

# MSP Liberator ASIC Design Flow Produces Full Custom Performance Required for Next Generation Military Electronics

**Warren Snapp, Paul Haug, Dave Sunderland, Roger Brees**

The Boeing Company  
1-253-773-0367 / warren.p.snapp@boeing.com

**Don Bouldin**

The University of Tennessee - bouldin@vlsi1.engr.utk.edu

**Carl Sechen**

The University of Washington - sechen@ee.washington.edu

**Wayne Dai**

The University of California at Santa Cruz - dai@cse.ucsc.edu

## Abstract

Next generation military systems and other national security applications require advanced digital signal processing electronics implemented in highly optimized ASICs with Mission Specific Processing Architectures. But there is a problem that currently blocks these systems from fully exploiting the potential of integrated circuit technology. Aerospace ASICs are typically, of economic necessity, designed differently and significantly suboptimally relative to high volume standard products. The project described in this paper, named Liberator, has the objective to develop and demonstrate new ASIC design methodology that will close the gap between Full Custom and ASIC methodologies. The paper will first describe the barriers that must be addressed. It will then describe the technical approach we are following to overcome these barriers. Improvement results for our methods versus standard ASIC design methods will be given. Finally a typical application which is being used to demonstrate the results of this project and the expected system benefits in performance and cost will be described.

## 1. INTRODUCTION

Next generation military systems and other national security applications require advanced digital signal processing electronics to provide revolutionary improvement in functionality and performance. At the same time many such systems are highly mobile and miniaturized. The resulting severe size, weight, and power requirements for the DSP electronics can only effectively be realized in the form of advanced application specific integrated circuits implementing mission specific architectures. Mission specific architectures can realize 100X improvement in GOPS/Watt for a typical DSP algorithm versus implementation in general purpose DSP chips [1]. But there is a problem that currently blocks these systems from fully exploiting the potential of integrated circuit technology. Aerospace ASICs are typically, of economic necessity, designed and implemented differently than high volume standard products such as microprocessors or telecommunications components. Some of the differences are listed in Table 1.

Aerospace ASICs are designed by small design teams using highly automated logic, circuit, and physical synthesis EDA tools. By contrast, standard high volume products are designed by large teams of specialists using more manual or "full custom" methods. This allows a high degree of optimization at all levels including

architecture, logic, circuit, and physical layout design.

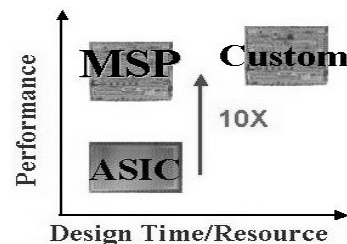
	Aerospace ASICs	Standard Products
Design Teams	Small (1-10)	Large(20-200)
Number of Passes	1	3
Design Capture	High-Level Language	Lower-Level
Automation	High	Medium
Physical Design	Standard Cells	Full Custom

**Table 1.** Different Methods for ASICs vs Custom

Automated synthesis has not been perfected to achieve the level of optimization possible with full custom methods.

In custom approaches architectures can be highly tailored with the use of parallelism, pipelining, optimal clock distribution and noise control, and highly optimized macro functions. Synthesis tools are not yet able to match the custom design team in dealing with these issues. Circuit designers can tune every transistor in a custom design to achieve maximum performance or minimum power. Custom designers can utilize nonstandard logic families such as high speed dynamic logic or clockless logic even if these are not supported by available EDA tools. Physical design can be highly optimized in custom designs with manual polygon pushing squeezing every micron of density out of the technology. And higher density translates to higher performance. Because of high prototyping costs and tight schedules, aerospace designs must work on the first pass and hence are often derated or deoptimized to provide safety margins allowing for inexactly modeled timing and noise issues. The net result is that aerospace ASICs typically suffer from roughly a 10X reduction in optimization as measured in clock speeds, power, and area, relative to full custom integrated circuits [1].

