CSCE 5610 – Computer System Architecture
Assignment 3

Uday Bhaskar Valapadasu - 11696364

1Ans:

a) Base CPU Time (Without Memory Stalls) –

Finding cycles for each instruction provided using the formula:

Cycles = Instruction Count / IPC

- Branch/Jump = 2*10^5 / 1.1 = 181,818.18 cycles
- Load/Store = 4.1∗10^4 / 0.9 = 45,555.56 cycles
- Arithmetic = 3.6∗10^6 / 1.5 = 2,400,000 cycles
- Float Add: = 5.5∗10^5 / 0.25 = 2,200,000 cycles

Total Base Cycles:
Total base cycles = 181,818.18+45,555.56+2,400,000+2,200,000 = 4,827,373.74 cycles

Base CPU Time:
Base CPU Time= Total Cycles / Clock Frequency

Plugging in the values:
Base CPU Time = 4,827,373.74 / (4×10^9)
$=1.207\times10^{-3}$ seconds

b) Memory Stall Time

Memory Stall Cycles = Memory Accesses × Miss Rate × Miss Penalty
= Instruction misses cycles + Data miss cycles
= (all Instructions * Instruction Miss Rate * Miss Penalty) +
(Instructions(load/store) * Data Miss Rate * Miss Penalty)

Total Instructions = 200,000+41,000+3,600,000+550,000 = 4,391,000

Instruction Access Stalls (for total instructions = 4,391,000):

1. L1 Instruction Misses: 4,391,000×0.02×2 = 175,640 cycles
2. L2 Instruction Misses: 4,391,000×0.013×5 = 285,415 cycles
3. Main Memory Instruction Misses: 4,391,000×0.01×100 = 4,391,000 cycles

Total Instruction Miss Cycles=175,640+285,415+4,391,000 = 4,851,055 cycles

Data Access Stalls (for Load/Store instructions):

1. L1 Data Misses: 4.1×10^4 × 0.05×2 = 4,100 cycles
2. L2 Data Misses: 4.1×10^4×0.02×5 = 4100 cycles
3. Main Memory Data Misses: 4.1×10^4×0.003×100 = 12,300 cycles

Total Data Miss Cycles = 4,100+4,100+12,300 = 20,500 cycles

Total Memory Stall Cycles:
Total memory stall cycles = 4,851,055 + 20,500 = 4,871,555 cycles

Memory Stall Time = Total Memory Stall Cycles / Clock Frequency
= 4,871,555 / (4 × 10^9)
= 0.00121788875 Seconds

(c) Total CPU Time

1. **Total Cycles:**

Total Cycles=4,827,373.74+4,871,555 =9,698,928.74 cycles

2. **Total CPU Time:**

Total CPU Time=9,698,928.74 / (4×10^9) = 0.002424732185 seconds

(d) Percentage of Time Taken by Memory Stalls
Using the memory stall percentage formula:
Memory Stall Percentage = (Total CPU Time / Memory Stall Time) ×100
Memory Stall Percentage: = (0.00121788875 / 0.002424732185) × 100 = 50.23%

2Ans:

Given Information

1. **8 KB Cache**:
   - Hit time = 0.22ns
   - Miss rate = m_1
   - Miss penalty = 100ns
2. **64 KB Cache**:
   - Hit time = 0.52
   - Miss rate = m_2
   - Miss penalty = 100ns
3. **Average Access Time Formula**:

t_s =Hit time of the smaller cache + (Miss rate of smaller cache * Miss penalty)

t_1 = Hit time of the larger cache + (Miss rate of larger cache * Miss penalty)

Part (a): Solving with a Miss Penalty of 100 ns
The smaller cache will be advantageous when:

$$t_s < t_1$$

Substitute the given values:

$$0.22+ m_1 \times 100 < 0.52+ m_2 \times 100$$

Rearrange to isolate $m_1 - m_2$:

$$(m_1 - m_2) \times 100 < 0.3$$

Divide both sides by 100 to find:

$$\mathbf{m_1 - m_2 < 0.003}$$

This inequality indicates that the miss rate difference ($m_1 - m_2$) should be less than 0.003 (or 0.3%) for the smaller cache to be advantageous with a miss penalty of 100 ns.

Part (b): Solving for Miss Penalties of 10 and 100 Cycles

- **Miss Penalty = 10 Cycles (Assuming 1 ns per cycle, so 10 ns total)**

Substitute the given values:

$$0.22 + m_1 \times 10 < 0.52 + m_2 \times 10$$

Rearrange to isolate $m_1 - m_2$:

$$(m_1 - m_2) \times 10 < 0.3$$

Divide both sides by 100 to find:

$$\mathbf{m_1 - m_2 < 0.03}$$

Here, for a 10 ns miss penalty, the smaller cache is advantageous if $m_1 - m_2 < 0$ or, if the miss rate difference is less than 3%.

- **Miss Penalty = 100 Cycles (Assuming 1 ns per cycle, so 100 ns total)**

Substitute the given values:

$$0.22 + m_1 \times 100 < 0.52 + m_2 \times 100$$

Rearrange to isolate $m_1 - m_2$:

$$(m_1 - m_2) \times 100 < 0.3$$

Divide both sides by 100 to find:

$$\mathbf{m_1 - m_2 < 0.003}$$

For a 100 ns miss penalty, the smaller cache is advantageous if $m_1 - m_2 < 0.003$, or if the miss rate difference is less than 0.3%.

Conclusion

The inequalities show that the smaller cache might be more advantageous when the difference between miss rates ($m_1 - m_2$) is within a certain threshold, which decreases as the miss penalty increases. Thus, as the miss penalty grows, the larger cache becomes more favorable unless the smaller cache has a very low miss rate close to the larger cache's miss rate.