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

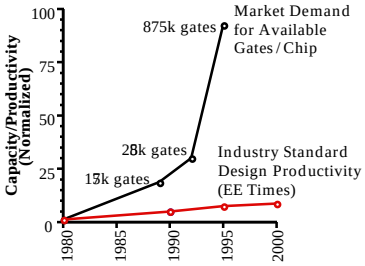

Figure 1: VLSI Manufacturing Capacity and Demand are Growing Much Faster than VLSI Design Capability.

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.

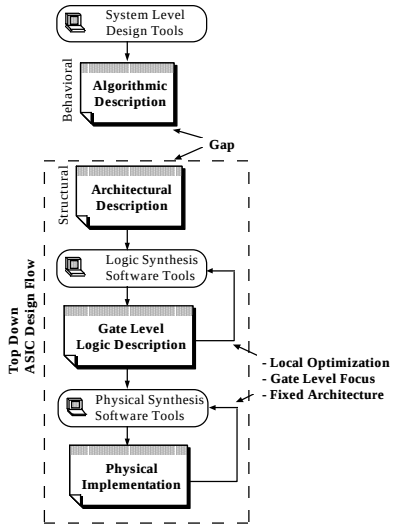

Figure 2: Current "Top-Down" Design Methodology Process Flow.

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 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.

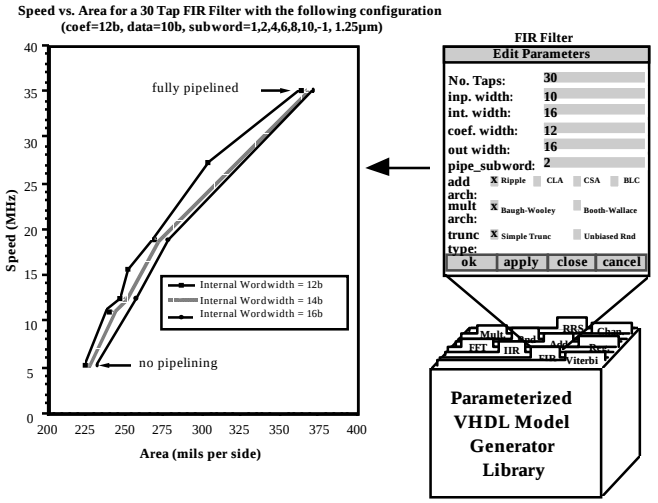

Figure 3: Using VHDL Generators to Explore Bit-Level Pipelining Speed-Area Tradeoffs of a 30 Tap FIR Filter.

The VHDL model generators help bridge the gap between system-level design tools and top-down ASIC design tools. The model generators are similar in function to the system-level building blocks found in system-level design tools (filters, encoder/decoders, transforms, etc.). A prototype integration of VHDL model generators within a system-level design environment has been created in the Ptolemy tool set. Ptolemy is a University of California at Berkeley (UCB) graphically-based design tool which is openly available to the design community (developed with RASSP funding) [6]. Figure 4 illustrates the instantiation, interconnection, and parameterization of several VHDL model generators within the Ptolemy VHDLF domain. Once a block diagram is entered and the parameter menus filled-out, Ptolemy generates a synthesizable and simulatable VHDL model of the block diagram which references lower level VHDL model generators as VHDL component instances. This VHDL model is then synthesized to gate-level for progression to physical place & route activities. Reasonably accurate timing, area, and power estimates are available once a design has progressed to the technology-specific gate-level. Using VHDL model generators integrated with Ptolemy, the time required to enter a block diagram and synthesize to a technology-specific gate-level implementation is on the order of hours; permitting many architectural and microarchitectural alternatives to be explored.

A library of VHDL model generators has been used in the design of seven DSP ASICs. Figure 5 shows the actual design time for these ASICs (from system specification through verified layout) compared to

