T0: A Single-Chip Vector Microprocessor with Reconfigurable Pipelines

Krste Asanović, James Beck, Bertrand Irissou,
Brian E. D. Kingsbury, & John Wawrzynek
Department of Electrical Engineering
and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720-1776
and the
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1105

Abstract

A single-chip fixed-point vector microprocessor is described. The chip contains a MIPS-II RISC core with a 1 KB instruction cache, dual eight-way parallel vector arithmetic pipelines, a 128-bit memory interface, and an 8-bit serial host interface. Each vector arithmetic pipeline contains a cascade of six functional units that can be dynamically reconfigured by each instruction. The resulting peak performance is 4.3 billion 32-bit arithmetic operations per second at a clock speed of 45 MHz.

1 Introduction

Vector microprocessors have the potential to provide high-performance, low-cost computing support for a wide range of multi-media and human-machine interface applications. Vector instruction set architectures (ISAs) simplify the control logic needed to produce multiple results per cycle, and can be readily scaled to exploit advances in fabrication technology while remaining object-code compatible with earlier designs [1].

T0 is the first implementation of the Torrent ISA. Torrent is a vector-register architecture, with all vector instructions encoded in the coprocessor 2 space of the MIPS-II ISA [2]. The chip (Figure 1) contains 730,701 transistors on a $16.75 \times 16.75\,\mathrm{mm}^2$ die, fabricated in Hewlett-Packard's CMOS26G process using $1.0\,\mu$m MOSIS scalable CMOS design rules with two layers of metal. Operating at a peak clock rate of 45 MHz with a 5 V supply, T0 dissipates less than 12 W. It has 428 pads, of which 208 are signal pads. First silicon was received April 4, 1995, and was fully functional. The entire project required under 10 person-years to go from having no ISA to running applications on a complete system.

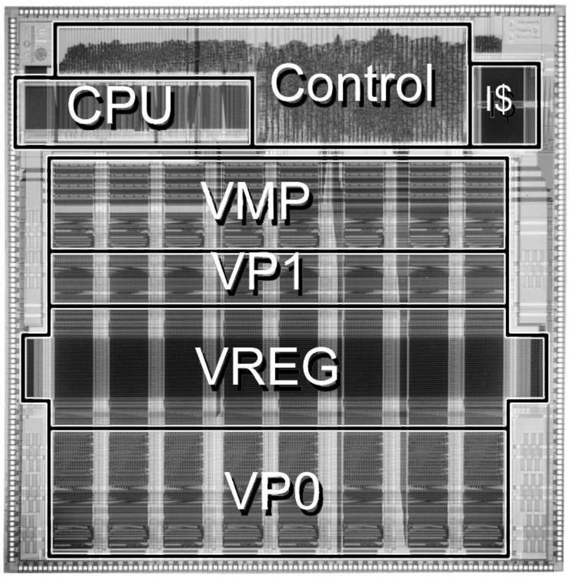

Figure 1: T0 Die Photograph.

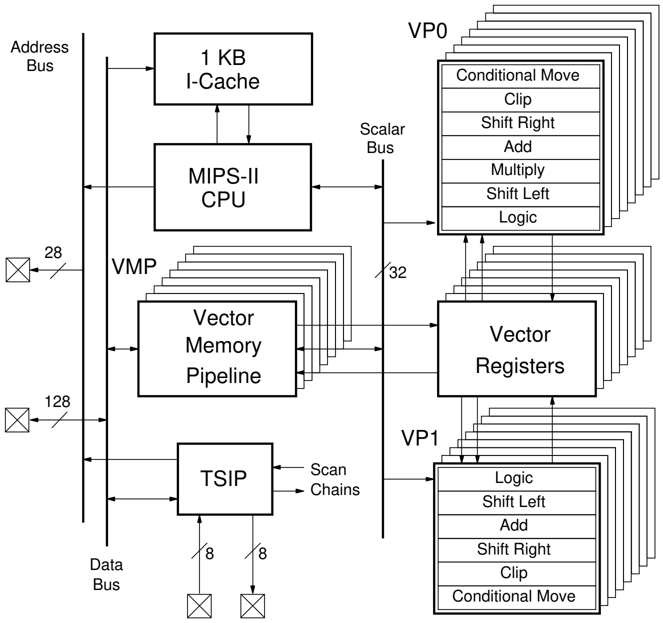

Figure 2: T0 Block Diagram.

