

An Embedded Reconfigurable SIMD DSP with Capability of Dimension-Controllable Vector Processing

Liang Han¹, Jie Chen², Chaoxian Zhou, Ying Li, Xin Zhang, Zhibi Liu, Xiaoyun Wei, and Baofeng Li

Institute of Microelectronics, Chinese Academy of Science
Email²: jchen@ime.ac.cn, Email¹:leonardhan@sina.com.cn

Abstract

A programmable parallel digital signal processor (DSP) core for embedded applications is presented which combines the concepts of single instruction stream over multiple data streams (SIMD) and reconfigurable architecture. Equipped with eight SIMD-controlled 16-bit datapaths which can also be reconfigured as two 32-bit datapaths, the DSP core can process both 16-bit and 32-bit data in parallel, showing high performance, especially in the applications preferring parallel data flow computations, such as image processing. The SIMD scheme is extended with the instant-scalability of datapaths (ISSIMD), which offers the DSP a capability of dimension-controllable vector processing, so that to provide flexibility for different embedded applications. A first prototype in 0.18- μm CMOS technology has been fabricated, which achieves 1GMACS performance at the clock of 125MHz.

1. Introduction

With the fast development of digital products, new applications and algorithms have caused a still growing demand for processing power and data throughput in today's signal-processor architectures. The most popular approach to improve the performance of DSP is to increase the frequency of operating clock with high endeavors. The state-of-the-art DSP chip, such as TMS320C6X, can operate at 1GHz owing to its elaborated custom design [1]. However, this kind of ideas is limited for the abandon of custom design in embedded DSPs which especially emphasize technology independence and synthesizability. With the development of advanced deep sub-micro process technologies, more and more transistors can now be integrated into a single circuit, leading to an age of implementing a complicated

system on a chip (SOC). Under such circumstance, the traditional approach to improve the performance of the processor by merely increasing the operating frequency with great endeavors seems no longer a wise choice, since higher performance can also, or more easily, be attained by effectively employing more parallel function units.

The DSP described in this paper is designed mainly for multimedia applications, such as image processing. A dominant share of these applications' processing power is required by the "low-level algorithms", which feature a predetermined data-independent control and data flow. For example, for MPEG-2, the share of low-level processing exceeds 80%, and for the recent MPEG-4 standard it is still on the order of 60% [2]. Data level parallelism is not only well adapted to these kinds of algorithms, but also the largest parallel resource by far in these kinds of applications. Therefore, the architecture of the DSP described in this paper focuses on data-level parallelism by employing multi-way SIMD-controlled processing units. Since SIMD machines works best in dealing with arrays in for-loops [3] which are employed in large amount in these kinds of algorithms, such architecture can take advantage of DSP parallelism and deliver maximum performance. The first implementation of the DSP employs eight 16-bit datapaths. At 125MHz, it provides 1 billion 16-bit MACs per second needed in applications that demand significant DSP performance such as multi-channel infrastructure equipment. This is also the case for some "client" applications such as 3G wireless handsets where the DSP will also be used for new features like MPEG4 video, MP3 audio, Voice over Internet Protocol (VoIP) for low-cost long distance voice, and JPEG2000 still pictures [4].

On the other hand, scalable hardware architecture and flexible software controllability are more and more emphasized for embedded DSPs, especially in such an age of SOC, so that they can fast and easily adapt to different applications. To attain a relatively high flexibility, a kind

of instantly scalable SIMD (ISSIMD) scheme is introduced in this paper to enable the DSP process array or vectors whose dimensions can be easily influenced by software. With this scheme, each of the parallel ISSIMD-controlled processing units can either be picked out at will to form a vector datapath or be “demobilized” at any moment under the control of instruction. Furthermore, the 32-bit datapaths reconfigured from several 16-bit processing units at nearly no extra area costs additionally supply the DSP with a capability of high bit-width data processing. All these scalability and

reconfigurable features enable the DSP core adapt to different embedded applications instantly.

