

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Curso: Programa de Residência em Tecnologia da Informação para a GOOGLE

Disciplina: Arquitetura de Computadores

Sala 3010 ICEx, SQ 09:25-11:05 h

1o. Semestre de 2007

Home Page: www.dcc.ufmg.br/laboratorios/verlab/wiki/cursos/arquitetura

Professor: Mario Fernando Montenegro Campos (mario@dcc.ufmg.br)

Monitor: Antônio Celso Caldeira Júnior (caldeira@dcc.ufmg.br)

LISTA DE EXERCÍCIOS 2

1) Several researchers have suggested that adding a register-memory addressing mode to a load-store computer might be useful. The idea is to replace sequences of

LOAD R1,0(Rb)

ADD R2,R2,R1

by

ADD R2,0(Rb)

Assume the new instruction will cause the clock cycle to increase by 10%. Use the instruction frequencies for the gcc benchmark on the load-store computer from Figure B.27 (page B-41). The new instruction affects only the clock cycle and not the CPI.

- a. What percentage of the loads must be eliminated for the computer with the new instruction to have at least the same performance?
- b. Show a situation in a multiple instruction sequence where a load of R1 followed immediately by a use of R1 (with some type of opcode) could not be replaced by a single instruction of the form proposed, assuming that the same opcode exists.

2) Your task is to compare the memory efficiency of four different styles of instruction set architectures. The architecture styles are

1. Accumulator - All operations occur between a single register and a memory location.
2. Memory-memory - All three operands of each instruction are in memory.
3. Stack - All operations occur on top of the stack. Only push and pop access memory; all other instructions remove their operands from stack and replace them with the result. The implementation uses a stack for the top two entries; accesses that use other stack positions are memory references.
4. Load-store - All operations occur in registers, and register-to-register instructions have three operands per instruction. There are 16 general-purpose registers, and register specifiers are 4 bits long.

To measure memory efficiency, make the following assumptions about all four instruction sets:

- The opcode is always 1 byte (8 bits).
- All memory addresses are 2 bytes (16 bits).
- All data operands are 4 bytes (32 bits).

- All instructions are an integral number of bytes in length.

There are no other optimizations to reduce memory traffic, and the variables A, B, C, and D are initially in memory. Invent your own assembly language mnemonics and write the best equivalent assembly language code for the high-level-language fragment given. Write the four code sequences for

$A = B + C;$

$B = A + C;$

$D = A - B;$

Calculate the instruction bytes fetched and the memory-data bytes transferred. Which architecture is most efficient as measured by code size? Which architecture is most efficient as measured by total memory bandwidth required (code + data)?

3) Suppose we have a deeply pipelined processor, for which we implement a branch-target buffer for the conditional branches only. Assume that the misprediction penalty is always 4 cycles and the buffer miss penalty is always 3 cycles. Assume 90% hit rate and 90% accuracy, and 15% branch frequency. How much faster is the processor with the branch-target buffer versus a processor that has a fixed 2-cycle branch penalty? Assume a base CPI without branch stalls of 1.

4) Determine the improvement from branch folding for unconditional branches. Assume a 90% hit rate, a base CPI without unconditional branch stalls of 1, and an unconditional branch frequency of 5%. How much improvement is gained by this enhancement versus a processor whose effective CPI is 1.1?