Speed and Energy Performance of an Asynchronous MIPS R3000 Microprocessor

Alain J. Martin Mika Nyström Paul Penzes

Catherine Wong

California Institute of Technology

Pasadena, CA 91125

June 22, 2001

Abstract

This paper presents the speed and energy figures for an asynchronous implementation of a MIPS R3000 microprocessor. The design is almost entirely QDI and introduces a new fine-grained pipeline.

The performance figures show that this design is four times as efficient as equivalent clocked designs and that its cycle time in FO4 units compares to that of high-perofrmance dynamic pipelines.

1 Introduction

Asynchronous techniques for digital VLSI, in particular microprocessors, are enjoying a regain of interest as the power consumption of traditional clocked designs keeps growing and clock distribution is demanding an increasing fraction of the cycle time. But how does the speed and energy performance of those experimental designs compare to that of equivalent clocked designs? The purpose of this paper is to provide a solid "data point" by evaluating and discussing the performance (cycle time and energy consumption) of an asynchronous implementation of a MIPS R3000 microprocessor, the "Caltech MiniMIPS."

The MiniMIPS was designed at Caltech betweeen 1995 and 1998. One goal of the project was to provide strong evidence of the performance advantages of asynchronous design by comparing the performance of an asynchronous implementation of a standard commercial microprocessor to existing synchronous implementations. While the low-power advantage of asynchronous circuits was more or less acknowledged at the start of the project, the conventional wisdom at that time was that asynchronous circuits are significantly slower than clocked circuits especially for the implementation of a complete RISC microprocessor. We therefore focused the experiment on achieving high execution throughput and assumed that the low-power advantage of asynchronous design "would take care of itself".

Another goal of the project was to demonstrate the effectiveness of the Caltech synthesis method on a large design. Starting from a high-level description of the MIPS, the design was obtained by a succession of correctness-preserving transformations. The complete two-million-transistor design was the only chip we fabricated in this technology. It was found entirely functional on first silicon.

This paper reports the performance of the fabricated prototype and compares its performance as accurately as possible to that of commercial microprocessors—MIPS and others. This is not an easy task. Published performances are usually reduced to a single number for frequency and a single number for power consumption, with minimal information about technology and operating conditions. In order to compare our design to others, we had to reduce the performance of the MiniMIPS prototype to the same pair of numbers. We have strived to minimize the arbitrariness of this process by careful measurements and meticulous reporting of the procedure used to obtain the numbers.

We believe the results to be reliable. Based on the results, we claim that the MiniMIPS prototype is at least four times as fast as a commercial MIPS R3000 in the same technology at its initial introduction. As we already mentioned, the MiniMIPS was not designed for energy efficiency. Nevertheless, the fabricated prototype is as energy-efficient as the low-power R4600, and simulation indicates that a redesign that would simply eliminate some layout errors could be twice as energy efficient.

2 Asynchronous and QDI Implementation

Asynchronous circuits do not use clocks. The sequencing between different parts of a system, for instance two stages of a pipeline, is implemented by a handshake protocol, usually a four-phase (return-to-zero) protocol. The time required to complete the four steps of a handshake protocol (raising the data wires, raising the acknowledge wire, lowering the data wires, lowering the acknowledge wire) sets an upper bound on the throughput of the system.

There are different types of asynchronous circuits depending on the type of delay assumptions that are used. We are interested in the asynchronous circuits that make the fewest delay assumptions because circuits of this type are the most robust to variations in the physical parameters induced by both logical and electrical incertainties. These circuits are called "quasi delay-insensitive" (QDI).

If no timing assumption is used, as in QDI circuits, detecting that all data wires have been raised or lowered requires a piece of circuitry, called a completion tree, the delay of which is proportional to the logarithm of the number of bits in the data path. For a 32-bit data path, the completion tree mechanism adds approximately 10 elementary gate delays to the critical cycle period. This added delay is in most cases unacceptable. Up till this project, most designers believed that the only way to reduce the completion detection delay was to replace the completion tree with a delay line whose timing charateristics guarantee that the transitions on the data have been completed by the time the delay line makes a transition.

