EE360N: Computer Architecture Lecture #3
Department of Electrical and Computer Engineering
The University of Texas at Austin January 27, 2010

Disclaimer: The contents of this document are scribe notes for The University of Texas at Austin EE360N Fall 2008, Computer Architecture[*] The notes capture the class discussion and may contain erroneous and unverified information and comments.

Microcoded Machines

Lecture #3: 01/27/2010
Lecturer: Derek Chiou
Scribe: Ankit Bansal, Kiran Divakar

1. Recap of Last Lecture and Outline of This Lecture

ISAs depend on the underlying micro-architecture of the computer. Microcoded machines are easy to understand, implement, and add instructions to. It is also easy to implement ISAs on top of microcoded machines. However, a downside to the ease of adding instructions to a microcoded machine is that it often results in a proliferation of instructions, making the ISAs too complex. For example, the VAX polynomial solve instruction may have seemed like a good idea at the time it was created because it was easy to create and someone had a need for it, but it quickly became a liability. To maintain software backward compatibility, instructions cannot disappear from an instruction set. Hence addition of instructions, even though easy, should be done judiciously.

This lecture builds more upon the control structure of the microcode engine and improving microcoded machine performance.

2. Changing Instruction Sets

ISAs act as an interface between hardware and software and hence changing an ISA would require changes at both ends making it a difficult and expensive task. But sometimes it is advantageous to do so to adapt to technology changes and to enable better microarchitectures. For instance, the VAX instruction set was originally implemented as a microcoded machine, but with time complex instructions like the polynomial solve instruction were added, making it difficult to implement efficiently. So Digital Equipment Corporation (DEC or Digital) switched instruction sets from the VAX instruction set to the Alpha instruction set, which was much simpler and efficient to implement. The machines that used Alpha ISA were so efficient that their simulation speed of the x86 instruction set was comparable to an x86 processor.

Apple is another example of a company which switched ISAs not once, but twice. Apple switched from the Motorola 68K ISA to PowerPC to get IBM's and Motorola's support in their bid to stay competitive against WinTel. Later on when once when it became clear that the