The remainder of this paper is organized as follows. In Section 2, the architecture of the DSP core in terms of available parallelization resources, reconfigurable processing units, and control mechanism for scalability is presented. Implementation details of the design are described in Section 3, while performance data of the DSP are discussed in Section 4. Section 5 concludes this paper.

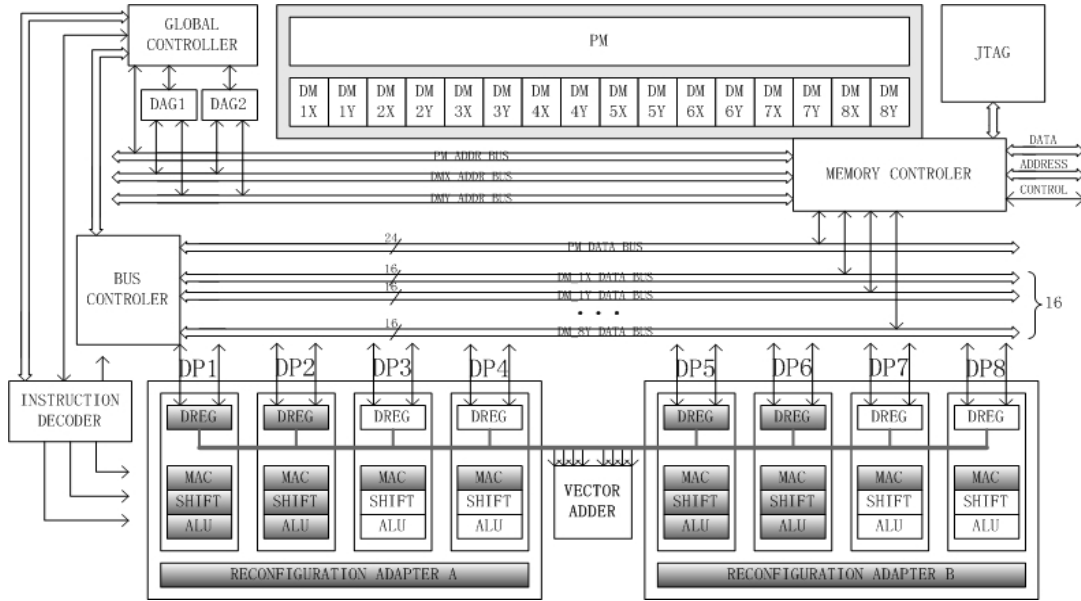


Figure 1. Block diagram of the DSP architecture

2. Processor Architecture

2.1. Overview

The DSP is a modified Harvard architecture based processor, as is shown in Figure 1 and described in [5]. The homogenous core consists of a scalable array of identical fixed-point processing units. In this paper, a first prototype with eight 16-bit parallel processing units is presented, which could also be reconfigured as two 32-bit parallel datapaths with the assistance of local reconfigure adapters. In the next generation of the DSP, the 16-bit arithmetic units will also be splittable to allow 8-bit sub-word parallelism. This will be well applicable, as algorithms of image processing mainly operate on 8- or 16-bit.

These processing units are centrally (SIMD) controlled by a RISC control unit and simultaneously operate on a same 24-bit instruction, which is read from program memory (PM). The SIMD controlling principle leads to low hardware implementation costs compared with multiple-instruction multiple-data (MIMD) [6] or multithreaded approaches.

Additionally, a vector adder is introduced to support a fast operation of adding up all data from selected datapaths in a single cycle. Such a function is very effective in fast implementation of FIR.

The DSP has enhanced scalability and flexibility compared with common SIMD processors. Number, functionality, and data-width of the parallel processing units to be found in a datapath are derived from the targeted applications and are fully controllable by users. Several dedicated instructions are employed to switch the processing modes, which determine the datapath to be

configured as 16- or 32-bit one, processing scalar or vector. Users can even select exactly which one or several of these parallel processing units will actually contribute to the coming instructions. With this ISSIMD controlling scheme, the DSP can flexibly process vectors with different dimensions in a same program, and thus broaden the arithmetic space for certain applications.