A major breakthrough in the design of the MiniMIPS was to show that timing assumptions are not necessary for high performance asynchronous pipelines. The completion-tree delays were reduced by introducing a new technique called *pipelined completion*, and high throughput was achieved by ultra-fine pipelining of pipeline stages. Hence, the MiniMIPS project dispels two myths at once. The first myth was that asynchronous circuits were necessarily slow. The second one was that, if asynchronous circuits could be made fast at all, it would be through heavy use of timing assumption and delay lines for completion. As we report here, the MiniMIPS achieves performances superior to that of equivalent clocked circuits and vastly superior to all other asynchronous microprocessors. At the same time, it remains entirely QDI except for the cache cores and the off-chip interface, hence preserving the robustness properties that are one of the main advantages of asynchronous QDI design.

3 The Architecture of the MiniMIPS

The specification of the microprocessor is the MIPS-1 instruction set as described in [1]. The structure of the processor, in terms of pipeline choice, exception implementation, register accesses, etc., is completely unrelated to existing synchronous architectures, and quite different.

We chose the R3000 because it is a "classic" RISC processor. It consists of a full 32-bit RISC CPU and a memory management unit. The CPU has 32 32-bit general-purpose registers, a program counter, and two special-purpose registers for multiplication and division. It has two operating modes, user and kernel. Branches have a single delay-slot. Instructions are issued one at a time. There are two four-kilobyte direct-mapped caches: an instruction cache, and a write-through data cache. Cache lines are one-word long, and refills are four-word blocks. Exceptions are precise. It is important to mention that the MiniMIPS is a single-issue architecture (instructions are fetched one at a time) since instruction fetch will turn out to be the main performance bottleneck.

The chip we fabricated as a first prototype is not the complete MIPS R3000 but a slighty reduced version. We have left out the TLB (address translation mechanism) which we found much too complicated, the partial-word memory operations, and some cache instructions.

The micro-architecture of the MiniMIPS is described in some detail in [2]. Here are some key points. In order to take advantage of the branch delay slot of one provided by the instruction set definition, instruction $i+1$ is fetched before instruction i is decoded. We say that we have "two instructions in the fetch loop." Also, in order to minimize the latency of computing branch targets, we use a simple branch prediction mechanism.

The instructions are decoded in sequence by the decoder, and then dispatched to the different execution units. Unlike synchronous pipelines, the execution units are not placed in any order in the pipeline, and thus instructions can terminate out of order. Writing back the results of the instruction executions to the registers is ordered only in so far as required by the precise exception mechanism, the data dependencies on registers, and the allocation mechanism for the two Z-busses (the busses used for writing back results in the register file).

Caches are pipelined: Each cache is decomposed into sixteen blocks that form the leaves of a routing tree. The blocks are organized in a four-by-four array. Read/write accesses to different leaves can take place concurrently. The design is pipelined so that the cache control issues a lookup to the cache core before deciding whether the previous lookup was a hit or a miss.

The throughput requirement we had fixed for the design was such that a critical cycle should contain about 18 transitions. Given that the return-to-zero protocol doubles the number of transitions on a cycle, this requirement implied that each pipeline unit had to be decomposed into a collection of very small pipeline stages. In particular, the 18-transition requirement pro-

