4/26/21

1. Speedup doing conditional jumps by a factor of 10

machine, as a whole, speedup of 2 in execution time

what fraction of the original execution time, before optimization, was spent doing conditional jumps?

$$\text{Speedup}_{\text{system}} = \frac{1}{(1-x)+\frac{x}{nn}}$$

$$2 = \frac{1}{1-x+\frac{x}{10}}$$

$$2-2x+\frac{2x}{10}=1$$

$$1-2+2x=\frac{2x}{10}$$

$$10-20+20x=2x$$

$$-10=-18x$$

$$x=.556$$

Original execution time = 55.6%

2. What is theoretical maximum speedup by speeding up conditional jumps?

$\lim_{x \to \infty} = \frac{1}{1-0.556} = 2.25$

2.25 is the theoretical maximum speedup

3. ALU instructions make up 25% and take 2 cycles to execute. Load/store instructions take 10 cycles to execute and make up 30%. Jumps take 4 cycles and make up 15%. All other instructions average 1.5 cycles. What is the average CPI?

Non-pipelined

$(0.25 \times 2) + (0.30 \times 10) + (0.15 \times 4) + (0.30 \times 1.5) =$ average cpi

$1 - \begin{array}{r} 0.25 \\ 0.30 \\ +0.15 \\ \hline 0.30 \end{array} \rightarrow$

$= 4.55$ cycles

4. Suppose the architecture above is pipelined. If there are no stalls, what is the new CPI?

if $\frac{\cancel{c}4.55}{4} = 1.1375$ cycles per instruction

total instructions

5. If for the architecture described in questions 3 & 4, load/store instructions generate 2 stalls on average, alu 0.2 stalls, and jumps cause 1 stall, what is actual pipelined cpi?

$(3 \times .30) + (1.2 \times 0.25) + (2 \times 0.15) + (1 \times 0.30)$

$= 1.8$ cpi

6. Describe two CPU design techniques that exploit instruction level parallelism to give an average CPI of less than one?

VLIW (very long instruction word): all instructions are executed at the same time without checking for dependicies, since the compiler generated them with respect to the data dependencies

Superscalar: extract ILP at run time from legacy code, and execute instructions in parallel on multiple pipelines / functional units

7. In the MIPs processor we designed in lab, what is the purpose of the LMD register.

LMD (local memory data) register

if the instruction is load, data returns from memory and placed in LMD

8. Arrange the following list of storage types in order from fastest access to slowest access: cloud storage, registers, main memory, level 1 cache, level 2 cache, hard drive

cloud storage — hard drive — main memory — level 2 cache — level 1 cache — registers

slow to access — fast to access

9. Why was cache memory common on mainframe computers in the 1960's with magnetic core memory, and in today's microprocessor-based computers, but not in computers of the late 1970's and early 1980's which used early integrated-circuit-based DRAM and SRAM memory?

In the time between, main memory and CPU were a comparable speed to each other. As the CPU got faster, and main memory remained the same, a cache was needed to access main memory faster than the CPU could.

10. In a machine with 32-bit physical addresses, a cache line is 256 bytes in size. The cache can hold 32K bytes. It is 2-way set associative. How many bits is the index value used with this cache?

$\frac{32K}{256} = 128 = 2^7$ sets

7 bits for index

11. In cache system above, the following sequence of memory reads occurs. At the end of the sequence, what blocks will be in each set of the cache? Give set number and tag (in binary) for the non-empty blocks in each set. Assume the replacement policy is to discard the oldest block in the set.

8 bits offset

8

32 17

tag

	index	offset
0x5fff 03e2 =.. 0000011	0000011	1110 0010
0x5ff1 03f7 =..	0000 0001	1111 0011
0x5fff0 3ab =.	10000001	1111 0111
0x5f1f 5100 =..0	1010001	0000 0000
0x0000 0000 =.0	0000000	00000000
0x0000000 1 =..0	0000000	00000001

[illegible] 0000 0011 1110 [illegible]

Set 1111 0001 contains 0000 0011 1111 0111

12. Why does RISC architecture typically not implement register-memory ALU operations as part of the instruction set?

RISC only uses simple instructions that take one clock cycle, and that operation would take more than one

13. Assuming there are 1.8 memory references per instruction in a machine with a single-level integrated instruction-and-data cache, and the hit time for the cache is 1 cycle, the miss time is 10 cycles, and the miss rate is 1%. What is the average time for memory access per instruction?

$1.8+[1+(.01\times10)]=1.8+1.1)$
$=2.9$ average cycle memory access time