Rapid IP Deployment Speeds Time to Market

Integrating MIPS® synthesizable 32- and 64-bit
CPU cores into SOC design flows

Christian Berg
MIPS Technologies, Inc.
January 2002

Increasingly shorter time-to-market requirements demand the integration of third-party IP (intellectual property) into complex SOCs. How rapidly and effectively this integration is achieved is crucial to the success of a project. This article provides a detailed overview of MIPS Technologies' deliverables and the necessary tools to support rapid IP deployment and help ensure fast time to market.

Introduction

Today's chip design process can be characterized in three basic steps (see Figure 1). The first step is the entry of the RTL code describing the functionality of the SOC and its verification. Second is the synthesis step, which maps the RTL code into a library-dependant gate-level representation. The third step is the physical layout of the design. Any provider of synthesizable IP blocks should fully support these steps with deliverables that guarantee uncomplicated and correct integration into the customer's design.

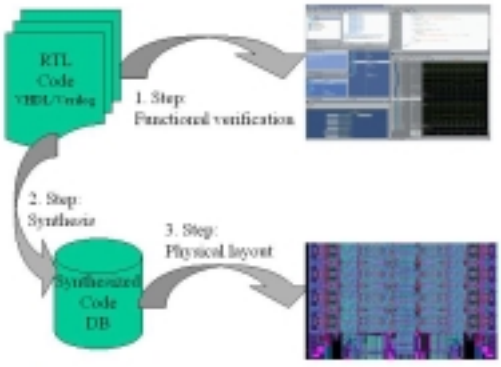

Figure 1: Simple design flow.

Getting Started

Delivery of the core is usually the first point of contact for the customer, and MIPS Technologies makes it easy and comprehensive. Cores are packed into an encrypted tar

archive available for download. For additional security, the download is protected via the HTTPS protocol using a 1024-bit RSA algorithm. Login credentials are submitted using express mail with a return receipt or via direct fax.

The tar package is unpacked using the Unix tools GNU tar and crypt. This yields an installation of the core package with a directory structure as shown in Figure 2. This structure is common to all MIPS® deliverables. A special configuration file is used to adapt the core environment to the customer's specific conditions.

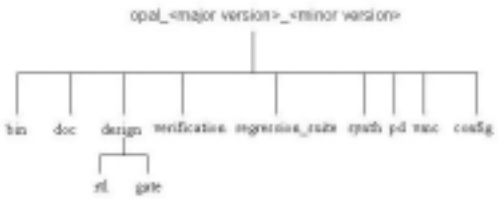

Figure 2: Sample directory structure of the 64-bit MIPS64™ 5Kc™ core.

The directories are as follows:

- The 'bin' directory contains all the scripts and executables needed to work with the core.
- Documentation in the form of manuals and guidelines can be found in the 'doc' directory.
- The actual code of the core is located in the 'design' directory. The RTL code is placed in 'rtl' and the 'gate' structure holds the results of the synthesis run.
- The 'verification' and 'regression_suite' directories contain all files needed to run the functional verification of the core. This test is supposed to be run as is and will verify the correctness of the core after changing its configuration for a special project (i.e. cache size, flip-flop or latch-based, clock gating, etc. See also Figure 3).
- The 'synth' directory contains the synthesis scripts for the Synopsys Design Compiler.
- The results of the physical back annotation are placed in the 'pd' directory.
- The golden models used for the functional verification are contained in the 'vmc' directory.
- In the 'config' directory is the configuration GUI for the core.

Required Tools

A collection of tools is required in order to work with MIPS core deliverables. The basic tools are not included with the delivery but may be obtained free of charge under an open license from third-party vendors. They consist of:

- **Tar:** In order to unpack the deliverables files, a GNU version of the standard tar must be used. Current releases are generated using GNU tar version 1.12. Tar can be downloaded from http://www.gnu.org/.
- **Perl:** A Perl version of at least 5.004 is required for some of the scripts used in the flow. Perl can be downloaded from http://www.perl.com/.

- **Make:** All Makefiles included in a release have been written to work with the GNU make tool (using version 3.77). GNU make can be downloaded from the GNU web page at http://www.gnu.org/.
- **C compiler:** the GNU gcc compiler version 2.8.1 is needed for compiling simulation models. GNU gcc can be downloaded from the GNU web page at http://www.gnu.org/.
- **Gzip and gunzip:** In order to run the supplied regressions, these GNU tools must be available in the PATH. This package is available for download from the GNU web page at http://www.gnu.org/.
- **Tcl/Tk:** Some scripts and the GUI elements are written in the Tcl/Tk scripting language. The Tcl/Tk package can be obtained at http://www.scriptics.com/.