To achieve high clock rates, memory access is pipelined into two instruction cycles. Thus, based on the typical three-stage RISC pipeline, the entire architecture features a six-stage pipelining scheme with instruction address generate, pre-fetch, fetch, data address generate, instruction decode, and execute phases. All sub-modules are organized according to this scheme.

2.2. Processing Unit and Datapath

Figure 2 shows the architecture of a 16-bit processing unit. It consists of a 16-bit ALU, a 16-bit (40-bit result) multiply/accumulator (MAC) and a 16-bit shifter with parallel access on a local data register file (DREG) including 16 16-bit (mainly special-purpose) registers. The last stage of the pipeline begins with reading operands from registers, executes operations according to the instruction, and ends up with writing back the results into DREG. All arithmetic units operate in this execute phase and finish computations in one cycle.

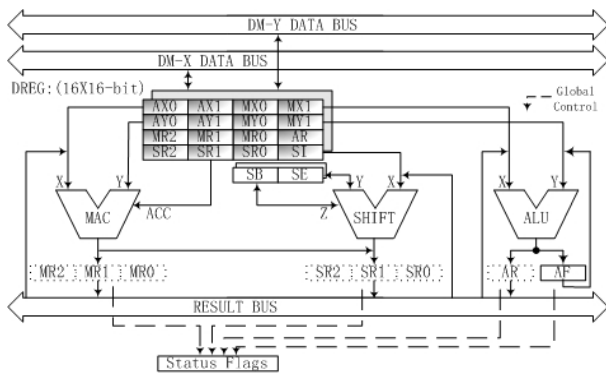


Figure 2. 16-bit processing unit architecture

Since the most critical timing path of the DSP lies in the MAC, great endeavors are paid to speed it up -- fast implementations were chosen and manual synthesis method is partially introduced. The MAC consists of a 17x17 multiplier with a radix-4 Booth encoder [7] and a Wallace tree compressor [8] followed by an adapted fast carry look ahead (fast CLA) adder, where the two outputs of the Wallace tree and the optional value in 40-bit accumulator register can be added. Both decimal mode

and integer mode are supported by the multiplier, the factors of which can be individually interpreted as signed or unsigned values. Both cumulatively addition and cumulatively subtraction are supported. The operations of rounding and saturating are optional.

Shift operations are implemented with a barrel-shifter, which mainly accomplishes functions of arithmetic shift, logical shift, and normalization. The shifter also performs derivation of exponent or that of common exponent for an entire block of numbers. These functions can be combined to implement numerical format control, including full floating-point representation. The ALU is a typical one, implemented with more considerations on low power and small area rather than high speed, as is the shifter, since neither of them is a timing-critical component.

DREG is used as source and destination of the arithmetic units, also as the interface to the memory. A secondary register set is employed to support fast interrupt switch. To achieve acceptable clock rates, the number of DREG ports has been reduced to a minimum, so as to reduce the complexity and thus the delay of the crossbar connecting DREG and arithmetic units. In order to allow an unrestricted parallel access, all registers can be read through one 40-bit and three 16-bit ports and be written through one 40-bit and two 16-bit ports, respectively.