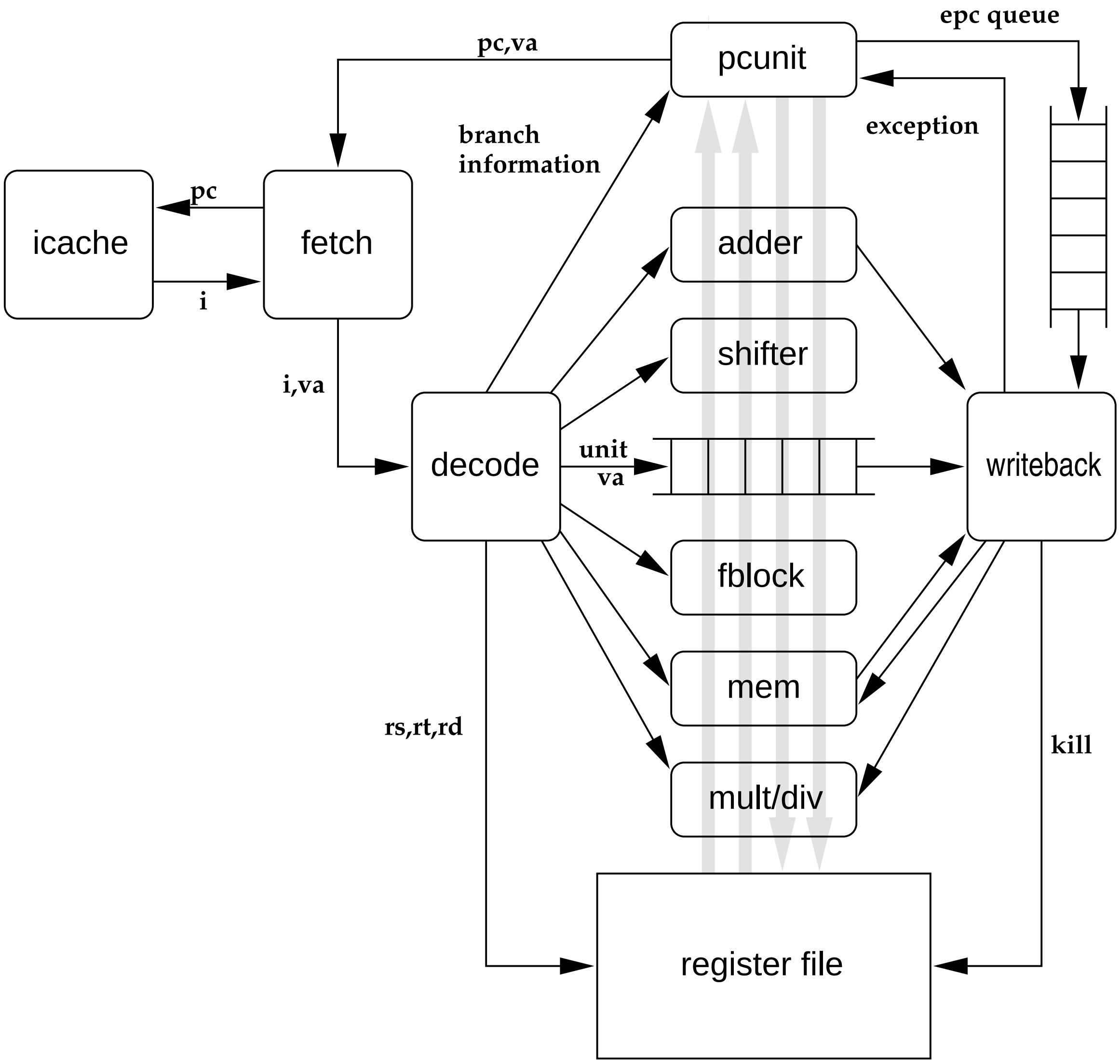

Figure 1: The MiniMIPS pipeline

hibits having a 32-bit completion tree on a pipeline stage. We therefore decomposed the data-path into four byte-slices and pipelined the different completion trees using a novel technique called pipelined completion. The combination of functional pipelining and data-path decomposition was called "two-dimensional pipelining." It is the only way to achieve high throughput in a QDI design style with return-to-zero protocol.

4 Performance Evaluation

The performance of a clocked microprocessor is the highest clock frequency at which the processor executes all instructions correctly at the standard power-supply voltage (Vdd) for the chosen technology. The power consumption is estimated by measuring the average current drawn from the power supply.

The performance of an asynchronous microprocessor is measured differently. First, there is no clock to adjust. The processor is running freely at the chosen power-supply voltage and settles at its natural frequency. Of course since the execution time of instructions varies with the nature of the instruction and also with the data manipulated by the instruction, the frequency may vary. But pipelining absorbs the variations to a large extent. In the MiniMIPS, the critical cycle is entirely determined by the rate at which the instructions are fetched from the cache, and does not depend on the rate at which they execute. Hence, the frequency is independent of the nature of the instructions except for branches, which add a fraction of a cycle delay to the cycle time. As in the case of clocked designs, the power consumption is estimated by measuring the average current drawn from the power supply.

