualification process, they have fixed many bugs and ended up with a clear document about the IP verification results. This is a win-win situation for both the IP provider and consumer.

3.3 Experimental Results on IPQ Checks

We acquired two IP designs “DWT (Discrete Wavelet Transform)” and “High-Speed SRAM Controller” to walk through the IPQ rule checker with the list of rules provided by [9]. The results are as shown in Table 1:

The “RTL”, “Syn”, “DfT”, “PKG”, and “User” stand for the groups of rules for RTL design, logic synthesis, Design for Testability, packaging, and user-defined, respectively, while “U/P/W/NP” mean the “Unknown/Passed/Warning/Not passed” rule checking statuses. By checking the source codes, log files, and documents from the design teams, we,

as design verification engineers, can validate these IPQ rules within 3 and 2.5 hours, respectively.

Table 1: IPQ Rule Check Results

	DWT (3 Hrs)				SRAM (2.5 Hrs)			
	U	P	W	NP	U	P	W	NP
RTL	2	43	9	41	0	79	1	15
Syn	0	20	2	1	0	23	0	0
DfT	0	10	0	1	0	11	0	0
PKG	0	6	5	5	0	15	2	3
User	0	5	0	1	0	4	0	1
Total	2	84	16	49	0	132	3	19

The results show that the DWT design has more “not-passed” IPQ rules. Many of them are due to the RTL coding style checks. For example, it fails on the following rules: “Consistent clock/reset name”, “No file header”, “All timing exception must be identified”, and “Clock strategy must be documented”. It also has problems on rules like “(Syn) No Specify code fragment for combinational logic completely”, “(PKG) Document for verification related information”, and “(User) Power consumption should be less than 5mw (5.09 mW)”, etc. We have feedbacked these failures to the design team to help improve their IP quality.

4. CONCLUSION AND FUTURE WORK

In this paper, we propose several new ideas in constructing a robust IP qualification framework. We believe that our proposed methodologies are simple enough so that the readers should be able to reconstruct a similar IP qualification framework on their own. We will continue to deploy our framework to more design teams in order to further improve the proposed IP qualification flow. At the same time, we will also work with the IP vendors to improve our features in the IP protection and web-based IP delivery.

5. REFERENCES

- [1] <http://www.vsi.org>, VSIA homepage.
- [2] <http://www.OpenMORE.com>, OpenMORE Homepage.
- [3] http://vsi.org/news/synplicity_joins_vsi_alliance.htm, “Synplicity Donation to VSIA Drives IP Encryption Standardization.”
- [4] http://www10.edacafe.com/nbc/articles/view_weekly.php?articleid=209263&page_no=6, quoted interview, Ralph von Vignau, SPIRIT consortium chairman.
- [5] H.-J. Brand, S. Rülke, M. Radetzki, “IPQ: IP Qualification for Efficient System Design”, Proc. International Symposium on Quality Electronic Design (ISQED), 2004, pp 439 – 444.
- [6] A. Vörg, W. Rosenstiel, and M. Radetzki, “Measurement of IP Qualification Costs and Benefits”, Proc. DATE, 2004, pp.996- 1001.
- [7] R. Seepold, N. M. Madrid, A. Vörg, W. Rosenstiel, M. Radetzki, P. Neumann and J. Haase, “A Qualification Platform for Design Reuse”, Proc. ISQED, 2002.
- [8] M. Radetzki, P. Neumann, “Automated Qualification Flow for Soft IP”, Proc. MEDEA+Conference on Application-Oriented SoC Design, 2001.
- [9] <http://www.itri.org.tw/index.jsp>, ITRI IPQ rules.