The project described in this paper has the objective to develop and demonstrate a new ASIC design methodology that will close the performance gap between Full Custom and ASIC methodologies. As shown in Figure 1, the goal is to achieve the



**Figure 1.** MSP Goal: Optimality of Custom with Design Efficiency of ASIC

optimization of full-custom while maintaining or even improving on the design efficiencies of an ASIC methodology.

The project consists of the development of several techniques, which when integrated into standard ASIC design tools, promise to improve resulting ASIC performance as measured by metrics such as clock speed, power, or power-delay-area product by a factor of at least 10X. The technical approach we are following has six major elements:

- Algorithms to GDSII Design Flow
- Mission Specific Micro Architecture
- Parameterized Mission Specific Macro Generators
- Multi-Level Optimization Techniques
- Integration Into Commercial Toolsets
- Foundry/Fab Process Flexibility Throughout

The following sections will describe each of these in detail.

## 2. TECHNICAL APPROACH

### 2.1 Algorithms to GDSII Design Flow

The development flow for a typical ASIC design using the MSP methodology is shown in Figure 2. The design flow includes

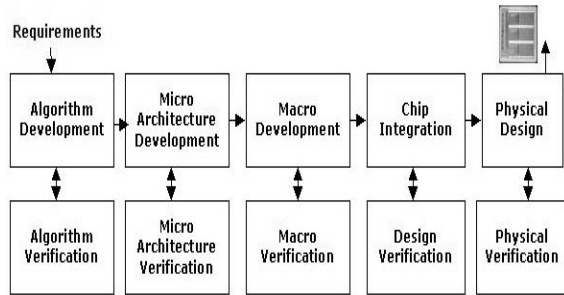


Figure 2. MSP Algorithms to GDSII Design Flow

support for optimization at multiple levels including DSP algorithms, hardware microarchitecture, logic design, circuit design, cell layout, placement and routing. It begins with mission requirements and results in a GDSII layout ready for fabrication in a process such as TSMC or IBM 0.13 micron CMOS. The following sections describe the MSP design flow.

### 2.2 Mission Specific Micro Architecture

Our design of DSP systems begins with the development of processing algorithms based on mission requirements. These algorithms are then translated into highly optimized mission-specific architectures. Figure 3 illustrates key elements of a typical microarchitecture. The algorithm, in this case, is captured in the hardware. The hardware has necessary programmable configurability but is highly tailored to perform the mission algorithm. The hardware is tailored to the point that maximum efficiency is obtained through use of optimum word-width busses as well as the judicious use of pipelining and parallelism. Such mission-specific architectures achieve 100X improvement in performance, as measured by GOPS per watt, over off the shelf programmable DSP chips. A critical element of this approach is an integrated toolset that allows efficient capture and verification of the microarchitecture based on synthesized input data as well as the ability to later validate the detailed design and prove equivalence against the same input scenario.

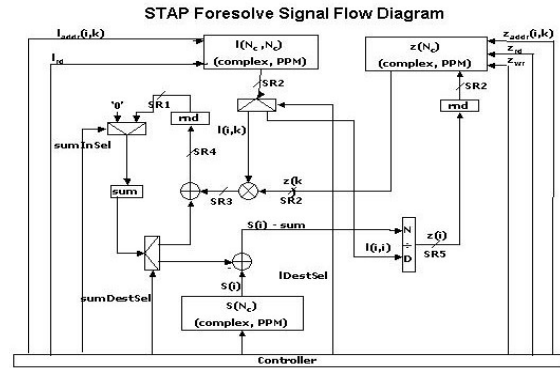


Figure 3. Microarchitecture Design

### 2.3 Parameterized Macro Generators

Once a microarchitecture is developed the various functional blocks must be defined in detail using VHDL. We have developed a library of standard DSP macros so that the design time can be significantly reduced. Furthermore the macros have been developed by experts and are highly optimized and validated. The macros are parameterized so that they can be tailored to a mission specific microarchitecture. Some of the macros being developed for this program are listed in Table 2.