The arithmetic units are reconfigurable -- every four 16-bit processing units can cooperate to form a 32-bit datapath, as the shadowy blocks show in Figure1. With the aid of the adapter, two 16-bit cells (ALU, shifter, or DREG) process high and low part of 32-bit data respectively to reconfigure a 32-bit cell, while the reconfiguration of a 32-bit MAC bases on such a fact:

$$A_{32} \times B_{32} = (A_{h16} \cdot A_{l16}) \times (B_{h16} \cdot B_{l16})$$

$$= 2^{32} \times (A_{h16} \times B_{h16}) + 2^{16} \times (A_{h16} \times B_{l16} + A_{l16} \times B_{h16}) + 2^0 \times (A_{l16} \times B_{l16})$$

From this formula, it seems that we can implement a 32-bit multiply operation mainly with four 16-bit multipliers, a 4-2 compressor, and an 80-bit adder. In fact, the latter two are combined with the accumulator and rounding logics as one sub-module named ACC80, which is implemented with a Wallace tree followed by a CLA adder. Figure3 shows the block diagram of the reconfigured 32-bit MAC.

Introducing a self-timed 2-stage pipelined multiplier in our recent research [9], only two 16-bit MAC units are necessary to reconfigure a 32-bit MAC which can also perform a multiply-accumulate operation within a single instruction cycle. With such an improvement, the number of 16-bit processing units needed to reconfigure a 32-bit datapath is reduced from four to two in the next generation of the DSP. Obviously, this will either lower

the area cost of reconfigurable datapaths by approximately 50 percents or result in higher parallelization grades (four) and thus higher performance in 32-bit data processing.

The instant-scalability of the datapath is achieved mainly by changing the values in mode registers, under the control of which the processing unit(s) without valid select signal will be disabled and disconnected from the datapath.

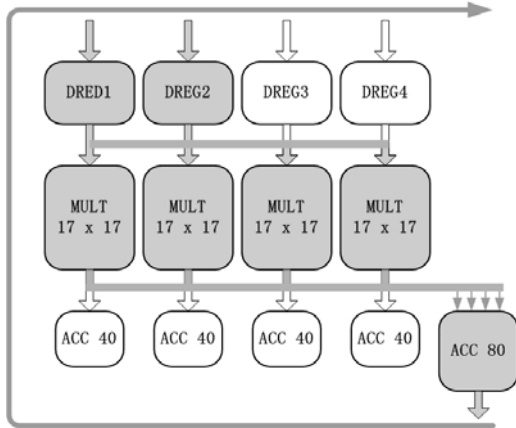


Figure 3.

Block diagram of reconfigured 32-bit MAC

2.3. Bus and Memory Subsystem

For SIMD processors, data supply and exchange are the key issues [10]. To supply large data throughput for the paralleled datapaths, buses with high bandwidth are necessarily introduced. For each 16-bit datapath, there are two dedicated data buses to access a couple of corresponding data memory (DM) blocks -- DMX and DMY -- as Figure 1 shows. The transmitting of data has no conflict with the fetching of instruction because of the separation of data buses and program bus. With such architecture, the DSP can smoothly perform many parallel operations in a single instruction, maximally including eight times of 16-bit computation and 16 times of DM access with two times of DM address modification. All data exchanges through buses between different DREGs, memory blocks, and control register files are centrally controlled by a bus controller.

The separated program memory (PM) and data memory blocks are accessed and governed by a memory controller, which also serves as an interface to the external data and address buses. Since the DSP is a synthesizable core for embedded applications, the memory subsystem will vary with implementations. Page mode access as well

as multiprocessor arbitration can also be supported.

For SIMD scheme, the eight couples of DM blocks are actually accessed by a same pair of addresses, which is computed by two data address generators (DAG1 and DAG2) respectively, i.e. every eight DM blocks use a same address. A variety of special addressing modes optimized for DSP algorithms are supported, such as bit-reversed addressing and circular addressing. Addressing with pre- or post-modification is also supported.

2.4. Control Unit

The global controller is similar to typical RISC-type control units. It is made up of two main parts: an instruction fetch unit and a program sequencer used for program counter calculation, program flow control, and housekeeping of all control registers. To support nesting of interrupt, loop, and subroutine, respectively, the PC stack has a depth of 33 levels, the loop stack has a depth of eight levels, and the status stack has a depth of 16 levels.

