

P-11-10L

Embedded Signal Processor Design for Biosensor System-on-a-chip applications

Yeong-Kang Lai, Yu-Chieh Chung, Yu-Fan Lai, and Chen-Lung Hwang

Department of Electrical Engineering, National Chung Hsing University
No.250, Kuo Kuang Road, Taichung, Taiwan
Phone: 886-4-22855268 E-mail: yklai@dragon.nchu.edu.tw

1. Introduction

A new emerging field in electronics is the development of biosensor system-on-a-chip (SoC) applications [1]-[2]. These kinds of SoCs can be implemented as programmable and automated processors for biological and pharmaceutical analysis, as shown in Fig 1. As the advance of semiconductor technology, the SoC can integrate different functional units. However, much care has to be put in the design of embedded signal processor which has to provide real-time processing capabilities in order to deal with complex systems composed of sensors, actuators, signal conditioning and processing circuits. Moreover, programmability and reconfigurability are key design features to get a flexible and reusable architecture. There are already many existing programmable DSPs [3]-[4]. But they do not focus on design-reuse. System-on-chip designers can not use them in their design. Our proposed embedded signal processor can help the designers to translate their signal processing algorithms to actual systems quickly such as a noise reduction filter. In addition, our design is aimed at a fully synthesizable signal processor, which can be easily embedded into a biosensor system.

2. Proposed VLSI Architecture

In this paper, this proposed DSP uses an advanced modified Harvard architecture that maximizes processing power with seven buses. Separate program memory, data memory, and coefficient memory allow simultaneous access to program instructions and data, providing a high degree of parallelism. For example, two data reads and one instruction read can be performed in a single cycle. Instructions with parallel memory access and arithmetic execution utilize this architecture. For flexibility, data can be transferred between data and coefficient memories. There are specialized computational units to support powerful arithmetic, logic, and bit-manipulation operations. Also, the DSP includes the control mechanisms to manage interrupts, repeated operations, and function calling. Fig.2 shows the architecture of this DSP.

In order to maximize the throughput, the DSP employs a five-stage instruction pipeline (excluding MAC-type instructions) divided as follows: 1) program prefetch, 2) program fetch, 3) instruction decode, 4) operand address generation, 5) execute/write back. Because the delay of the multiplier-accumulator is very long and in order to balance the delay of each pipeline stage, the MAC-type instructions

uses the fifth stage to perform multiplication and another stage (sixth stage) to perform accumulation. With this arrangement, the throughput of the DSP is not restricted to the MAC. Excluding the two-stage pipelined 17x17-bit MAC, there are two arithmetic units. One is ALU, and the other is shifter. They are arranged to operate in parallel. And each unit has input registers to latch the operands from the data bus (DB), and result registers to latch the computation result. These result registers drive the internal bus (R-Bus), and R-Bus is drawn back to each computational unit without passing the input registers. So the result generated by any arithmetic unit can be the operand of any arithmetic unit at the next cycle. The input and output of the input registers and result registers are all connected to DB-Bus (Data Bus), so the data in registers or in memory can be loaded into any register through DB. Because the data can be transferred among the registers of the three arithmetic units, and any output register can offer the operand to any arithmetic unit, these registers can be regarded as a register file. This feature makes programmers get more flexibility.

The ALU performs a standard set of arithmetic and logic operations in addition to division primitives. The MAC performs multiply, multiply/add, and multiply/subtract operations. The shifter performs logical and arithmetic shift, normalization, denormalization, and derive-exponent operations. The two-stage pipelined MAC is adopted in this DSP in order to release the long delay of it. However it results in a type of data hazard (called MAC-type hazard), while an instruction I1 uses the result of the MAC instruction, which is just one instruction before I1, as its source. When the hazard happens, I1 and the instructions following it must be delayed on cycle to protect the pipeline correctness. There is the other type of data hazard happened, while an instruction loading data into the register in a DAG is followed by the instruction using the same DAG to access memory. The resolving method is the same as the first type of data hazard. So the hardware overhead of using two-stage pipelined MAC is only to add the MAC-type hazard detector. And the two-stage pipelined MAC makes the DSP operate at higher frequency.

There are two data address generators (DAGs) in this DSP. They are used to indicate the memory address that the programmers may access separately. Because they can act independently, the DSP can access two operands in one clock cycle. They both contain a special circuit to support circular addressing, which is useful when the DSP is performing filtering operations. And a bit-reverse circuit is

added to DAG1 to support the bit-reversal addressing which is useful when the DSP performs FFT operations.