MSP Macros	
Complex Multiplier	Divide
Configurable Rounder	Square Root
Output Gain Section	1-Pole IIR Filter
Decimating Cascaded Int Comb	2-Pole IIR Filter
Windowed Integrator	Numerically Controlled Oscillator
Fixed Rounder	Decimated FIR Filter
Finite Impulse Response Filter	2D Linear Filter
Fast Fourier Transform	Cholesky Factorization
Polyphase Filter	Covariance Matrix Estimation
Hilbert Transform	Fore-Solve/Back-Solve

Table 2. Macros List

Figure 4 shows details of one macro, a FIR filter. It shows the block diagram, available parameters, and the user interface that is available to the designer.

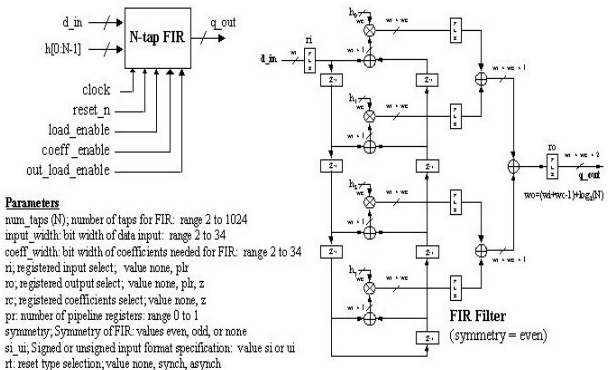


Figure 4. Parameterized FIR Filter Macro

### 2.4 Multi-Level Optimization Techniques

The differences between ASIC design and full custom that leads to the performance gap occurs at all levels in the design process.

Therefore our approach to optimization involves development of techniques to incorporate at all levels. Figure 5 lists the main

<u>Design Level</u>	<u>Optimization Technique</u>
Architecture	Architectural Optimization
	Domain Specific Macros
Logic	Logic Synthesis Optimization
	Power Compiler
	Ultra
	Gain-Based Synthesis
Circuit	Optimal Skew
	Circuit Optimization
	Transistor Tuning
	Alternative Logic Families
	Output Prediction Logic
Layout	Variable $V_t$ , $V_{DD}$ , Body Bias
	Regularity-Based Placement
	Domain Specific Macros

**Figure 5.** Multilevel Optimization Techniques

optimization techniques that are in development and the level in the design process at which they apply.

Architectural optimization techniques are supported by integration of algorithm simulation tools based on MatLab, multi-chip system simulation tools that are clock and bit accurate based on C modeling, and logic level simulation. By integration of these modeling levels with instrumentation and macros, microarchitectures are highly optimized to specific algorithms and are guaranteed to work at the system level on first pass.

Domain Specific Macros are DSP functions and arithmetic elements that are custom designed using custom-like design optimization and captured into foundry independent parameterized generators that generate layout level hard macros.

Logic synthesis optimization includes a variety of techniques involving accessories and enhancements to commercially available synthesis tools. They include scripts to automate searches of design space to find the optimum design points based on system requirements. They involve the use of standard synthesis tools such as Synopsys Design Compiler, Power Compiler, and DC-Ultra. Also under investigation are the use of Gain-Based Synthesis [2] and the use of Optimal Skew to remove timing slack.

Circuit Optimization techniques involve optimizing transistor-level topologies and device sizing (transistor tuning) to reach the ideal tradeoff between speed, power and area. We have used commercially available tools to evaluate cost functions and then feed back optimized circuits into the physical design process.

Alternative logic families such as Output Prediction Logic (OPL) [3] are being incorporated into the libraries. These can provide additional optimization, mainly speed in the case of OPL, over that available from the standard CMOS logic.

Variable  $V_t$ ,  $V_{DD}$ , and Body Bias techniques are being explored that allow further optimization. For example high  $V_t$  transistors can be used to minimize leakage/power and low  $V_t$  devices can be used to optimize speed on critical path circuits at the expense of increased static power. Variable  $V_{DD}$  can also be used to trade power for speed. Variable body bias can be used to dynamically adjust transistor thresholds.

