Rapid Design and Exploration of Signal Processing Systems using a VHDL Model Generator Based Paradigm[1]

Scott R. Powell and Thomas M. Cesear

dQdt
5962 La Place Court, Suite 201
Carlsbad, CA 92008
phone: (619)929-9929 fax: (619)929-0280
powell@dqdt.com, cesear@dqdt.com

Abstract

An upgradable design methodology is described for the rapid design and implementation of ASIC-based embedded DSP systems; with results from a detailed design example. The design methodology is based around a library of VHDL model generators which create application-specific simulatable and synthesizable VHDL models of DSP algorithms. The VHDL model generators are extensively parameterized to encapsulate microarchitectural design expertise and elevate it to higher level design activities where there is a greater amount of leverage. Generators provide a cost-effective means to explore the area, speed, and power tradeoffs of algorithmic, architectural, and microarchitectural design alternatives at very early stages in the design process. An expanding library of application-specific, parameterized VHDL model generators is described which is oriented toward common signal processing functions. The library is hierarchical with model complexity ranging from single arithmetic operators to large core functions to entire VLSI chips. The VHDL generator library has been used to design several DSP ASICs; a specific design example is presented demonstrating a 3 to 5 times reduction in overall design time from system specification to mask-level layout.

I. Introduction

Due to rapid advances in IC technology and computer-aided design, it is becoming increasingly practical to design special-purpose VLSI signal processing components to alleviate the computational bottlenecks and speed/area/power penalty imposed by general-purpose processors. VLSI technology continues to follow an exponential advancement (Moore's law) providing increasing gate count and performance to address increasing electronic systems requirements. Design productivity, however, is

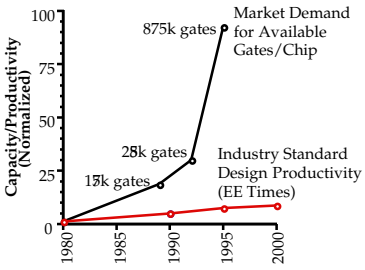


Figure 1: VLSI Manufacturing Capacity and Demand are Growing Much Faster than VLSI Design Capability.

advancing linearly. This is shown in Figure 1. Over the same time span in which VLSI technology has provided a nearly 100x increase in available gates per IC, electronic design automation (EDA) tools have provided less than a 10x improvement in design productivity. This has lead to a substantial gap between available gates and "designable" gates. Thus, new products are particularly vulnerable to premature technology-driven obsolescence.

A unique challenge facing systems/algorithm designers is developing increasingly complex systems within shrinking time-to-market windows for shorter product life-cycles. Further complicating this is the overwhelming abundance of new design alternatives. In mapping an algorithm to a particular architectural implementation, the designer is faced with choosing synchronization methods, resource sharing, numeric formats, task assignments, and low-level operator architecture tradeoffs, which are often independent of the fabrication technology. Each of these choices significantly affect how well the architecture implements the desired algorithm, and consequently, how well the algorithm solves the overall system problem. However, currently available design methodologies are not well suited for cost-effective high-level design exploration to uncover architectures which offer inherently better solutions in the speed-area-power design space.

[1] This work was directed and funded in part by Wright Laboratory Solid State Elect. Dir. SBIR Contract No. F33615-94-C-1432 *Application Specific Electronic Design Synthesis*

Through a cluster of closely related Air Force Small Business Innovative Research (SBIR) projects [1][2][3][4] dQdt has initiated the development of an innovative new design methodology for next-generation IC-based systems. A key feature of this design methodology is support for rapid exploration of area, speed, power and functional tradeoffs at early stages of algorithm and architecture design. This approach is based on the concept of an expanding library of VHDL model generators. These VHDL model generators are extensively parameterized to encapsulate and elevate expert designer knowledge at the architectural and microarchitectural level. Parameters are set or swept to *rapidly create application-specific simulatable and synthesizable VHDL models* of common signal processing functions. This approach extends currently available standard VHDL simulation and synthesis tools to work at higher levels of abstraction to help *bridge the gap between system-level algorithm design and structural implementation*. In Section II, a brief review of the current VHDL-based top-down ASIC design methodology is discussed to provide background. Section III contains a discussion of the VHDL model generator concept, integration into a design methodology, and a prototype library of VHDL model generators for signal processing applications. In Section IV, a detailed application example is described where VHDL model generators were used in the design of a schedule-critical signal processing ASIC with extreme power sensitivity.

