

## AN APPLICATION SPECIFIC DSP FOR SPEECH APPLICATIONS

Shaul Berger, Yair Be'ery, Bat-Sheva Ovadia, Ronen Peretz, Gideon Wertheizer, and Yael Gross  
DSP Semiconductors, Inc.  
2855 Kifer Road, Santa Clara, CA 95051

### ABSTRACT

A 16-bit Digital Signal Processor Core, designed for speech applications in telecommunications and consumer electronics is described. It enables low-cost, low-power DSP processing with several levels of modularity, permitting efficient DSP-based ASIC development. The DSP core can perform speech compression for applications such as digital answering machines and cellular phones.

### INTRODUCTION

In recent years we have seen a proliferation of compact, powerful consumer electronic devices, including CD players and cellular telephone products. These devices push the boundaries of both performance and miniaturization, and, combined with the cost constraints of the consumer markets, require cost-efficient, performance-optimized DSP hardware and sophisticated, application-tuned DSP algorithms.

A major segment of the DSP IC market, that is addressing these needs, is that of function- and algorithm-specific ICs or FASICs. They may be custom, standard-cell or gate-array implementations optimized to perform a specific function. The existing FASIC design methodology has many limitations. The ever shortening design

cycles are more and more demanding from the designers to move functionality from hard-wired algorithms to software algorithms. For every application the designer had to design a special FASIC instead of using a DSP core that can support broad range of applications and allows the designer to re-use existing and field proven algorithms. Silicon compilers are not efficient enough to provide cost effective FASICs. Many of the emerging applications have different operating modes, each using different algorithm, making the traditional FASIC approach not cost effective. The disadvantage of the general purpose DSP solutions is that they are not cost effective since they are not optimized per application. The approach that is presented, using a DSP core, has all the advantages of both approaches. The DSP core can be seen as general purpose DSP from the aspects of processing power, instruction set and architecture, flexibility and development tools. On other hand core-based design provides the designer with all the benefits of the FASIC approach - low cost optimized solution per application

The 16-bit Digital Signal Processor core is designed for communications and consumer electronics products. It enables low-cost, low-power DSP processing with several levels of modularity in the RAM, ROM and I/O, permitting efficient DSP-based ASIC development. It's design was application-

driven, based on extensive analysis of the target applications and the instructions and architectural features essential to those applications. The main target applications were speech/audio processing and personal communications. The instruction set was enhanced with instructions to support microcontroller applications.

The DSP core's modular design approach allows the same DSP core to be used for various applications simply by adding on-chip memory, peripherals and custom logic. To quickly move from concept to silicon requires the right development tools, especially software development tools, which support this architectural approach.

## ARCHITECTURE

The 16-bit DSP core is using double metal CMOS technology to achieve performance of 25 ns cycle time @ 5V and 33 ns cycle time @ 3.3V. The architecture supports high level of modularity with expandable data and program memory, user defined registers and stack. This architecture is supported by advanced development tools. The main blocks of the DSP core are computation unit, which includes ALU, multiplier and accumulators, address arithmetic unit, and program control unit. All other peripheral blocks including program and data memory, which are application-specific, are defined as part of the user-specific logic, implemented around the core on the same silicon die.

