3-Way 32 bit VLIW Multimedia Signal Processor

***Jaebok Choi, **Jaehee You**
***LG Electronics Digital TV Research Laboratory**
****School of Electrical Engineering, Hongik University**
E-mail : kumsin@lge.com

Abstract

A 3-way VLIW multimedia signal processor capable of efficient repeated operations as well as both load/store and type transformations for various data types is presented. It is composed of a 32-bit execution unit that can execute two instructions in parallel, an independent load/store unit and a control unit. The processor is implemented with 0.6㎛ gate array and the results are discussed.

1. Introduction

Today, with the expansion of signal processing application areas such as multimedia and mobile communication, DSP processors become more important than ever, particularly to meet the demand for real-time processing of massive data. Especially for multimedia data processing, parallel processing and rapid data fetch from external memories with limited bandwidth are required. VIS instructions of Ultra-SPARC, MMX processor [1, 2] and Mediaprocessor [3] have been presented in microprocessor and signal processor areas to meet these needs. However, these processors are designed for general purpose uses and tremendous amount of hardware is required. For an application areas with growing market, ASIC implementation is preferred for chip size, speed and power, but long design time and rapid standard change make it difficult. To mitigate the above problems, a 3-way VLIW signal processor and instructions capable of efficient repeated operations with various data types as well as reconfigurable inter-connections between units according to application areas are proposed. In section 2, proposed instructions effective in multimedia processing will be discussed. In section 3, the architecture will be presented. In section 4, the verification results based upon designed assembler and performance evaluations will be discussed. In section 5, the VLSI implementation results and test board design will be described.

2. Instructions

For rapid computation of digital signal processing algorithms, it is essential to execute effectively and quickly identical computations composed of addition, subtraction, accumulation, bit manipulation, and shift with massive amount of low precision data such as 8 and 16-bit. For these purposes, the instructions capable of continuous calculations with low precision data packed in 32-bit data type such as blockwise multiplication, addition, subtraction, shift are proposed. For example, to compute filtering

algorithm with eight 8-bit data and coefficients, 8 multiplications and additions are needed in general purpose processor. In contrary, only one instruction is enough with proposed instructions. For effective block data processing operation, it is crucial to have I/O instructions to fetch the data scattered all over external data memories into processor as well as to store the results in the memory in high speed. For example, to load and compute eight 8 bit data from memory, eight load, shift, bitwise OR instructions are required in conventional processors. To alleviate this problem, continuous load/store instructions along with pack/unpack instructions for data type transformations are proposed. Also, min, max, average arithmetic instructions and blockwise clip instructions are added.

The structures of the overall instructions are shown in Table 1. If more parameters than allowed in 32 bit instruction length, they are stored in special purpose registers. Using the instructions for high speed signal processing in addition to conventional instructions, the most of digital signal processing algorithms can be implemented in software effectively.

Table 1. Instruction Format

op	rd	op3	rs1	I	unused(zero)	rs2
31 29	24	18	13	12	4	0

op	rd	110110	rs1	opf	rs2
31 29	24	18	13	4	0

3. Architecture

As shown in Fig. 1, the proposed 32-bit processor consists of controller, load/store unit with memory address generation unit (AGU), program counter unit, register file(RF) and execution unit with shift /pack/unpack block, two ALUs and a multiplier. Operands are fetched from RF and the computations for various data

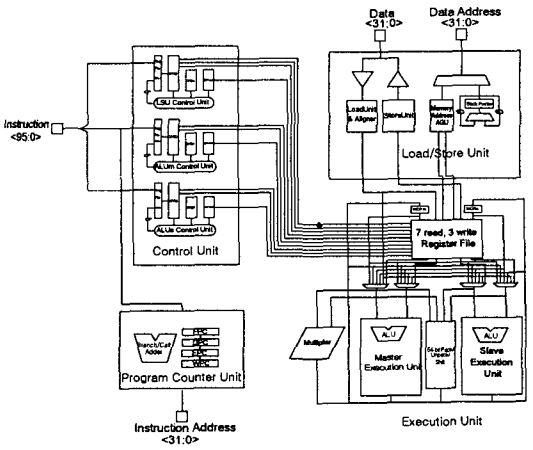

Fig. 1. Overall architecture

types including 8 or 16 bit are executed in the execution unit. The execution results are written back into RF. For parallel processing with multiple processing units, 7 read and 3 write multi-port RF is used. To facilitate continuous RF access for repeated operations, AGU is designed to generate RF address effectively.

In Fig. 2, reconfiguration methodologies by rearranging the I/O of ALU and multiplier are shown. In blocks Fig. 2, the reconfiguration structures of accumulation after absolute value computation, multiplication-accumulation and two independent ALU operations are shown in Fig.