5 The $E * t^2$ Metric

In CMOS, energy and delay can be traded against each other through voltage adaptation by varying the power-supply voltage Vdd. It is therefore impossible to evaluate the energy efficiency of a design independently from its speed efficiency. In order to provide a meaningful measure of energy efficiency, we therefore have to combine energy and delay (cycle time or latency) in a single measure of performance. In current CMOS technology, the bulk of the energy consumption is due to the switching energy—the energy spent charging and discharging the circuit nodes. This energy is proportional to the square of the voltage. The delay to switch a node is inversely proportional to the voltage if we assume that the current during switching is mostly the saturation current. In first approximation, the product $E * t^2$, where E is the average energy of an instruction execution, and t is the average cycle time, is independent of the voltage, and is therefore a good performance metric of a design at any voltage. (See [5] for more details.) We will use this metric to compare the MiniMIPS to other microprocessors.

6 Fabrication and Performance

This first prototype is a full-custom hand layout. It was fabricated in HP's 0.6 μm CMOS process via MOSIS (3-metal, 1-poly, linear cap, silicide block, n-well, 3.3 V). The area is 8 by 14 mm^2, and the transistor count is 2M (1.25M for caches).

From extensive SPICE simulations, the performance should have been 280 MIPS at 3.3 V. But this first prototype did not achieve these performances.

The performance of the fabricated batch varies significantly from chip to chip, with instruction fetch rates ranging from 160 MHz to 190 MHz. We chose to quote a fetch rate of 180 MHz as standard for this batch at a Vdd of 3.3 V at the pins and at room temperature. The average power consumption

Vdd	Freq (MHz)	Power (watts)	$E * t^2$ ($\times 10^{-24}$)
1.00	9.66	0.021	2.33
1.51	66	0.29	1.01
3.08	165	3.4	0.76
3.30	177	4.2	0.76
3.51	185	5.1	0.81
4.95	233.6	13.7	1.07

Figure 2: Performance measurements for a typical chip at room temperature.

is 4 watts. The test performances on small programs at different voltages are summarized in Table 1. The table includes the $E * t^2$ metric value in each technology.

The performance figures running Dhrystone are 185 MHz at 3.3 V on the pins and at room temperature, which gives a performance figure of about 165 VAX-MIPS.

The results suggest a typical performance of 180 MIPS at 4 W at 3.3 V at room temperature.

We expected 280 MHz, we got about 180 MHz. What happened? A performance loss of 20% was caused by the HP process. This particular run was slow, which brought the performance down to 220 MHz. The other 20% performance loss was caused by a layout error: a long polysilicon wire was forgotten in a single critical unit, and was unnoticed in SPICE simulations

R3000	1.2μ	25MHz	cpu only
R3000A	1.0μ	33MHz	cpu only
VR3600	0.8μ	40MHz	cpu+fpu

Figure 3: Performance of commercial processors similar in complexity and technology. Source:*Alpha Implementations and Architecture*, Dileep P. Bhandarkar, 1996, Digital Press

because resistance in long wires was not modeled. (We were relying on good layout discipline.)

In spite of this slight disappointment, the performance obtained on this very first prototype is excellent as indicated by the following comparisons.

7 Comparison with Commercial MIPS R3000 Microprocessors

We compare the speed performance of the MiniMIPS with that of commercial synchronous MIPS R3000 microprocessors. For the three designs used in Table 2, the clock rate is the fastest available at initial introduction. The source of the data is [7].

