

CSE 560M - Computer Systems Architecture I

Final Exam

Shakir James

1 Introduction

This description of the MIPS R10000 (R10K) processor includes details about its organization, operation, and memory hierarchy. We begin by discussing the microarchitecture (hardware organization) and operation of the MIPS R10K. Then we discuss the operation and physical characteristics of its memory hierarchy.

2 Microarchitecture & Operation

The R10K is superscalar, dynamically scheduled, RISC processor that exploits ILP [H&P, A&C]. Four R10K processors can be directly connected (via the system bus) to exploit some TLP [A&C]; the multiple R10K configuration does not use a shared cache.

The main parts of the microarchitecture are: i) instruction fetch & branch prediction, ii) instruction decode and register renaming, iii) instruction dispatch, iv) instruction execution and commit [S&S].

2.1 Instruction Fetch & Branch Prediction

The R10K implements an integrated instruction fetch unit [H&P]. The fetch unit includes integrated branch prediction, and may possibly include instruction pre-fetch [H&P]. The instruction buffer "stockpiles" instructions to mitigate the effect of cache misses and branches [A&C, S&S].

The processor fetches up to four instructions per cycle [A&C]. Thus, a separate instruction cache helps to sustain this "high bandwidth instruction fetching" [S&S]. Instructions in the instruction cache are pre-decoded to reduce latency of the fetch process [A&C]. The pre-decode stage re-arranges the (memory) format of instructions, and adds four *unit* bits to each instruction [A&C]. The *unit code* selects which functional unit the instruction will use. The *unit*-bits are "pre-decode bits" that simplify instruction-type identification logic [S&S].

An eight entry TLB converts the virtual address of an instruction to its physical equivalent. The virtual address is input into the Branch History unit perhaps to determine if the current instruction is a branch. The eight-entry TLB refills from the main TLB; a refill maybe caused by a context switch.

The presence of a branch predictor [A&C] indicates the use of dynamic branch prediction; the prediction is probably based outcome of previous branches. The virtual address indexes the 512x2-bit Branch History unit [A&C] or Branch-Prediction Buffer [H&P]. The Branch History unit two-prediction bits to the Branch predictor. The 4-entry *Branch Stack* [A&C] may hold instructions on the not-taken path of a predicted taken branch [S&S]. The instruction in the branch stack maybe used to recover from a maximum of four mispredicted branches.

2.2 Instruction Decode and Register Renaming

Alignment and decoding of the four fetched instructions occur in parallel [A&C]. Since there is only one branch predictor, decoding of only one branch instruction is possible per cycle.

32x6-bits Integer and Floating-point (FP) *Mapping Tables* map (32-bit) logical registers to (64-bit) physical ones [A&C, S&S]. The physical registers are assigned by checking the appropriate 32x6-bits *Free List* for available registers [A&C]; we assume a separate *Free List* for Integer and FP registers. The *rs*, *rt*, and *rd* logical registers map to *OpA*, *OpB*, and *Dest* physical registers [A&C], therefore the mapping tables have 12 read ports to decode three operands of four instructions in parallel.

An *Active List* contains 32 instructions in execution order [A&C]. This "ordered list of instructions" includes a Done (*D*) bit, Exception bits (*Excp*t), the old physical destination register (*OldD*), and the Logical Destination (*LDest*) [A&C]. The *Excp*t bits together with the *D* bit is probably part of the mechanism to preserve precise exception behavior. The illusion of the "sequential execution model" [S&S] is maintained by the register renaming process; therefore, the primary function of a ROB is replaced by the register renaming process [H&P].

2.3 Instruction Dispatch

After an instruction is decoded, it issues to one of the three instruction queues: integer, address, or floating point. Each instruction queue holds a maximum of 16 entries [A&C]. The *Op*, *sa*, and *funct* bits make up the function code field (*Func*) which identifies the instruction queue and execution unit [A&C]. The *OpA*, *OpB*, and *OpC* fields are pointers to the operands' physical registers, and the *Dest* field points to the physical destination register [A&C, S&S]. Instructions from the queues issue dynamically; dispatch is not in program order [S&S].

The Microarchitecture of Superscalar Processors paper [S&S], refers to (in Section 4.1) a reservation bit that is set "to busy" for each physical result register at the time of instruction dispatch. The same paper also mentions that the reservation bits for the operands are checked to ensure they are "not busy" before the instruction is issued.

2.4 Instruction Execution and Commit

Issued instructions are processed by one of the execution pipelines. Each pipeline has a RF and Result stage in common. The RF stage reads operands directly from the appropriate 64x64-bit register file, and the Result stage writes the results back to the register file [S&S]. The Result stage is equivalent to the WB stage in the classic five-stage pipeline; similarly, the RF stage is like the ID stage. There are five execution units: two integer ALUs, an address adder, a FP adder, and a FP multiplier [A&C, S&S]. Thus, there are five execution pipelines.

The two Integer Pipelines have four stages: Issue, RF, ALU, and Result. Both ALU units have a Latency of one; i.e. results are available on the next cycle.

The Load/Store Pipeline has five stages: Issue, RF, Address Calculation, Data Cache, and Result. Effective address calculation (integer addition) probably occurs in the Address Calculation stage; the Address Calculation unit has a latency of two. The Data Cache phase accesses the data cache by first accessing the TLB for virtual-to-physical address calculation.