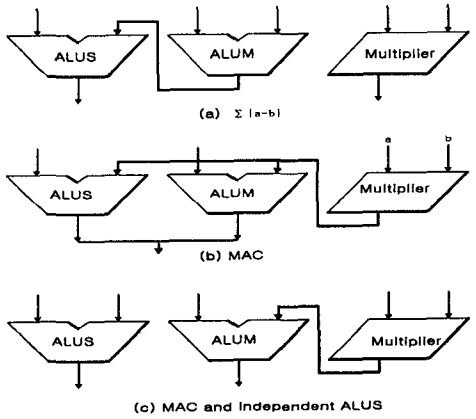

Fig. 2. Reconfiguration of hardware

2(a), 2(b) and 2(c) respectively. In case of the accumulation, RF utilization is enhanced and RF read/write execution time is eliminated by using bypass or feedback at the output of RF instead of storing at RF for high speed operation. For the computation of various data types 8, 16, 32 bit as well as 64 bits for multiplication accumulation results or loop unrolling, two 32 bit execution units are partitioned into 8 bit units for the parallel computations with min resolution of 8 bit data. They are joined by connecting lower bit carry output to higher bit carry input or connecting lower and higher bit shift units to make data bit width scalable. Therefore, the proposed execution unit can easily accommodate the computations for both a few high and multiple low precision data ranging from 64 to 8 bit with high hardware utilization.

4. Assembler and performance evaluation

For the proposed processor, a new assembler is designed. Benchmark tests are done based on the assembler results. The performances are evaluated in instruction counts and clock cycles by comparing the execution results of the assembly codes of SPARC V8 processor and the proposed processor assembler results in case of hierarchical search motion estimation algorithms. About 70% and 90% are reduced for instruction counts and clock cycles respectively.

In Fig. 3 the simulation results for continuous loads are shown. The signals in Fig. 3 are clock, instruction address, data address, data bus, the write address of RF and the data written into the memory.

5. VLSI Processor Implementation and Test board design

5.1. VLSI Processor Implementation

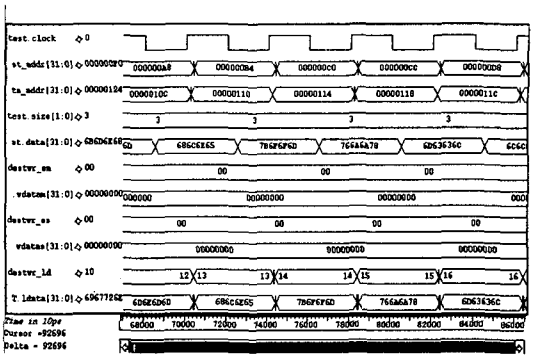

Fig. 3. Simulation results

Verilog code is used to define the functions of each block. The processor chip is implemented using 0.6㎛ Samsung gate array technology. Fig. 4 shows the placement and routing results using gate ensemble.

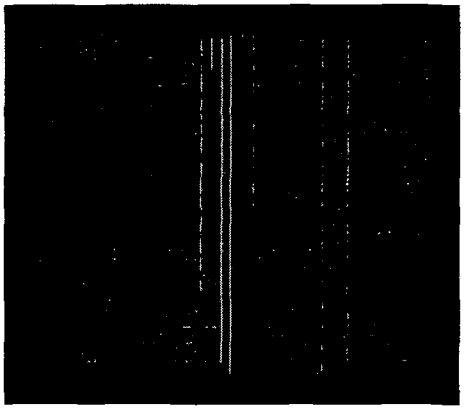

Fig. 4 Placement and routing results

5. 2. Test board design

Fig 5. Shows the schematic of the test board for the designed processor. The test board is designed using the processor along with instruction and data memories. The assembler results are relocated such as separate two instructions for execution unit and one instruction

for load /store unit.

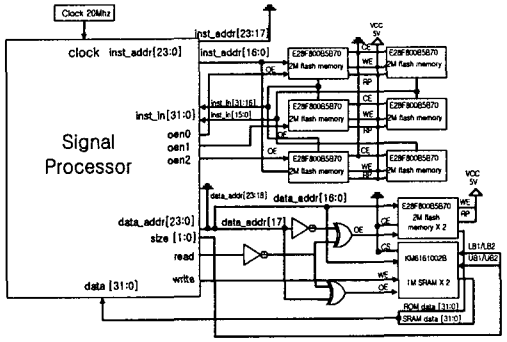

Fig. 5. Test board schematic

The instructions are stored at two 16 bit flash memories and the data are stored at external SRAM and two 16 bit flash memories. MSBs of address are used to distinguish the flash memories and SRAM.

6. Conclusion

A 32 bit 3-way VLIW multimedia signal processor is proposed. The proposed processor is capable of various types of data I/O and data type transformations with block data load/store and arithmetic. Also, new multimedia instructions are proposed. The hardware blocks are designed to be reconfigurable for each application areas. The processor is designed with verilog code and synthesized with synopsys and implemented using Samsung 0.6㎛ gate array. The processor consists of about 112,000 gates and the maximum clock frequency is 25MHz . The assembler for the processor is designed and the performance is evaluated in case of motion estimation algorithm.

References

[1] L. Kohn et.al., The Visual Instruction Set(VIS) in UltraSPARCTM , in Proc. COMPCON, pp.462-469, Spring 1995.

[2] Mustafiz R. choudhury, et al., A 300MHz CMOS Microprocessor with Multi-Media Technology , ISSCC, Feb. 1997.

[3] Craig Hansen, Architecture of a Broadband MediaProcessor, COMPCON 96, 1996.