There are two control units controlling the whole DSP. The Program Sequencer controls the flow of the program, and the Central Controller controls the operations of each instruction. Program Sequencer decides which instruction to be fetched and generates its address. It also contains an interrupt controller. It manages not only the call, jump, return, and interrupt but also hardware zero-overhead looping. Because typical signal processing programs contain a lot of loops, there is a special circuit to control the program flow when the DSP enters a loop. Also, there are some registers to store the mode and status of the DSP in this unit. In this DSP, we use two level controllers to control each circuit. The central controller is the first level, and the sub-controller in each block is the second level. The function of the central controller is to decode the instruction into sub-codes and dispatch the sub-codes to the sub-controller of each block. Due to the two level controllers, the control signals between blocks can be reduced very much, and the two level controllers can be arranged to act on different pipeline stage to release the decoding delay. In this DSP, they occupy two consecutive pipeline stages. The central controller also takes response for the pipeline flushing at the discontinuity of PC (Program Counter).

3. Performance Analysis

The function of every instruction of the DSP is completely verified from the front end to back to the back end. We also simulate some kernel operations of the most digital signal processing applications. They are a 56-tap lowpass FIR filter, a 4-tap lowpass IIR filter, a 21-tap LMS adaptive filter, and 128-points FFT. And while executing the three kernel operations, the performance of the DSP is listed in Table 1 with comparison to other DSPs, where N is the order of the filter, A,B are constant, and X indicates that the author did not mention the operation. The DSP processor is implemented using TSMC CMOS technology. It can operate at 100 MHz. The chip layout and its features are shown in Fig. 3 and Table 2, respectively.

4. Conclusion

This paper proposed a fixed-point, low-cost and fully synthesizable programmable DSP. It can provide real-time processing capabilities in order to deal with complex systems composed of sensors, actuators, signal conditioning and processing circuits for biosensor system-on-chip applications. From implementation results, it is suitable to be integrated on a system-on-a-chip.

References

- [1] N. Manaresi, G. Medoro, M. Tartagni, L. Altomare, and R. Guerrieri, "Microelectronics Meets Biology: Challenges and Opportunities for Functional Integration in Lab-on-a-chip". *European Solid State Devices Research Conference - European Solid State Circuits Conference (ESSDERCESSCIRC)*,

pages 31–36, September 2002.

- [2] G. Medoro, A. Leonardi, L. Altomare, N. Manaresi, M. Tartagni, and R. Guerrieri, "A Lab-on-a-chip for Cell Detection and Manipulation". *IEEE Sensors*, June 2002.
- [3] Wai Lee *et al.*, "A 1-V Programmable DSP for Wireless Communications," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1766-1774, Nov. 1997.
- [4] F. Campi, R. Canegallo, and R. Guerrieri, "IP-Reusable 32-Bit VLIW Risc Core". *European Solid State Circuits Conference (ESSCIRC)*, pages 456–459, September 2001.

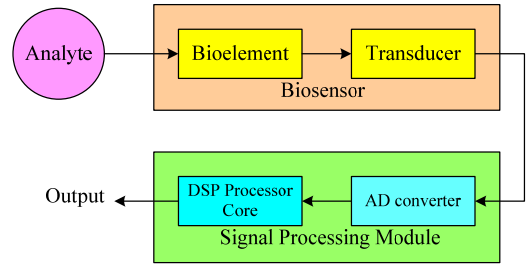


Fig.1 Block diagram of biosensor system.

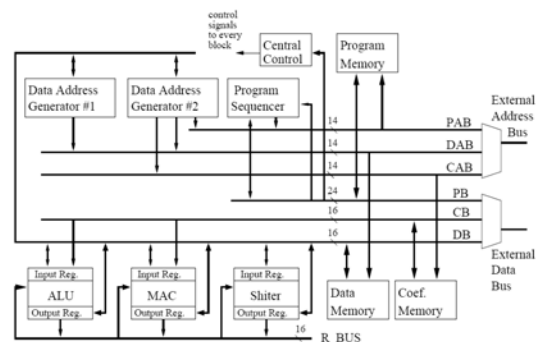


Fig.2 Architecture of DSP processor core.

Table 1. Performance comparison.

Operations/Cycles	TI's C54x	MDSP-II(two MACs)	Proposed
FIR filter	$N + 4$	$N/2 + A$	$N + 11$
IIR filter	$2N + 11$	$N + B$	$2N + 16$
adaptive filter	$2N + 14$	X	$4N + 22$

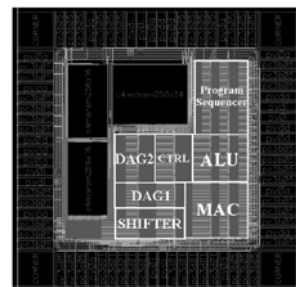


Fig. 3. Chip layout.

Table 2. Chip features

Technology	TSMC CMOS 1P6M
Die size	3100x3100 (μm^2)
Transistor count	30,000
Working frequency	100MHz
Memory	1 256x24-bit and 2 256x16-bit
Voltage	1.8V