There are two FP pipelines: add and multiply. Each FP unit in these pipelines are also fully pipelined [A&C]. The FP multiplier includes an iterative division and square root unit. Both iterative units use adder logic (*CPA*) as their final step [A&C]. The latency of FP adder and multiplier units is two.

Instructions retire in program order. The physical registers of committed instructions go back to the appropriate *Free List*. [A&C]

3 Memory hierarchy

The R10K uses a load/store architecture. It implements an on-chip L1 data and instruction cache, and an off-chip unified L2 cache. The main memory unit connects to both the L1 and L2 caches. The processor also uses a TLB for VM support.

3.1 L1 Cache

Like most superscalar processors, the R10K implements a separate L1 data and instruction

cache to sustain the "high instruction fetch bandwidth" [S&S]. To fit on-chip, the L1 caches are relatively small. Using on chip caches reduce the time penalty of having to go off chip [H&P].

The 32KB instruction cache is 2-way set associative; therefore, it has a miss rate of a direct-mapped cache of 64K [H&P]. The instruction cache uses 16-word blocks (64 bytes, assuming 32 bits per word); thus, 6-bits specify the block offset. The number of sets is $32 \text{ KB} / (64 * 2) = 512$ sets; thus, the index field requires 9 bits. The remaining $64 - 6 - 9 = 49$ bits represent the tag field. We can speculate that the instruction cache may include some extra bits for way prediction to reduce the miss rate [H&P]. A TLB with 64-dual entries speeds up address translation [A&C]. Address translation and instruction access is probably overlapped [S&S]. Cache access is unaligned and requires four cycles to refill a 16-block word (or 128 bit per cycle) [A&C].

There is also a high performance 32KB, write-back, data cache [A&C]. The data cache is non-blocking [A&C] to allow overlapping of memory operations; thus, a cache miss will not preclude other instructions from proceeding [S&S]. Simulated performance results show the non-blocking cache significantly reduces the miss latency, by as much 47% in tomcatv benchmark [A&C]. It also appears that the data cache uses way prediction. Furthermore, the data cache uses two techniques to reduce the hit time: interleaving and pipelining [A&C, H&P].

The L1 data cache is split into two 16K banks to increase cache bandwidth and take advantage of parallelism. Each bank is two-way set associative. A bank includes two arrays of memory cells (one array for each way). Each bank uses 8-word blocks. The CPU interleaves accesses to both banks on a 64-bit doubleword. External access, via the external interface, is on (128-bit) quadwords as two doublewords in parallel [A&C]. The interleaving allows words to be read/written sequentially which improves the performance of the write-back data cache (since writes and reads are sequential) [H&P]. Additionally, two banks can "reduce conflicts for resources" [A&C].

The data cache is also pipelined. Pipelined cache access increases the data bandwidth as opposed to reducing the hit latency [H&P]. There are three pipelines: address calculation/translation, tag check, and data load/store [A&C]. The calculation/translation pipeline uses the Address Calculation unit and the TLB. The Tag check pipeline performs cache tag comparison to check for a cache hit. In the event of a cache miss, a refill from the secondary cache initiates. Two instructions can initiate a refill in parallel. In other words, the cache blocks on a second miss [A&C]. The load/store stage of the pipeline handles memory accesses via separate load and store units.

The *Secondary Cache Ctlr* connects the primary caches to the unified secondary cache [A&C].

3.2 L2 Cache

The R10K supports 512K-16MB, 2-way associative, SSDAM, L2 cache. A 234-bit Secondary Cache interface connects the off-chip, L2 cache to the CPU. The 234 bits include 19 bit replicated address bus, 26-bit tag bus with 7-bit ECC, and 128-bit data bus with a 9-bit ECC. The L2 cache supports clock dividers of 1, 1.5, 2, 2, 5, and 3; therefore, it is possible to have a L2 cache running in sync with the processor clock.

The L2 cache uses a MRU replacement algorithm. On a cache miss, the system first reads the 8192-bit MRU Table. Then a sequential, the systems sequentially checks the tags in each of the two ways; thus, a miss in one way initiates a read of the second way. If both ways miss then the L2 must refill for the main memory.

The secondary cache implements error protection, which includes both detection and recovery [A&C]. The cache can detect multi-bit error using the ECC Check unit and recovery from single-bit errors using the ECC Corrector. It appears the memory protection system can run in two modes: fast, normal. In fast mode, a simple odd-parity check prevents the loading of invalid data. In normal mode, the check checks the ECC bits and performs single bit recovery, if necessary.

The 64-bit system bus connects both the L2 and L1 caches to main memory.

3.3 Memory

The R10K supports a 64-bit physical address [A&C]; an address space of 18 quintillion bytes! The External Agent connects main memory to the CPU. The External Agent sends messages from the "rest of the world" to the CPU over the 64-bit system interface bus [A&C].

The system interface bus provides internal caching and buffering mechanisms, and connects directly to both caches. The direct connection to the caches reduces memory latency, and is probably part of a data coherency model.

4 References

[A&C] Ali Ahi, Yung-chin Chen, et al. "R10000 Superscalar Microprocessor"

[S&S] James E. Smith and Gurindar S. Sohi. "The Microarchitecture of Superscalar Processors"

[H&P] Hennessy and Patterson. *Computer Architecture: A Quantitative Approach 3rd Ed*