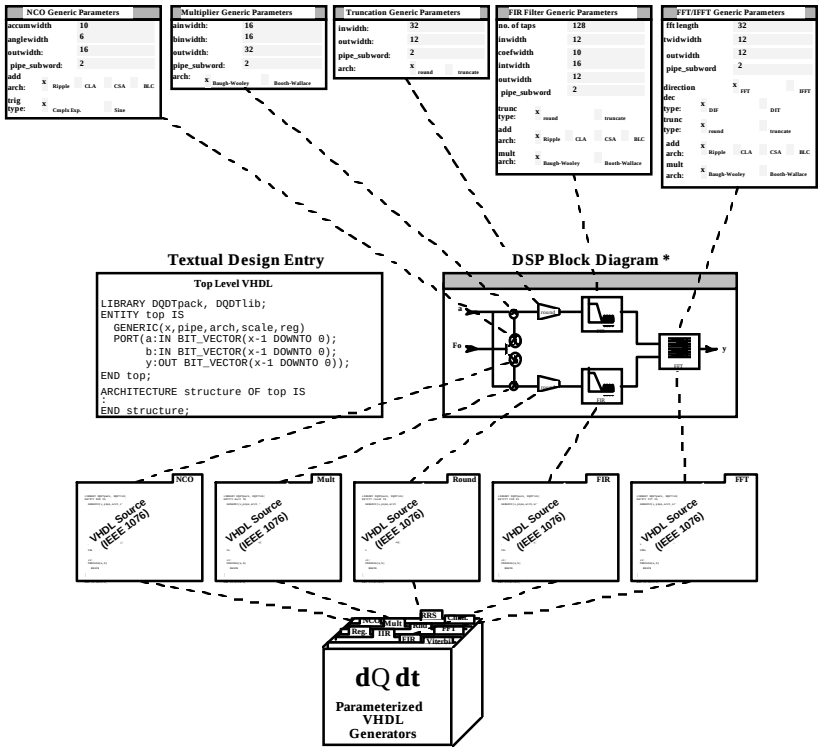

Figure 4: VHDL Model Generators Integrated into the Ptolemy System-Level Design Environment for Rapid High-Level Design Exploration / Synthesis.

industry standard design time; complexities range from 25-75k gates with clock rates ranging from10-125MHz. Using the VHDL model generator design methodology, a *3-10 times reduction in overall design time has been demonstrated* with these ASICs.

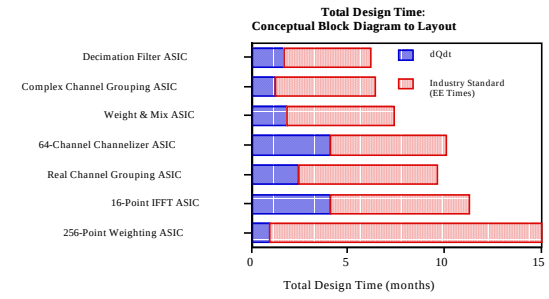

Figure 5: Design Productivity Improvements Demonstrated in Recent ASIC Designs Using VHDL Model Generators.

IV. Design Example: Low Power FFT-Based Digital Channelizer

IVa. Reduced Power Design Techniques

Design techniques exist for almost any signal processing algorithm which permits tradeoffs between area and speed. The most common involve some variation or combination of pipelining, parallelism, and time-shared processing elements (PEs). The performance and density possible with advanced submicron technology enable these techniques to cover an increasingly broader range of functionally equivalent, but physically distinct, implementation alternatives. Recent techniques have been reported which tradeoff excess speed (gained by increasing area) for reduced power dissipation. Power dissipation is a complex function of both area and speed: design tools are needed to select the appropriate amounts and locations of parallelism, pipelining, and / or time-sharing to minimize power dissipation.

Bit-Level Pipelining to Tradeoff Excess Speed for Reduced Power

In CMOS circuits, speed is reduced in direct proportion to a reduction in the power supply voltage (within limits). However, power dissipation is reduced in quadratic proportion to a reduction in the power supply voltage. Thus, as the supply voltage is decreased, power dissipation decreases more rapidly than speed. The University of California at Berkeley (UCB) recently proposed a technique to leverage this relationship where the power dissipation of a CMOS

circuit is reduced by reducing the supply voltage and the speed of the circuit is maintained through pipelining [7]. Traditional pipelining clocks data at word boundaries; bit-level pipelining provides finer granularity by pipelining across word boundaries (hierarchical bit-level pipelining is a feature of the model generators). For applications which are not sensitive to latency, the pipelining granularity of a given architecture is a free design parameter. At some point, however, the power dissipated by the additional pipeline registers is greater than the reduction realized by further supply voltage reduction. The application of this concept to actual designs requires a means to predict chip or system power dissipation at high levels of abstraction.

For a given architecture implemented in a given technology there is an optimum amount of bit-level pipelining beyond which the power dissipation begins to increase, this is shown in Figure 6. To realize

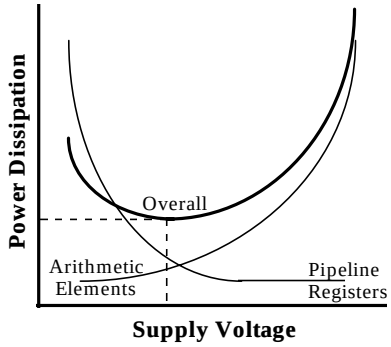

Figure 6: Optimum Bit-Level Pipelining for Reduced Overall Power Dissipation due to the Increasing Power Dissipation Associated with the Pipeline Registers.

the impressive power savings potential of the UCB approach, the chip or system level power dissipation must be determined as a function of pipelining granularity. One of the unique features of dQdt's VHDL model generators is rapid, high-level exploration of the impact of pipelining on signal processing architectures. Pipeline parameters in the generators control the compilation of a VHDL model describing the pipelined chip. Speed and area information about the pipelined architecture is immediately available for any specified target technology using standard CAD synthesis tools. Power dissipation can be determined using standard switch-level techniques. This capability provides the designer with a cost-effective means to quickly iterate the amount of bit-level pipelining to determine the minimum power dissipation.