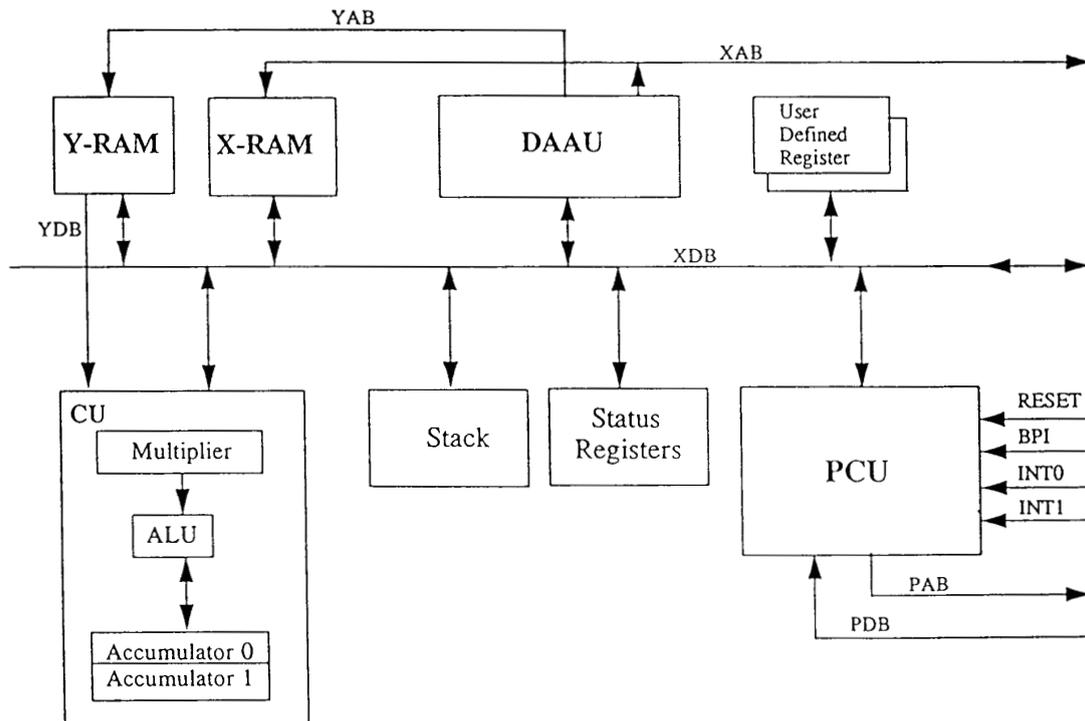


Figure 1. DSP core Block Diagram

Data is transferred on X Data Bus (XDB), Y Data Bus (YDB) and the Program Data bus (PDB). The X Address Bus (XAB) and the Y Address Bus (YAB) drive the addresses for XRAM and YRAM. The Program Address Bus (PAB) drives the program memory addresses.

The Data Address Arithmetic Unit (DAAU) performs all address calculations necessary to address data operands in data and program memories. In addition, it supports loop counter operations. This unit operates in parallel with other core resources to minimize address generation overhead. The DAAU can implement linear and modulo arithmetic. The DAAU can generate two addresses every instruction cycle which can be post-modified by two modifiers: linear and modulo. The address modifiers allow the creation of data structures in memory for circular buffers, delay lines, FIFOs, etc.

The Computation Unit (CU) consists of a multiplier and data ALU. The multiplier unit consists of a 16x16 bit single-cycle, non-pipelined multiplier, two 16-bit input registers (X and Y), a 32-bit output register (P), and an output shifter. Together with the Data ALU, the DSP core can perform a single-cycle Multiply-Accumulate (MAC) instruction.

The Data ALU performs all arithmetic, logical and shifting operations on data operands in single cycle. The Data ALU consists of a 36-bit, single-cycle, non-pipelined ALU unit, two 36-bit accumulator registers, and a saturation unit. The ALU unit supports normalization, division and rounding. It uses a 2's complement arithmetic. Each accumulator is organized as two regular 16-bit registers and a 4-bit extension nibble for protection against 32-bit overflow. Saturation arithmetic is

provided to selectively limit overflow when reading accumulators.

The Program Control Unit (PCU) performs instruction fetch, instruction decoding, exception handling, and wait state support. The PCU generates the next address to the program memory and controls hardware loops with no overhead. The program controller implements a three-level pipeline architecture. In the operation of the pipeline, concurrent fetch, operand fetch and execution occur. This allows instruction execution to overlap. Thus, the effective execution time for most instructions is one cycle. Each pipeline stage is completed before its result is needed by the next instruction. The pipeline is an "interlocking" pipeline, transparent to the user, which simplifies programming.

Two independent 64 Kword memory spaces are available: the data space (XRAM and YRAM) and the program space (PROM). The data space is divided into an X data space for the XRAM, and a Y data space for the YRAM. The internal XRAM and YRAM sizes can be expanded in 2x64 word blocks up to 2x1K words. The XRAM and YRAM can also be expanded by data ROM. The X data memory can be expanded off-core (with 0 wait states). The DSP core peripherals are memory mapped I/O into the data space and are depending on the ASDSP (Application Specific DSP) configuration. The internal PROM is at least 4K words and can be extended in 1K word blocks up to 32K words. The program space may be expanded off-chip up to 64K words.

The DSP core supports eight optional user-definable registers, which can be located off-core. These registers appear in the data register fields of all relevant instructions.

With these registers, external computation units can be loaded with data and read at the end of the computation directly into internal registers in a single cycle. Operations such as parity calculation, location of first-1-bit in a word and min/max/mid value can be easily performed in parallel in a few cycles with the DSP core.

The DSP core was designed with special consideration for reducing the power dissipation without affecting the performance. The DSP core supports two operating modes which reduce further the power dissipation - SLOW and STOP modes. The DSP core has wide voltage operating range from 3V up to 5.5V. Since the core VLSI design is fully static, the clock can be stopped by setting the STP bit. The interrupt signal will reactivate the core. The RAM content and all registers which are not defined as being affected by the reset will remain unaffected. The core clock can be slowed down by writing a 4-bit value to a special memory-mapped register located external to the core, thus reducing the power dissipation. This feature is very useful for applications which have different processing loads in various operation states or modes.