As described in [11], with the technique of type-separated instruction decoding, the low-power designed instruction decoder can shut off all arithmetic units which is unnecessary for current instruction, so that to minimize the power consumption of the core.

3. Implementation

A first prototype of the DSP for experimental purpose has been fabricated. The chip micrograph is shown in Figure 4. Control units are located in the center, while processing units (or DP) are symmetrically placed in the corners -- close to their corresponding data memory blocks (DM symbolizes both DMX and DMY in Fig. 4). Such a floor plan conduces to reducing clock skew and increasing operation frequency. All dedicated hardware and interface units except for the RAM and PLL are designed in Verilog HDL, synthesized, placed, and routed following a conventional design flow with Synopsys' EDA tools. The arithmetic modules in the datapaths were synthesized from structure-optimized Verilog HDL descriptions, since standard high-level synthesis did not lead to reasonable results and full custom design methodology was not available. All samples delivered from the fab show typical voltage and frequency characteristics and can be used in application systems without any functional restrictions.

Table 1 summarizes the DSP's major features. Using 0.18 μ m CMOS standard cell technology with five metal layers, the DSP reaches a performance of 1 billion MACs per second at 125 MHz with relatively low power consumption. Due to the regularity of the architecture, the design can be scaled to a higher parallelization grade and clock rate.

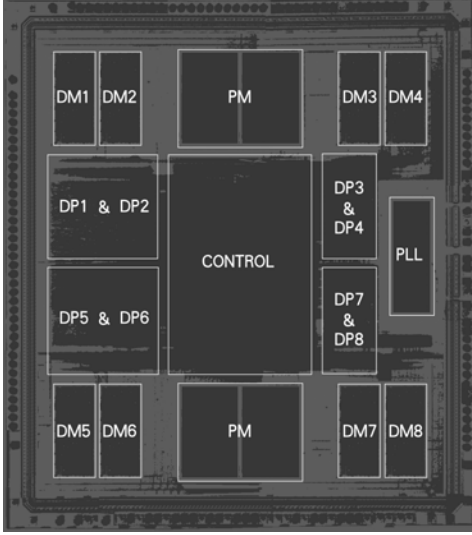


Figure 4. Chip micrograph

Table 1. Major features of the first prototype

Process	0.18 μ m, 5-Layer Al, CMOS standard cell
Parallel Datapaths	8DPs(16-bit), 2DPs(32-bit)
On-Chip Memories	8k words PM (8k \times 24-bit) 16k words DM (16 \times 1k \times 16-bit)
Clock Frequency	50MHz ~ 125MHz
Peak Performance (at 125MHz)	1 GMACS (16-bit) 250 MMAC (32-bit)
Supply Voltage	1.8V(internal), 3.3V(I/O)
Power Consumption	0.15 mW/MMAC (16-bit) 0.6 mW/MMAC (32-bit)
Chip Size	3.3mm \times 3.8mm
Logic Cell Area	2.48mm ²
Package	160-Pin MQFP

4. Test Results

The performance of the DSP has been tested by several kernel algorithms with assembler written implementations. Test stimuli vectors are not extracted directly from the interface but are pre-placed into the memories so that to measure the real performance of the

DSP as a core rather than a chip, since it is designed for embedded applications.

Table 2 shows an overview of the DSP's performance and comparisons with other DSPs. Comparisons with other DSPs for a 256-point complex FFT is also shown in Table.3. All these data are delivered from 8-way 16-bit processing. The results show a relatively high performance for a programmable DSP, which can competitive with that of state of the art DSPs; but a clearer improvement over them is its better scalability and flexibility besides the high performance.

The power consumption of the DSP is relatively low, a typical value of which, for the overall processor at 100MHz, is about 116mW for 8-way 16-bit FIR and 119mW for 2-way 32-bit FIR, respectively.