In addition, the following commercial EDA tools are directly supported by the MIPS core deliverables:

- **Functional simulation:** The simulation environment supports the following simulators for both RTL and gate level simulation:
    - VCS from Synopsys, version 6.0.
    - Mentor Graphics ModelSim version 5.4e.
    - Cadence NC-Verilog version 3.10.p1.
- **Synthesis:** The synthesis scripts support Synopsys Design Compiler version 2000.11-SP1.
- **Timing:** Static timing analysis using Synopsys PrimeTime version 2000.11-SP1 is supported.
- **Testability:** Scan insertion and ATPG is supported using DFTAdvisor and FastScan, both in version 8.9_1.10 from Mentor Graphics.

Other EDA tools also function with MIPS deliverables but require the customer to adapt the scripts.

Figure 3: Screenshot of the configuration tool window for a MIPS64™ 5K™ core.

Configuration and Functional Testing

A distinct advantage of MIPS synthesizable cores is that they can be configured in a number of ways to meet customer requirements, as opposed to hard cores, which have a

fixed feature set. Configuration of a MIPS core is achieved by means of a GUI like the one in Figure 3. The parameters that can be changed include but are not limited to:

- **Registers:** All registers are instantiated within a wrapper that enables the customer to chose the most effective implementation depending upon the available library cells.
- **Power:** Clock gating can be controlled via these switches.
- **Cache:** The size and configuration of the caches can be freely configured. The total size can vary between 0KB and 64KB, depending upon the cache associativity (1, 2, 3, 4 ways) and the way size (4KB, 8KB, 16KB). MIPS cores feature separate instruction and data caches, which can be configured separately and independently of each other.
- **TLB:** The TLB can be a fixed type (FMT) or programmable. If the programmable option is chosen, the size of the TLB can be set to 16, 32 or 48 entries. It can be configured to be register-based, latch-based or custom built.
- **EJTAG:** This configures the number of instruction and hardware breakpoints included in the core design.
- **BIST:** The memory BIST modules for the cache RAMs are specified and configured here.

Configuration is a required step in the design flow. Because it changes some of the RTL code, a functional verification is needed to validate the changes. The MIPS deliverables contain a cycle- accurate model of the core. This so-called “golden model” is an encrypted VMC model that plugs into any RTL simulator that supports a SWIFT R41 interface. The testbench included in the verification suite compares the golden model against the configured RTL code. The VMC model is compared to the RTL instance during every cycle.

Synthesis and Layout

MIPS core deliverables contain a complete set of scripts for initial synthesis using the Synopsys design compiler. The synthesis scripts are Tcl-based so they can be shared with the static timing analysis tool PrimeTime. The synthesis strategy is modular. This means that every sub block has its own constraint file describing the in and output delays anticipated in the design. A ‘make’ mechanism will trigger a synthesis run. If the Sysnopsys Budgetshell license is available, the synthesis run can be parallelized. The use of parallel synthesis greatly increases the performance because sub blocks in the same hierarchy level can be synthesized in parallel. When the top level is reached, an incremental compile is started to perform final optimizations on inter-module paths. Further optimizations can be done by back annotating results from the layout run, for example.

The synthesis flow must be adapted to every specific process technology. A sample setup using the Artisan TSMC 0.18-micron library is included in each delivery and can be used as a starting point.

Since the layout is a very technology-dependant process, there are no preconfigured scripts for floorplanning and layout tools. However, topics like clocking methodology and floorplanning are covered in the manuals and help the user in the decision process.

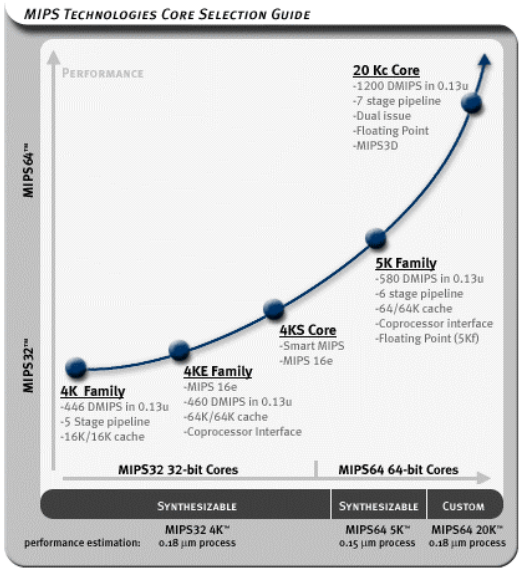

Figure 4: Current MIPS® 32-bit and 64-bit cores (Jan. 2002).

A Complete, Flexible Solution

Clearly, IP cores must be very flexible and need to easily fit into a given design environment. That requires an IP provider that takes into consideration customers' design flows and guarantees the functional correctness of the core, while providing the highest possible performance. The IP vendor should also provide support for common EDA tools and base the flow on openly available tools, thus enabling users to quickly adapt their design flows to their particular needs.

Whether a designer wants to integrate a very small, ultralow-power 32-bit core or a very high-performance 64-bit multimedia core (as shown in Figure 4), MIPS Technologies provides a complete solution for rapid IP deployment that helps ensure the success of SOC designs.