The 16-bit instruction encoding has been optimized to support the highest parallelism allowed by the architecture. Several of the most common DSP benchmarks are shown in Table 1. The instructions fall into 5 groups: arithmetic and logical, multiply and multiply-accumulate, move, branch and call, loop and control

N taps FIR filter	N x 25ns
N taps LMS	4N x 25ns
N taps complex FIR filter	4N x 25ns
N Cascaded IIR Biquads	5N x 25ns

Table 1. DSP Core Benchmarks

## SPEECH APPLICATIONS

The DSP core architecture, which provides powerful DSP while keeping low power dissipation, is attractive to many speech processing applications such as digital answering machines, mobile computing, cellular phones, wireless PBX and Personal Computers (PCs). Following are two applications which benefit from this architecture.

### Digital Telephone Answering Machines

Within the last decade, the Telephone Answering Device, or TAD, has become an integral part of everyday communications. Rather than storing messages on audio tapes, all-digital TADs receive the incoming messages, compress them for greater storage efficiency, and store them on memory chips. The digital answering machine brings the voice mail features such as selective delete and multiple mailboxes. Another advantage of digital answering machine is the increased reliability over the mechanical tape.

The enabling technology for the all-digital TAD is DSP implementation of speech compression algorithms. The ADPCM compression algorithm is relatively simple to implement and lends itself to variable rates of 16, 24 or 32 kbps (although voice quality at rates below 24 kbps is unacceptable). DSP-based compression algorithms are more complex but provide higher compression with comparable voice quality. For example, in order to implement 15 minutes of recording using sophisticated speech compression algorithm requires 4 Meg DRAM at 4.5 Kbps, while ADPCM compression requires more than 22 Meg DRAM memory at 24 Kbps. Following is block diagram of ASDSP for TAD.

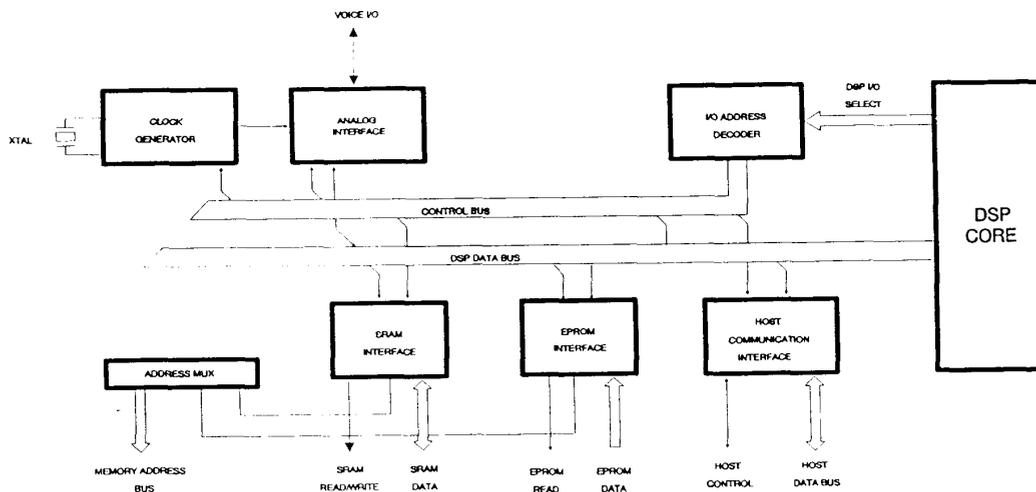


Figure 2: DSP Core based Application Specific TAD

The next logical step in the evolution of the DSP-based phone center (TAD/featurephone/cordless phone) is the integration of additional office equipment functions such as facsimile, into a single office communication unit. All necessary DSP algorithms would be combined on a single multi-functional ASIC DSP chip, reducing the component count, power consumption, cost and overall size of the end-product, while increasing reliability.

**Cellular Communications**

The new digital cellular standards call for state-of-the-art speech coding, channel coding and modem techniques needed to implement speech coders such as VSELP or half-rate GSM, channel coders and modems such as DQPSK. The speech and channel coders can be realized most efficiently with

DSP techniques. Similarly, the modulation/demodulation schemes set forth in the various standards, are best implemented with DSP-based solutions.

Integrating all these functions in a single application specific DSP, tightly tailored to meet the customer requirements will provide the best solution at lowest possible cost. Additionally, the reduced battery power requirements will extend battery life, or, alternatively, permit the batteries to be smaller. Other capabilities, soon to appear on the market, including acoustical echo cancellation, voice-activated dialing and noise reduction are all DSP-based and can be easily integrated either by adding more software (more program memory) or adding more on-chip peripherals to support these functions. Following is block diagram of application specific cellular speech/data processor.