Since we could not find published figures for a MIPS R3000 fabricated in 0.6 μm technology, we have extrapolated the performance of a hypothetical

0.6 μm MIPS R3000 from the performances of the designs in Table 2. We assume that the same scaling factor is applied to the feature size, the thin oxide and the Vdd voltage. In such a case, the throughput scales linearly with the feature size. Figure 1 indicates that this is the case for the three MIPS R3000 of Table 2, and thus the linear extrapolation for feature size of 0.6 μm is reasonable. The hypothetical 0.6 μm synchronous MIPS R3000 would have a throughput of 47MHz, and thus the Caltech MiniMIPS is approximately four times as fast as a commercial synchronous MIPS R3000 in the same technology, and a straight redesign simply correcting the layout error would deliver performance about five times superior to its clocked counterpart.

8 Energy Efficiency

In this section, we compare the MiniMIPS to a number of commercial designs from the point of view of energy efficiency. The choice of the designs we use in the comparisons was mainly dictated by the availability of data. As we discussed it, adjusting the performances for technology differences (feature size) is very unreliable because the uncertainty of the calibration is amplified by the exponential relationship to the scaling ratio. We therefore limit the comparisons to commercial designs in approximately equivalent technologies. We give two sets of figures for the MiniMIPS: the entry labeled "mm-fab" corresponds to the performance measurements of the fabricated prototype; "mm-sim2" corresponds to the simulated performance of the design with the layout error corrected and with the same technology file as the fabricated version; "mm-sim1" corresponds to the simulated performance of the original design (without layout error) with the original simulation parameters from HP. The power and energy are computed per bit; and the frquency of the 21064 was adjusted assuming an instruction issue rate of 1.5. The results are collated in Figure 4.

Extensive SPICE simulations indicate that a direct redesign of the MiniMIPS in TSMC 0.18 μm technology (1.8 V nominal voltage) would give the following performance: a frequency of 560 MHz and power consumption 2.4 watts at 1.8 V, and an Et^2 of 13.5e-27. This represents a factor 90X improvement in Et^2.

9 Performance Estimate in FO4 delays

Clearly, comparing performances of microprocessors designed in different technologies is very difficult. Scaling factors are difficult to estimate, in particular because thin oxide is no longer scaled linearly; and even at the same feature size, different technologies have vastly different performance because of the recent introduction of new materials like copper wires or silk. For instance, the TSMC parameters we used to simulate the MiniMIPS seem to indicate that the process is slow compared to other technologies at the same feature size.

A better way to estimate and compare performances has been proposed by Mark Horowitz. The idea is to express the basic performance (throughput, energy per instruction) in terms of the perfomance of a standard gate in the chosen technology, for instance a fan-out-of-four inverter (FO4).

We computed the cycle time of the mm-sim1 MiniMIPS in terms of FO4. The FO4 inverter simulation gave a falling delay of 194.1 ps and a rising delay of 214.5 ps. Those delays give us a cycle time of 17.5 in FO4 units for the MiniMIPS assuming 280 MHz.

This performance is very similar to the best commercial designs measured

Name	word	Tech (μm)	Freq (MHz)	Power per bit	Et^2	Energy (e-10 J)
mm-sim1	32	0.6	280	0.219	1.0e-26	7.8
mm-sim2	32	0.6	230	0.16	1.312e-26	7
R4600	64	0.64	150	0.0719	2.13e-26	4.8
21064	64	0.6	200	0.469	2.135e-26	23.5
mm-fab	32	0.6	180	0.125	2.143e-26	7
R4400	64	0.6	150	0.234	6.944e-26	15.6
SH7708	16/32	0.5	60	0.018	8.3e-26	3
P6	32	0.6	150	1.8	5.2e-25	120

Figure 4: Source: "General Processor Information," by Tom Burd, April 2000

in FO4. According to Peter Hofstee [4], the IBM Rivina has a cycle time of 14.4 FO4, including design clock skew and jitter, but excluding skew due to process variation. The same source indicates that Intel's Willamette is a little slower than the MiniMIPS measured in FO4.

10 Redesign for Et^2 efficiency: Reaching 10000 MIPS/watt

As we said earlier, this first design was not optimized for Et^2 or power efficiency. In this section, we will briefly describe what a redesign of the MiniMIPS for Et^2 efficiency would require and what kind of improvement we could achieve.

