Reusing VC Blocks

Reuse of Virtual Components (VC), also known as hardware or silicon Intellectual Property (IP), has become a crucial strategy for design teams. Designers now face design cycle times as short as 3 months for hot items like cell phones, internet routers, DVD players, digital cameras, etc. Short cycle times give the manufacturer a potential leg up in the race for market share, crucial for growth and survival in today's global markets.

Designers must also contend with the fact that IC complexity is growing at a rate of 58% per year, but design productivity is increasing at only 21% per year. Creating VC specifically formatted for the reuse needs of the integrator helps bridge this gap. Reusable VC requires a well-documented, parameterized general solution at a high level of abstraction. This block can then be easily adapted to meet various functional and process technology requirements in subsequent applications. Reuse provides both VC creators and system-on-a-chip (SoC) developers a method for sharing best practices and best design blocks. According to a recent industry study by Collett International, current non-optimized VC reuse requires 41% of the original design effort to implement. However, implementation under optimized conditions reduces the integration effort to 7% of the original design effort.

1. Platform Strategy for Faster Design Cycles

Today's designers simply don't have the resources available to develop new autonomous product designs from specification to tape-out in a three month long design cycle. The solution is to design a first-in-a-series platform for a new generation product offering. By establishing a design platform for each product family, the design team can leverage the platform to create future derivative versions of the product. With a platform in place, designers can reuse as much as 90% of the original embedded blocks as is. The design team can incorporate new features and advanced functionality using as little as 10% of the blocks. In cell phones, several manufacturers have added new phone platforms. Subsequent releases on the same platform have included new features such as extended battery life and web communications.

2. Selecting a Reuse Standard

The Semiconductor Reuse Standards (SRS) were developed within our company to cooperate with industry reuse initiatives such as the Virtual Socket Interface Alliance (VSIA). The VSIA specifications are available at **http://www.vsia.com**.
The Motorola Semiconductor Reuse Standards (SRS) are available at **http://www.mot-sps.com/srs**. The reuse standard provides the playing field, the rules, and the interface methodology between stages of the VC design flow. The five categories in the SRS range from system level aspects, through architectural, implementation, and verification issues to general VC related aspects such as documentation and the deliverables list. The architectural category, for instance, addresses on-chip and off-chip interfaces. The implementation category focuses on VC creation and HDL coding

standards, physical representation, cell libraries, and analog. Functional verification and testability standards are included in the verification category.

3. Socketizing for Reuse

SoC designs can incorporate modular blocks that are “plug-and-play” (hard VC), fully synthesizable (soft VC), or a hybrid of both approaches. The VC creator’s objective is to create blocks “pre-designed” for reuse, meaning they comply with the relevant reuse standard. System designers can then successfully integrate these blocks into their design. To properly integrate the VC, the system designer will require a set of “views.” Only the VC creator has sufficient knowledge of the block to create these views, eliminate false paths during timing characterization, and determine the correctness of the results.

4. Offering and Obtaining Reusable VC

The VC Repository provides an infrastructure where creators can offer and system designers can acquire VC for reuse. The repository system allows participants to share information in the form of Soft/Firm/Hard VC of many different types. The repository provides global access for a multi-site, global development community of system integrators. It allows these integrators to quickly and easily search, select, and integrate available blocks into their designs. The repository infrastructure is, in effect, an elegant, web-enabled database that allows designers to search for, screen, select, and download reusable VC blocks.

5. What Type of VC Block

Three crucial characteristics must be applied to any potential reuse need and guide the system designer to a sound choice; Reusability, Retargetability, and Reconfigurability. A reusable VC block complies with the relevant reuse standard’s requirements for delivered views, the associated data formats, and view specific standards, such as coding standards. A block is retargetable if it is available in a high level representation, such as synthesizable HDL, which allows the designer to easily migrate the VC to new technologies or processes. Configurability means that the VC block is parameterized to allow for adjustments by the user to meet specific functional or performance objectives. Examples of parameters include bus width, memory size, and enabling or disabling functional blocks.

6. The “Plug-and-Play” Choice

Reusable hard VC is optimized for power, size, or performance, and mapped to a specific technology. Examples include netlists fully placed, routed, and optimized for a specific technology library, a custom physical layout, or a combination of the two. Hard VC is process/vendor specific and generally expressed in GDSII format. It has the advantage of being more predictable, but consequently less flexible and portable due to process dependencies. Hard VC requires, at a minimum, a high-level behavior model, a test list, and full physical and timing models along with the GDSII data. Hard VC is the easiest to integrate into a design since it has already been modeled accordingly, delivers confirmed and reliable performance, is proven in silicon, and offers the design team established quality.

7. Or Choose to Synthesize

Reusable soft VC is delivered as synthesizable HDL code. The advantage for the system developer is the flexibility of the source code, because it can be retargeted to multiple manufacturing processes. The disadvantage is the difficulty in performance prediction (e.g., timing, area, and power). The designer selects a soft VC block to meet performance goals in the context of the specific design project immediately at hand. Because soft VC comes in the form of HDL code, there are few deliverables and the design team can configure and develop the VC into their application. Choosing soft VC gains design flexibility but gives up both time savings and ease of integration compared to hard VC.

8. Certifying an IP Block

Certification determines whether an IP block is reusable in accordance with the relevant reuse standard. It also provides metrics that indicate the quality of the block for reuse. Most of the certification is done before the IP enters the Repository. Additional certification may be done while the IP is in the repository. The certification team uses a View Generation System to test the block for consistency between views and also tests for compatibility with tools, libraries, and hardware platforms.

9. Grading & Feedback

The certification team also prepares a prediction of the quality and reliability of the IP block (deliverables, usage, and defect history), also called “grading.” In the SRS standards, the rating is Bronze, Silver, or Gold. The repository infrastructure also contains a feedback section that allows system developers to provide feedback, bug reports, fixes, and notes about any VC block in the repository. This feedback becomes part of that block’s database entry along with notes of bug fixes and improvements by the creator(s) of the block.

10. Downloading a Block

Meta data about each available VC block is maintained and stored in an Oracle relational database. The database operates on a client-server architecture. Using appropriate search criteria in a standard web browser, the consumer searches the meta data in the database. Using its powerful web based search capabilities, the relational database locates, selects, and lists the relevant available blocks that meet the user’s search criteria. The user can combine different search methods to locate the desired design block. The user can search by keyword and or conduct a category (taxonomy) based search. The user can also implement filtering capabilities based on pre-selected meta data to locate the requested design block. Criteria could include a high level functional description or the maximum speed of a requested block. In response to the search request, the browser queries the meta data database server and dynamically displays the search result on the user's web browser. Once the system integrator has selected a block, the integrator simply downloads via the web browser.

* * * END * * *

(Word Count: 1392)

Alternate #10

10. Optimizing VC Integration

For a VC block to be reusable, there are four process steps, three of which are performed by the creator of the block, socketization, certification, and depositing in the repository. Automated tools provide the creator the means to quickly perform socketization and operate the repository infrastructure. Some tools partially automate the certification and integration steps. Tool manufacturers are working dilegently to additional automated steps in these processes. Integration steps can be performed either partially by the creator, or wholly or in part by the system integrator. In In the ideal case, all VC blocks are being developed in a synthesizable soft VC form for easy migration to new technologies. On demand of the SoC designer, the VC author hardens and qualifies the block into a hard VC form. The author then ships the hardened VC block to the SoC designer to meet the required cycle time reduction.