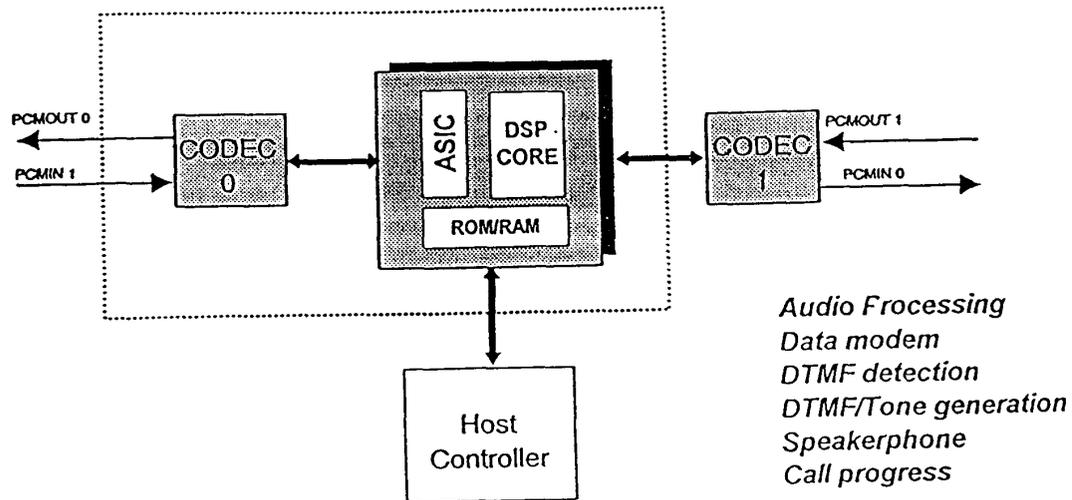


Figure 3: DSP Core based Application Specific Cellular Speech/Data Processor

#### SUMMARY

This paper presented a novel approach to a design of speech applications. The application specific DSP core approach fills the growing need for highly optimized, modular architecture, combined with a complete set of development tools, that permits designers not only to achieve their performance, cost, size and power goals, but to meet the ever shrinking windows of opportunity in today's consumer electronics markets.



**Shaul Berger** received his BSEE in 1973 and MSEE in 1979 from Technion, Haifa, Israel. Shaul Berger joined DSP Semiconductors in 1990 and was promoted to the position of Vice President of Marketing and Sales in 1992. Prior to joining DSP

Semiconductors, Mr. Berger worked for Rafael as director of digital signal processing, for Motorola Semiconductor designing DSPs and Daisy Systems leading CAE design. Mr. Berger has over 15 years of experience in development and application of digital signal processing.



**Yair Be'ery** received his B.Sc., M.Sc., and Ph.D. degrees all in Electrical Engineering from Tel-Aviv University, Israel, in 1979, 1980 and 1986, respectively. He is a Senior Lecturer in the Department of Electrical Engineering of Tel-Aviv University, and Vice

President of Advanced Technology of DSP Semiconductors. He joined the Academic Staff of Tel-Aviv University in 1985 where he is conducting research in coding and communication theory, and in VLSI processing arrays. In 1990 he joined DSP Semiconductors, as Director of VLSI Architectures and led the development of the PINE DSP core.



**Bat-Sheva Ovadia** graduated from the Technion Israel institute of Technology with a BSc EE in 1988. Before graduation she served at the Israel Defence Forces as an Electronics technician. She joined the DSP Group in 1988 as a real-time programmer in

DSP applications and as an hardware engineer. She currently works on DSP architecture development.



**Gideon Wertheizer** received a BSEE from Ben Gurion University, Israel, in 1982. He is a Project Manager for the PINE based family of products in DSP Semiconductors. He joined DSP Semiconductors in 1990. Before that he worked for National Semiconductors

LTD. He currently works on DSP architecture development.



**Yael Gross** received a B.Sc and M.Sc in Electrical Engineering from Tel-Aviv University, Israel, in 1984 and 1988 respectively. She joined DSP Semiconductors in 1990 and worked on DSP architecture development and on DSP chip design. In 1992 she became a

Project Manager of a DSP based FASIC chip design.



**Ronen Perets** received his B.Sc in Electrical Engineering in 1987 from Tel-Aviv University, Israel. He joined DSP Semiconductors in 1990 as a VLSI design engineer, and participated in PINE DSP architecture definition and design. Currently he is

working with DSP Architecture Group and pursuing his M.Sc. thesis at the Technion, Israel.