Regularity-Based Placement is used to exploit the logical regularity of DSP datapath designs, e.g., arithmetic functions connected by multi-bit buses, by mapping it to a corresponding regularity in the physical design which can result in improved density, speed, and power relative to more random standard placement.

## 2.5 Integration Into Commercial Toolsets

All of the above techniques are embodied in the form of scripts and library generators that integrate into standard ASIC EDA tools. The primary synthesis tools are the Synopsys toolset, the primary simulation tool used is ModelSim, the physical design is done using Cadence, sizing and analysis tools include Synopsys AMPS and NanoSim and Celestry Nautilus VT. Also we are using automated library generator tools from Prolific and Circuit Semantics. A user interface called Navigator, is used to manage the design and guide the designer through the flow. These scripts, libraries, and integrated toolsets are available, for use by the US Government and its contractors, at the Boeing ASIC Design Center.

## 2.6 Foundry/Fab Process Flexibility

Because of the dynamic nature of integrated circuit foundries available to low volume DoD applications as well as the continuous evolution of ASIC processes, the Liberator methodology is designed to be process flexible. This means that circuit optimizations, library generators, and all essential design tools are driven by process technology files. The intellectual property elements of Liberator are generally technology independent, e.g. circuit topologies or VHDL, and hence can be used across many generations of processes with little maintenance.

## 3. OPTIMIZATION RESULTS

To evaluate the optimization techniques several benchmark macros were selected. Each macro was synthesized and laid out using an Artisan standard cell library targeted to TSMC 0.18 micron CMOS using standard Cadence automatic layout tools. These became the baseline designs to which our optimization techniques are compared. The baseline designs represent today's standard ASIC design process.

Compound Optimization	Optimization Technique														Optimization Techniques	Lib.	
	A	B	C	D	E	F	G	H	I	J	K	L	M				
Baseline	X															A Design Compiler Scripts	Artisan
1	X	X														B Power Compiler Scripts	Artisan
2	X		X			X										C Synopsys Optimization	C
3				X		X										D DSP Synthesis	C
4	X		X			X	X									E Output Prediction Logic	C
5				X		X	X									F Transistor Sizing	C
6								X								G $V_{DD}$ Optimization	C
7								X	X							H Custom Cell Netlist	Artisan
8								X	X	X						I Library Cell Generator	C
9	X											X				J Regularity Placement	C
10													X	X		K Useful Skew	Artisan
																L VHDL Code Opt.	Artisan
																M DC-Ultra Scripts	Artisan

**Figure 6.** Compound Optimization Techniques

Several compound optimizations (multiple techniques listed in Figure 5) were then applied to the benchmark macros. A key that explains the compound optimizations is shown in Figure 6.

A variety of points in design space were targeted including minimum delay, minimum power at the testbench frequency, and minimum power-delay-area (PDA) product.

The resulting post layout designs were "measured" for power, delay, and area using Celestry and Synopsys tools. Several results of the benchmark comparisons are shown in Figure 7.