II. The Current "Top-Down" Design Methodology

The current "top-down" design paradigm is illustrated in Figure 2 as a process flow. A gap remains between the output of system-level design tools: a verified algorithm, and the input to the top-down ASIC design process: a complete structural-level VHDL or Verilog description of an architecture implementing the algorithm. Bridging this gap is currently a cumbersome and error prone process which represents a significantly large portion of the overall design cycle [5].

Design alternatives such as pipelining, timesharing, numeric format, arithmetic alternatives, testability, and programmability significantly impact the performance, area, and power of the final implementation, *independent of technology*. Experienced logic designers can often recognize and

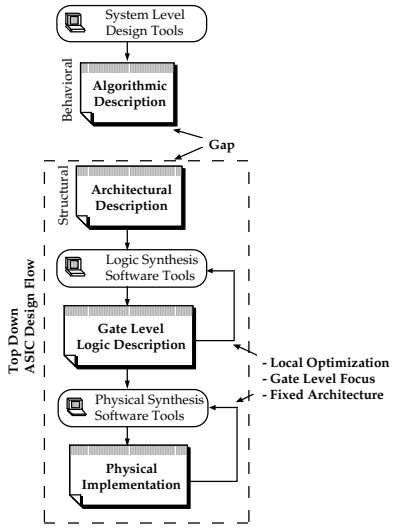


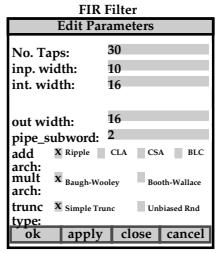
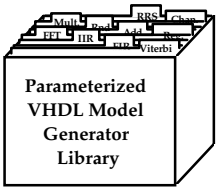
Figure 2: Current "Top-Down" Design Methodology Process Flow.

suggest system changes to select the most appropriate implementation strategy resulting in more efficient architectural implementations. However, there is an increasing reliance on synthesis tools to help manage growing system complexity. These tools offer productivity improvements but tend to omit much of the logic designer's skills, expertise, and feedback from the design process. A further limitation of the current "top-down" design paradigm is the design optimization tends to be highly localized with a gate-level focus.

III. VHDL Model Generators

VHDL offers a vendor, platform, and technology-independent method of describing, documenting, and simulating complex electronic systems. The extensive flexibility of VHDL also makes it suitable for describing function generators: highly parameterized, simulatable and synthesizable VHDL models encompassing whole classes of related application-specific algorithms. *The power of these generators lies in the encapsulation of expert architectural and microarchitectural design knowledge*. Generator parameters provide a mechanism for high-level specification and exploration of application-specific functional (e.g., number of taps, number of channels, pole-zero coefficients, etc.), architectural (e.g., parallelism, pipelining, numeric format, etc.), and microarchitectural (e.g., wordlengths, arithmetic operator implementation, etc.) attributes. When combined with existing VHDL simulation tools, these generators can be used to rapidly explore architectural

alternatives in terms of the overall system's functional performance. Similarly, when combined with existing logic synthesis tools, these generators can be used to accurately explore architectural alternatives in terms of power, area, and speed requirements and to rapidly synthesize to a specific VLSI technology. Parameterized VHDL function generators are, in effect, application-specific design automation tools providing DSP system designers a cost-effective means to explore a multitude of dimensions in the design solution space.

There exists a basic set of arithmetic operators (add, mult, truncate, delay, accumulate, etc.) which can be used to describe almost any DSP algorithm. A library of VHDL model generators of these lower-level functions can be used to create VHDL model generators for increasingly higher-level DSP functions. With this approach, as new microarchitectural alternatives are added to the lower level generators (e.g., a new multiplier micro-architecture), all higher level generators which reference the lower level generator automatically inherit the new microarchitecture as a parameterized option. Parameterized generators of increasingly higher level functions are created by referencing lower level generators in a hierarchical manner.

Various architectural alternatives can be rapidly explored by adjusting the VHDL model generator parameters and evaluating the effects on the functional and physical characteristics of the final implementation using standard VHDL simulation and synthesis tools. *Once the values for each of the available parameters are specified, the generator becomes an application-specific simulatable and synthesizable VHDL model*. As shown in Figure 3, the design solution space can be rapidly explored at very high levels by adjusting the generator parameters. Each point on the family of curves represents a fully synthesized gate-level design. The curves were developed by sweeping the bit-level pipelining (pipe_subword) parameter for three values of internal wordlength.