2 Processor Overview

The structure of T0 is shown in Figure 2. The MIPS-II RISC core executes one instruction per cycle in a six stage pipeline. The load latency is three cycles, and load delay slots are fully interlocked. Instruction cache miss penalty is two cycles with prefetch and three cycles without. Prefetches take place whenever the memory interface is otherwise idle.

The external memory interface supports up to 4 GB of asynchronous SRAM with 720 MB/s bandwidth. SRAM access is wave-pipelined over 1.5 cycles using industry standard 17 ns parts at 45 MHz. The serial host interface supports up to 30 MB/s of DMA I/O bandwidth, and also gives access to on-chip scan chains for die testing and system debugging.

The vector coprocessor contains a vector memory unit (VMP), two vector arithmetic units (VP0 and VP1), and an 8-port vector register file with 16 vector registers holding 32 elements of 32 bits each. The vector memory unit handles vector load/stores, scalar load/stores and vector editing instructions. Memory operations transfer 8-bit, 16-bit and 32-bit elements. Vector addressing in external memory can be unit stride with address post-increment, non-unit stride or indexed. In indexed addressing, a vector register supplies address offsets to the elements of the operand vector, supporting vector scatter/gather operations. Unit stride load/stores move eight 8-bit or 16-bit operands or four 32-bit operands per cycle, with a one cycle delay if the first element is not aligned to a 16-byte boundary. Indexed and strided load/stores can move only one element per cycle due to the single external memory address port. Vector editing instructions include scalar insert, scalar extract and a vector extract to support vector reduction operations.

The vector arithmetic units are composed of eight parallel pipelines, and can produce up to eight results per cycle with a latency of three cycles. All vector pipeline hazards are fully interlocked, so instruction scheduling is required for performance only, not to ensure correct operation. The vector registers are striped across the eight pipelines; with the maximum vector length of 32, each vector functional unit can accept a new instruction every four cycles. T0 can saturate all three vector units (VMP, VP0 and VP1) by issuing one instruction to each on successive cycles, leaving an issue slot open for the RISC core every four cycles.

Dynamically Reconfigurable Vector Arithmetic Pipelines

The arithmetic pipelines with the VP0 and VP1 vector arithmetic units contain a cascade of six 32-bit functional units including logic, left shift, add/subtract, condition compare, right shift, and clip. The pipelines in VP0 also contain a 16-bit $\times$ 16-bit multiplier that produces 32-bit results. The multiplier is an overturned-stairs, Baugh-Wooley array multiplier [4] constructed from transmission-gate full adder cells, and has a worst-case latency of 10 ns. Figure 3 shows the structure of the VP0 pipeline.

This cascade of functional units can be dynamically reconfigured on an instruction by instruction basis under control of configuration values held in a scalar register specified in the instruction. The pipeline can be arranged to perform a complete scaled, rounded, and clipped fixed-point operation in a single pass. Circuitry in the left and right shifters provides four rounding modes including round-to-nearest-even. The clipper unit can saturate results to lengths of 8, 16, or 32 bits, and can also convert the output of the condition comparator into boolean values. The condition comparator circuit is used to control conditional writeback of results.

The pipelines can also be reconfigured to provide other composite operations such as absolute value, max/min, and bit field extract. We have implemented a complete IEEE single-precision floating-point vector library using the reconfigurable arithmetic pipes, and this executes at around 14.5 MFLOPS with no other hardware support for floating-point arithmetic.

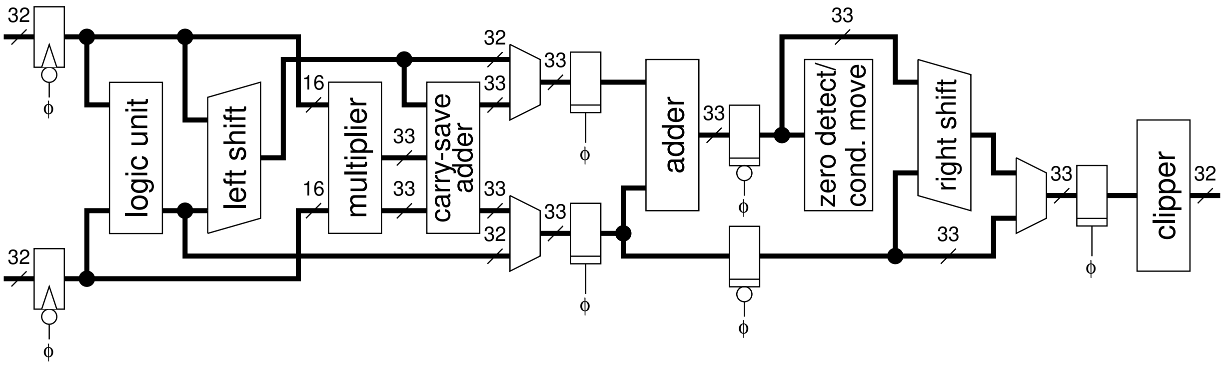

Figure 3: Vector Arithmetic Pipeline (VP0) Datapath.