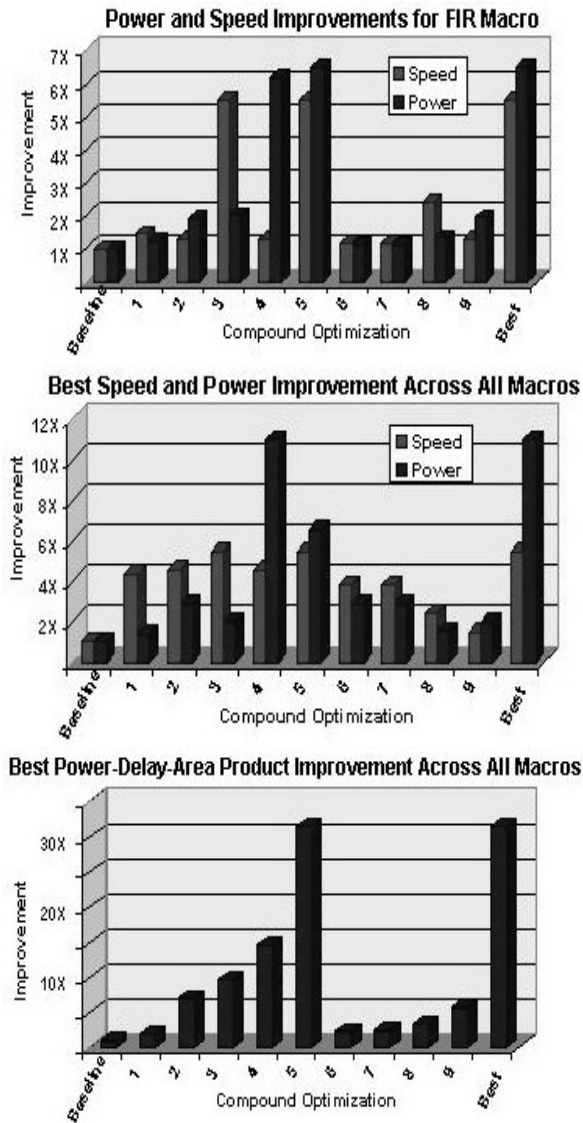


Figure 7. Optimization Results

The benchmark results show that  $V_{dd}$  optimizations can yield a significant improvement in power with little or no impact on

performance by reducing the power of off-critical-path circuits. Comparing the results between compound optimization 2 and 4 or between 3 and 5 shows the improvement obtained. Comparing compound optimizations 2 and 3 or 4 and 5 shows that another technique, DSP synthesis, yields tremendous improvements. This technique employs more efficient adder and multiplier structures, typically having more operands than usual, and carry deferment strategies to significantly improve DSP-type structures.

Comparing the optimized results to the baseline we are achieving typically 10X to 30X improvement in PDA. This allows gap closure to approach the optimization of a full custom design process while preserving the automation and design efficiency of ASIC design. Additionally, the reuse of macros saves 2-3 staff-months and the optimization scripts save 1-2 staff-months. Hence, we achieve an overall reduction in design time even though it takes extra time to run these optimization tools.

#### 4. DEMONSTRATION APPLICATION

The MSP design flow can be applied to a wide range of DSP and other digital processing applications, where high performance and integration is required. The macros developed were selected to span important algorithms commonly performed in communication, radar, navigation, targeting, electronic warfare and other military and national security applications. Boeing is demonstrating MSP Design Methodology on the design of a Mission Specific Processor that performs Space Time Adaptive Processing. This processor will allow advanced functionality and performance in a number of platforms including radar and infrared sensors. The STAP processor chip will, after application of MSP optimization techniques perform 200 GOPS/Watt. The design is being targeted to a 0.13 micron CMOS technology. It will have about 2 Million Gates and 36 Mbit onchip RAM. The same algorithm, if performed on Tiger SHARC DSP chips would require 50 chips operating in parallel and would not be cost effective nor meet size weight and power requirements in highly miniaturized and mobile applications.

#### 5. ACKNOWLEDGEMENTS

The work described in this paper was sponsored by the Defense Advanced Research Projects Agency (DARPA) the Air Force Research Laboratory (AFRL) and the Defense MicroElectronics Activity (DMEA) under contracts number F30602-01-2-0562 and DMEA 90-99-D-0005-0002.

#### 6. REFERENCES

- [1] Chinnery, D., Keutzer, K., *Closing the Gap Between ASIC and Custom*, Kluwer Academic Publishers, 2002.
- [2] Sutherland, I., Sproull, B., Harris, D., *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufman Publishers, 1999.
- [3] Kio, S., McMurchie, L., Sechen, C., "Application of Output Prediction Logic to Differential CMOS," IEEE Computer Society Workshop on VLSI 2001