

High-Performance Reconfigurable SIMD Machine *

Xizhen Xu, Ph.D candidate
Dept. of Electrical and Computer Engineering
New Jersey Institute of Technology
Advisor: Sotirios Ziavras
Email: xx3@njit.edu
Phone#: 973-412-9220
Mailing address: 425, Mt. Prospect Ave., Apt.210
Newark, NJ, 07104

***The work presented here was financially supported in part by US Dept. of Energy and academically advised by Prof. Sotirios G. Ziavras affiliated to Dept. of Electrical and Computer Engineering at New Jersey Institute of Technology.**

Key words: SIMD (Single Instruction and Multiple Data), PE (Processing Element), ILP (Instruction Level Parallelism), pipeline, reconfigurable computing, Virtex-II™, Wildstar™

In recent years, the move towards a new type of processor architecture known as adaptive, or reconfigurable, computing has dramatically changed the way of hardware design and test. It has greatly reduced the complexity of SOC design and prototyping. As the world leader of programmable logics, Xilinx® has released Virtex-II™ platform FPGAs to meet the ever-increasing needs of reconfigurable computing which enables the hardware designers to achieve high computing capability as well as flexibility with low cost.

My design objective is to build up a high performance SIMD processor (Single Instruction Multiple Data) to solve the large-scale matrix problem of linear systems. The design architecture is described as follows:

- 1) Xilinx® MicroBlaze™ 32-bit integer microprocessor is employed as an integer unit and system controller whose functions are to execute all incoming integer and control instructions and send all the floating point instructions to SIMD;
- 2) SIMD consists of fourteen PEs (Processing Elements) implemented on two Xilinx® Virtex-II™ platform FPGAs;
- 3) each PE is 11-stage pipelined and has its own 1KB distributed SRAM;
- 4) SIMD ISA (Instruction Set Architecture) includes floating point addition, subtraction, multiplication and division, load and store, and PE routing instructions.

In order to exploit ILP (Instruction Level Parallelism), the pipeline technique is widely used in this design. The PE's floating point unit consists of pipelined addition, subtraction, multiplication and division. SIMD has been pipelined into 11 stages for addition, subtraction, multiplication, load/store, and routing instructions; 36 stages for division, respectively. A finite state machine is designed to make sure there are 25 extra clock cycles to compensate division instructions. All instructions pass through the following pipe stages: IF1 (Instruction Fetch1), IF2 (Instruction Fetch2), ID (Instruction Decoding), EX1 (Execution1), EX11, EX2, EX3, EX31, MEM (Access Mem), WB1

(Write Back1), WB2. Additionally, a hazard detection unit and a bypass unit are designed to reduce data hazard (Most of the time, it is RAW hazard). If a hazard is detected, the controller will stall the SIMD pipeline and blow “bubbles” into it. Bypass unit will forward the execution results as soon as possible to reduce pipeline stalls and eliminate data hazards. If there is no data hazards in software, the pipelined SIMD can produce execution results each clock cycle, i.e., CPI = 1.

The implementation of SIMD design is targeted on Annapolis Wildstar™ II PCI board. There are two Xilinx® Virtex-II™ platform FPGAs with 12 million system gates populated in Wildstar™ II PCI board. After simulation, synthesis, verification and layout, the SIMD can successfully run on Annapolis Wildstar™ II PCI board at the speed of 34MHz. The critical path in the SIMD is floating point unit; The SIMD machine can run at the speed of 128MHz if an integer unit fits into it. The high-performance of the SIMD architecture was proved.

Several test programs have been successfully run on Annapolis Wildstar™ II PCI board. All these designs can be implemented without hardware PCB level modifications. Therefore, the significant reduction for hardware design cost and time to market has been achieved through reconfigurable computing.