IVb. Example: Low Power FFT-Based Digital Channelizer

dQdt's parameterized VHDL model generators were used in the design of an FFT-based 64-channel digital channelizer system. The key design requirements were *short* design cycle, *low* power, and investigate *multiple* (new) technologies. A block diagram of the digital channelizer is shown in Figure 7.

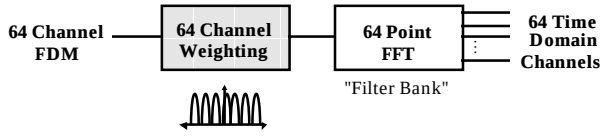

Figure 7: FFT-Based Digital Channelizer System.

A major portion of the design effort was to conduct a low-power architecture study to investigate the tradeoffs between supply voltage, technology, and fine-grained bit-level pipelining for the channel weighting ASIC. A parameterized VHDL-based model generator was developed to conduct the architecture tradeoff.

dQdt proposed a parallel architecture technique in the form of fine-grained bit-level pipelining to maintain throughput. The speed increase obtained from pipelining was traded off to reduce power by operating at reduced voltages. In this way, silicon area (in the form of pipeline registers) was traded off for reduced power consumption. The goal was then to use the parameterized VHDL channel weighting model generator to determine the degree of fine-grain bit-level pipelining which minimized the power dissipation for the specified speed goal. Power was reduced by lowering the supply voltage while using bit-level pipelining to maintain the throughput. At some point, the power consumed by the additional pipeline registers exceeded the power saved. Thus, there was a specific supply voltage which optimized the overall power dissipation for a given technology, as shown in Figure 8.

The channel weighting ASIC required 37k gates. Both the architecture investigation and design of the channel weighting ASIC was completed in 17 man weeks. The design was processed in both 0.8µm DLM and 0.6µm TLM technologies, with *first-time silicon success.* In this design, the parameterized VHDL multi-channel weighting model generator leveraged vast amounts of low-level design information to the architectural-level. With this information dQdt was able to *rapidly* explore the multi-dimensional design

space, allowing an *application-optimized* architecture to be designed.

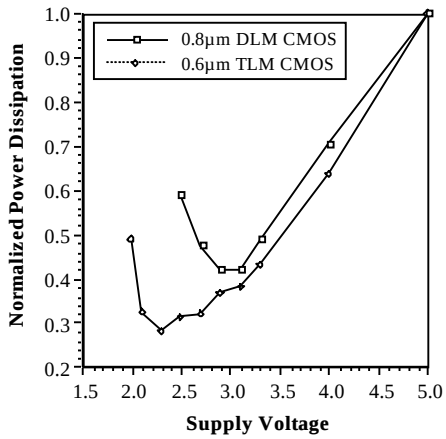

Figure 8: Normalized Power Dissipation as a Function of Supply Voltage (V_{dd}) for the Two Target Technologies.

V. Conclusions

dQdt has developed an innovative new IC-based systems design methodology based on parameterized VHDL model generators. The key feature of this design methodology, in addition to rapid design synthesis, is support for rapid exploration of area, speed, power, and functional tradeoffs at the early stage of algorithm and architecture design.

dQdt's parameterized VHDL model generators enabled the design of a 64-channel weighting ASIC with first-time silicon success. The high-level nature of the parameterized model generators allowed us to investigate 85 complete architecture designs over two target technologies. The generators provided the means to arrive at the optimal low power architecture via trading off reduced supply voltage and bit-level pipelining granularity.

VI. References

[1] S. R. Powell, "Application-Specific Electronic Design Synthesis," Solid State Elect. Dir., Wright-Patterson AFB OF, Final Report AF93-117, November 1993.

[2] S. R. Powell, "Application-Specific Electronic Design Synthesis: VHDL-Based Chip Generators" Solid State Elect. Dir., Wright-Patterson AFB OF, Phase II Proposal AF93-117, November 1993.

[3] W .L. Marking, "Design Automation for Low Power Electronics," Solid State Elect. Dir., Wright-Patterson AFB OF, Final Report AF94-136, January 1995.

[4] W .L. Marking, "Design Automation for Low Power Electronics" Solid State Elect. Dir., Wright-Patterson AFB OF, Phase II Proposal AF94-136, January 1995.

[5] D. P. Noel, T. A. Kwasniewski and A. Castonguay, "Converting DSP Software to Hardware: Automating the Process," Proceedings of DSP World, pp. 830-833, October 1994.

[6] B. Evans, A. Kamas and E. Lee, "Design and Simulation of Heterogeneous Systems using Ptolemy," Proceedings of ARPA RASSP Conference, pp. 97-105, August 1994.

[7] A. P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-Power CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 4, April 1992.