First, transistor sizing for Et^2 is different from sizing for throughput only. We have proved that, in a circuit optimized for lowest Et^2, the sizing of transistors should be such that the total switching capacitance is twice the total parasitic capacitance. Except for designs with unusually many long connections, this requirement results in smaller transistors than when sizing for throughput alone. (See [6].) We estimate that a resizing of the transistors alone would deliver a two- to three-fold Et^2-improvement in the same technology.

Second, extensive and accurate simulations of the fabricated design gives use invaluable insight in the relative contributions of the components of the MiniMIPS to the energy consumption. For instance, during execution of a NOOP, half of the energy is spent in the buffers. For a standard instruction (add), the breakdown is as follows: 44% for instruction fetch, 44% for moving operands and result, 10% for instruction execution, 2% for decode. Hence, 90% of the energy is spent transferring data.

Third, the throughput is currently limited by the rate at which instructions are fetched from the cache. Since the current design fetches one instruction at a time, a new design implementing multiple instruction issue would improved the Et^2 performance, since the energy consumption would hardly be changed by the multiple instruction issue.

We are considering two possible redesigns, one with multiple instruction issue and one without.

The redesign without multiple instruction issue would include: transistor resizing, unpipelining some units, reducing and modifying buffers, reducing the number of bits transmitted in the fetch stage and in the execution stage. In TSMC 0.18μm technology, we expect this redesign to deliver a 270X Et^2 improvement of the current design, with the following performance figures: 410 MHz, 0.32 W, Et^2 of 4.5e-27. *At 0.7 V, this redesign should deliver 10000 MIPS/W.*

A redesign with multiple instruction issue could deliver up to 400X improvement in Et^2 compared to the current design.

11 Conclusion

As of today, the Caltech MiniMIPS has the best energy and speed performance of all asynchronous designs. This paper shows that it also compares very favorably with commercial synchronous designs for both throughput and energy efficiency. Because of prevailing skepticism towards asynchronous techniques, we have strived to provide irrefutable evidence of the performance of our design. The figures presented are based either on direct measurements taken in the lab on the fabricated prototype, or on careful SPICE simulations calibrated on the actual measurements. They should provide a useful data point for any evaluation of this design style.

The design of the MiniMIPS was our first large-scale experiment with this fine-grain pipelining approach. There is certainly a lot of room for improvements at all levels, transistor sizing, basic cell-design, pipeline depth, buffers design, cache design and sizes, instruction issue, etc. Finally, it should be observed that a MIPS ISA is certainly not the best choice for an asynchronous implementation since it has been designed to fit all instruction executions into an exact multiple of a clock cycle, and therefore such an ISA makes it difficult for an asynchronous implementation to take advantage of the flexibility offered by the absence of delay constraint.

Acknowledgments

The research described in this paper was sponsored by the Defense Advanced Research Projects Agency and monitored by the Air Force under contract F29601-00-K-0184.

References

[1] G.Kane and J.Heinrich. MIPS RISC Architecture. Prentice-Hall, 1992.

[2] Alain J. Martin, Andrew Lines, Rajit Manohar, Mika Nyström, Paul Penzes, Robert Southworth, Uri Cummings and Tak Kwan Lee. The Design of an Asynchronous MIPS R3000 Microprocessor. *Proceedings of the 17th Conference on Advanced Research in VLSI*, IEEE Computer Society Press, 164-181, 1997.

[3] A.J. Martin, S.M. Burns, T.K. Lee, D. Borkovic, P.J. Hazewindus. The Design of an Asynchronous Microprocessor. *Decennial Caltech Conference on VLSI*, ed. C.L. Seitz, MIT Press, 351-273, 1989.

[4] H. Peter Hofstee. Private Communication, March 2000.

[5] Alain J. Martin. Towards an Energy Complexity of Computation *Information Processing Letters*, 77, 2001.

[6] Paul Penzes and Alain J.Martin. Global and Local Properties of Asycnhronous Circuits Optimized for Energy Efficiency. IEEE Workshop on Power Management for Real-time and Embedded Systems, May 2001.

[7] Dileep P. Bhandarkar. *Alpha Implementations and Architecture*, 1996, Digital Press.