Table 2. Benchmarks and performance comparisons in cycles [12]

Benchmark	ADSP2153x Blackfin	TMS320C55x DSP	8-Datapath DSP
Clock Rate (MHz)	300	200	125
MMACS	600	400	1000
Block FIR Filter	(x/2)(2+h)	(x/2)(4+h)	(x/8)(2+h)
Complex FIR Filter	2h+2	2h+4	h/2+2
Biquad IIR (4 coeff)	2.5bq+3.5	3bq+2	0.625bq+5

h= # of taps, x= # of samples, bq= # of biquads

Table 3. Performance comparisons for 256-point complex FFT [13, 14]

	ADSP 218x	TMS320 C5416	8-DP DSP	ADSP 2153x	TMS320 C6203
Clock Rate (MHz)	80	160	125	300	300
Processing Time (μ s)	92.8	65	26.8	12	9

5. Conclusions

A new reconfigurable parallel architecture for embedded DSP core with high performance and flexibility is proposed and implemented. The SIMD-controlled datapath array enhanced with instant-scalability supplies the DSP with a capability of dimension-controllable vector processing. With the upward reconfigurable ability of the 16-bit processing units, the DSP can also process 32-bit data or vectors. Both of these features are also

applicable beyond this design, contributing better flexibility as well as higher performance. The operation performance of the prototype DSP chip fabricated by 0.18 μ m CMOS process reaches 1GMACS at the clock of 125MHz with the peak power less than 0.15mW/MMAC in 16-bit, and 0.6 mW/MMAC in 32-bit. These features show that the DSP core can be used in many applications.

References

- [1] T. J. Dillon, Jr., "The VelociTI architecture of the TMS320C6x," in *Proc. Int. Conf. Signal Processing and Technology*, vol. 1, San Diego, CA, Sept. 1997, pp. 838–842.
- [2] Willm Hinrichs, Jens Peter Wittenburg, Hanno Lieske, et al., "A 1.3-GOPS Parallel DSP for High-Performance Image-Processing Applications," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, Jul. 2000, pp. 946-952.
- [3] John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed., Morgan Kaufmann, San Mateo, CA, 1996.
- [4] Scott Beach, "The Advantages of a Scalable DSP Architecture," www.starcore-dsp.com/technology/articles.html.
- [5] Jie Chen, Chaoxian Zhou, et al., "A reconfigurable architecture of high performance embedded DSP core with vector processing ability," in *Proc. Int. Conf. ASIC*, Beijing, China, Oct. 2003, pp. 377-380.
- [6] H. Igura *et al.*, "An 800 MOPS 110 mW 1.5 V parallel DSP for mobile multimedia processing," *ISSCC Dig. Tech. Papers*, 1998, pp. 18.3.1–18.3.10.
- [7] A. D. Booth, "A signed binary multiplication technique," *Journal of Mechanics and Applied Mathematics*, vol.4, 1951, pp. 236-240.
- [8] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Elect. Comput.*, vol. EC-13, Feb. 1964, pp. 14–17.
- [9] Ying Li and Jie Chen, "A reconfigurable architecture of a high performance 32-bit MAC Unit For embedded DSP," in *Proc. Int. Conf. ASIC*, Beijing, China, Oct. 2003, pp. 1285-1288.
- [10] Kai Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, New York, 1993.
- [11] Han Liang, Chen Jie, and Chen Xiao-dong, "Design of Instruction Decoder of Reconfigurable Embedded DSP Processor," *Microelectronics and Computer*, vol. 21, no. 8, Aug. 2004.
- [12] Analog Devices, Inc., "DSP Benchmark Comparison," <http://www.analog.com/processors/processors/blackfin/benchmarks/comparison.html>.
- [13] Analog Devices, Inc., "DSP Selection Guide," ed. 2002, <http://www.analog.com>.
- [14] Berkeley Design Technology, Inc., "Evaluating DSP Processor Performance," http://www.bdti.com/articles/benchmk_2000.pdf