Vector Register File

The vector register file serves as the interconnect between the vector functional units. Each arithmetic unit requires two reads and one write per cycle, while the memory unit requires one read and one write per cycle, for a total of five read and three write ports. As with all functional units, the vector register file is organized as eight parallel slices of 32 bits each; so all ports into the register file are 256 bits wide. To reduce latency between chained vector operations, values written in a cycle can be read on the same cycle. For each 32-bit slice of each write port there is a separate write enable to support conditional moves and vector lengths that are not divisible by 8.

To minimize the overall size of the register file, the address decoders, select lines and bit lines are time-multiplexed between read and write operations, with three writes occurring on the first phase of the clock, and five reads occurring on the second phase. Reads are single-ended, and use a chain of ratioed inverters for sensing, while writes are differential. A total of six bit lines, to support the three differential writes, and five address decoders, to support five independent reads, are used in this design.

Although time-multiplexing the address decoders, select lines and bit lines is area-efficient, it creates two design challenges. First, to avoid overwriting the data stored in the registers read on the previous phase, write data must not be driven onto the bit lines before the register select lines have stabilized. Second, both the bit line precharge and bit cell read must occur in a single phase of the clock. We employed a self-timing mechanism to solve both problems. The register file contains 213K transistors in $2.7 \times 15.7\,\mathrm{mm}^2$ of die area (Figure 1), and provides 11.5 GB/s of operand bandwidth at 45 MHz.

Circuit Style

T0 is implemented using full custom layout for the datapaths and standard cells for the control logic. The full custom logic uses a mixture of circuit styles. To reduce power dissipation and design complexity, static CMOS or pass-transistor logic is used wherever possible. All latches and flip-flops are TSPC dynamic elements [3]. Dynamic logic is also used in speed-critical circuits such as the register files, adder carry chains and zero-detect logic. The chip clock load of 1.4 nF is driven from a single clock driver with a total transistor width of 12 cm. The worst-case clock skew is estimated to be 1 ns. To control dI/dt and IR noise on the power rails, we include 25 nF of on-chip decoupling capacitance.

3 Summary

We have described the first complete single-chip implementation of a vector ISA. Compared to conventional scalar ISAs, vector ISAs allow simpler implementations as the number of parallel operations completed per cycle increases. This simplicity resulted in a processor with extraordinarily high performance per unit area·time and allowed us to implement the complete chip with a relatively modest design effort. Performance is further enhanced through the use of unique dynamically reconfigurable arithmetic pipelines. While the architecture we present here is fixed-point, it could be easily be extended to floating-point or could be scaled to higher performance by increasing the number of parallel slices in the functional units.

Acknowledgements

Thanks to Nelson Morgan, David Johnson, and John Lazzaro for their contributions to the T0 design effort, and to John Hauser for his work on the IEEE floating-point emulation libraries. Thanks to Andrew Beck for preparing the annotated die photo. Primary support for this work was from the ONR, URI Grant N00014-92-J-1617 and ARPA contract number N0001493-C0249. Additional support was provided by the NSF and ICSI.

References

[1] J. Wawrzynek, K. Asanović, B. Kingsbury, J. Beck, D. Johnson, and N. Morgan. SPERT-II: A Vector Microprocessor System. *IEEE Computer.*, **29**(3):79–86, March 1996.

[2] G. Kane and J. Heinrich. *MIPS RISC Architecture.* Prentice Hall, 1992.

[3] M. Afghahi and C. Svensson. A unified single-phase clocking scheme for VLSI systems. *IEEE Journal of Solid-State Circuits.*, **25**(1):225–233, Feb. 1990.

[4] Z. Mou and F. Jutand. "Overturned-stairs" adder trees and multiplier design. *IEEE Trans. on Computers.*, **41**(8):940–948, Aug. 1992.