CSE 240A Test case: Naughty Boy

M Barrow

Introduction

This document describes a basic test for a MIPS R10000 (R10k) simulator and provides expected results of that test. Because certain details of the MIPS R10000 implementation are unclear from the assigned reading, “The Mips R10000 Superscalar Microprocessor”[1]. Some design decisions that affect execution have been taken and will be described prior to the expected results.

Test motivation

This test is intended to complement a suite of crowd sourced test traces that verify a plethora of R10k simulators submitted by the 240A cohort.
It was expected that most of the class test cases would focus on edge cases of the R10k and that most of the cohort would make unique design decisions when realizing the complex R10k processor. In contrast, this test is designed to be simple to maximize the commonality of results between the cohort and simplify functional verification.
It is also designed to focus on the learning outcome of this project, i.e. the out of order execution and in order commit possible in the R10k.
If each simulator is able to demonstrate register mapping, parallel execution and in order commit with this test, then each member of the cohort is able to demonstrate a germ of understanding with regards to the course learning outcomes.

Notable design decisions

As specified in the project description, register files are fully unified. That is to say all instructions may be dependent on others for their registers, regardless of instruction class. For example, branches and integer trace lines may depend on the destination registers of floating point instructions.
A conservative approach was taken with the load store unit due to lack of details on the exception handling mechanism of the R10k. It is understood that the memory unit is pipelined for increased CPI of non-contentious memory operations (RAR, WAW, non-dependence etc.). However it is also understood that the R10K is capable of hardware exceptions[2]. To guarantee memory consistency without details of the exception handling mechanism, a decision was taken to fully execute all memory operations sequentially, with a cycle separating graduation and execution of each memory operation.
It is supposed that all traces represent test programs and that the r10k has been bootstrapped. As such ISA registers will be mapped to machine registers the first time they occur in a trace. Therefore the initial state of the register map table is empty and populated by the trace during simulation.

Test design

The test is designed to demonstrate full utilization of the fetch decode execute bandwidth, in order commit and the conservative load/store implementation. This test will demonstrate exotic deviations of implementation with only 7 instructions through comparisons of the cohorts register mapping tables and output traces. The trace was originally called “Naughty Boy” since no variation of the R10K would allow out of order commit of the integer instruction (see test trace line 4 below).

Test format

The format of the trace follows the project specification, with some clarifications provided by class

discussion. i.e.

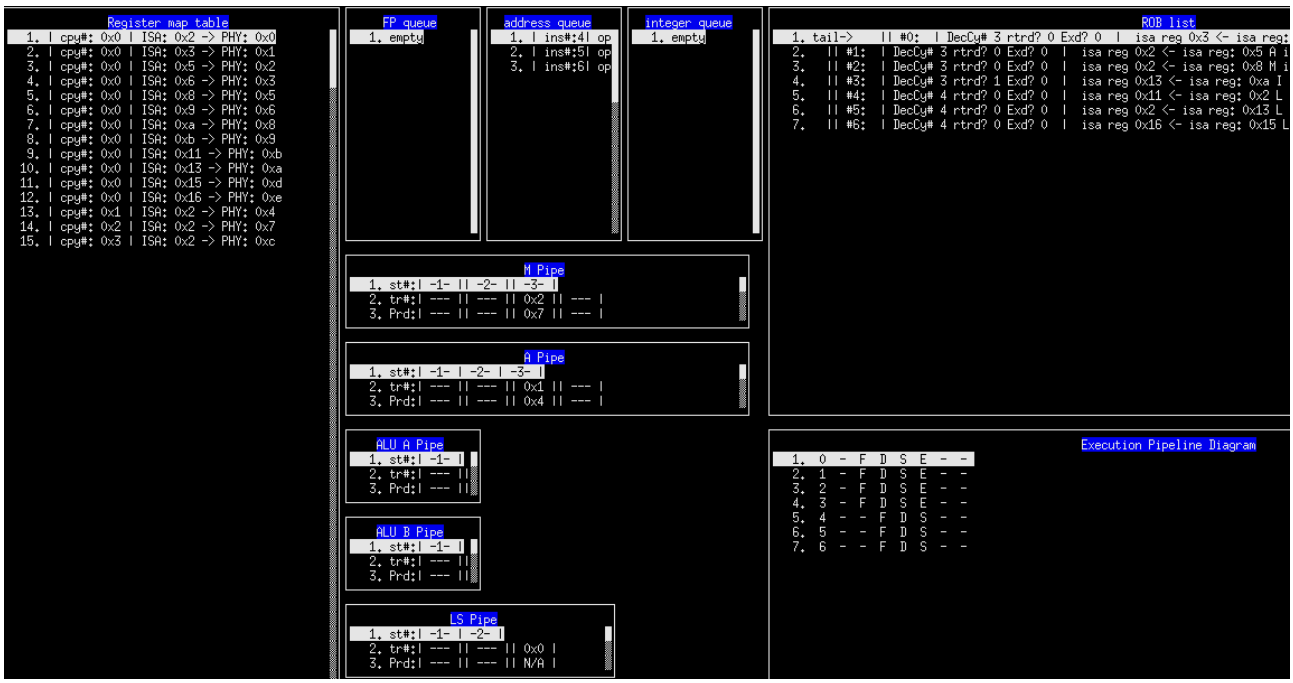
Instruction type	Trace format	Actual mapping
F,A,I	<op> <rs> <rt> <rd> <extra>	<op> <rs> <rt> <rd> <XX>
L	<op> <rs> <rt> <rd> <extra>	<op> <rs> <rd> <XX> <extra>
S	<op> <rs> <rt> <rd> <extra>	<op> <rs> <rt> <XX> <extra>
B	<op> <rs> <rt> <rd> <extra>	<op> <rs> <XX><XX> <extra>

Test trace

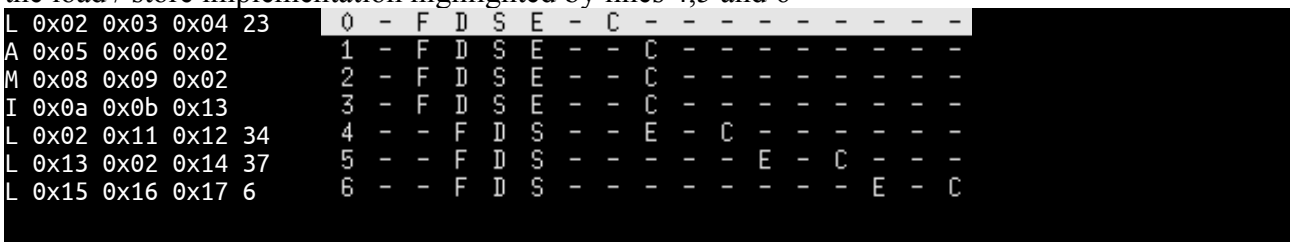
L 0x02 0x03 0x04 23
A 0x05 0x06 0x02
M 0x08 0x09 0x02
I 0x0a 0x0b 0x13
L 0x02 0x11 0x12 34
L 0x13 0x02 0x14 37
L 0x15 0x16 0x17 6

Expected outcome

Fully populated register map table on cycle 4 demonstrating register mapping design decisions. It is expected other implementations will vary in their mappings.



Complete pipeline diagram demonstrating full parallel execution of trace lines 0,1,2 and 3 as well as the load / store implementation highlighted by lines 4,5 and 6



- The L,A,M and I are fetched, decoded and executed in parallel
- The Load instruction commits 1 cycle later due to the 2 cycle pipeline length of the LS unit
- The A and M instructions commit 2 cycles after execution due to the 3 cycle pipeline length of the floating point pipelines
- The I instruction commits at the same time as the A and M instructions because of the in order commit of the processor. G**raduating earlier is a definite failure of this benchmark**
- The subsequent L instructions execute in sequence with a cycle space between Graduation and execution for memory continuity. **Variance is not a definite failure of this benchmark**

Conclusions

Even a simple test such as naughty boy is expected to show variations between project implementations, however Naughty boy is a concise useful functional test case to demonstrate the kernel properties of in order commit and parallel execution of the R10k

References

[1]Yeager, Kenneth C. "The MIPS R10000 superscalar microprocessor." *Micro, IEEE* 16.2 (1996): 28-41.
[2] MIPS Technologies “MIPS R10000 Microprocessor User's Manual”
http://techpubs.sgi.com/library/manuals/2000/007-2490-001/pdf/007-2490-001.pdf (1997)