Homework1

- 1.1 Your company has just bought a new Intel Core i5 dualcore processor, and you have been tasked with optimizing your software for this processor. You will run two applications on this dual core, but the resource requirements are not equal. The first application requires 80% of the resources, and the other only 20% of the resources. Assume that when you parallelize a portion of the program, the speedup for that portion is 2.
- a. Given that 40% of the first application is parallelizable, how much speedup would you achieve with that application if run in isolation?
- b. Given that 99% of the second application is parallelizable, how much speedup would this application observe if run in isolation?
- c. Given that 40% of the first application is parallelizable, how much *overall system speedup would you observe if you parallelized it?*
- d. Given that 99% of the second application is parallelizable, how much overall system speedup would you observe if you parallelized it?

Answer

- **Amdahl's law**

- a) 1/(1-0.4+0.4/2)=1.25
- b) 1/(1-0.99+0.99/2) =1.98
- c) 1/(80%* (1-0.4+0.4/2)+20%)=1.19
- d) 1/(20%* (1-0.99+0.99/2)+80%)=1.11

homework2

- 2.1 a 4-stage pipeline, the third stage consumes 2△t while the rest consumes △t .
- 1) what will happen in the pipeline if one task enters the pipeline per △t ? Blocking due to bottleneck stage
- 2) what's maximum throughput? If one task enters the pipeline per △t and 10 tasks joins the pipeline, what's the throughput and efficiency?

  TP=10/23△t, $E = \frac{25}{46} = 54.34\%$
- 3) if keep the stage time unchanged, how to improve the throughput? If one task enters the pipeline per △t and 10 tasks joins the pipeline, what's the throughput? Repeat or refine the bottleneck state S3

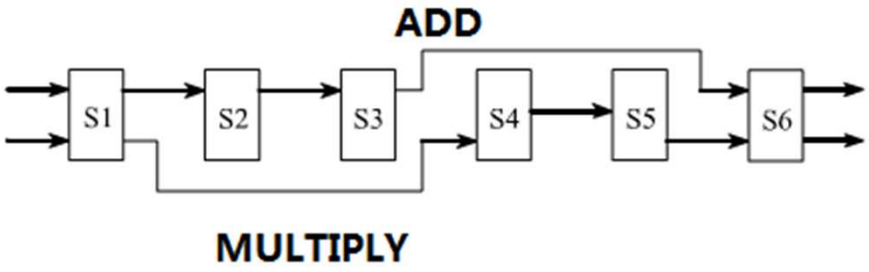
homework2

- 2.2 analyze all the possible dependences and potential hazards in the following program
- For(i=2;i<100;i++)
- a[i] += b[i];
- c[i+1]=a[i]+d[i];
- a[i-1] =2*b[i];
- b[i+1] = 2*b[i];
- Inner-loop RAW
- Outer-loop RAW, WAW, WAR

homework2

- 2.3 using a dynamic 6-stage pipeline as follows, design a fastest me) calculate the target function

$$\sum_{i=1}^{5} x_i y_i z_i$$



- 1) draw th
- 2) calculate the three metrics: TP,S,E
- TP=14/22 △t
- S= 28/11=2.54
- E=14/33 =42.4%

Homework3

- 1. For a 16 KB instruction cache with a 16 KB data cache or a 32 KB unified cache, Use 3.82,40.9,43.3 misses per 1000 instructions to help calculate the correct answer, assuming 32% of the instructions are data transfer instructions. Assume a hit takes 1 clock cycle and the miss penalty is 100 clock cycles. A load or store hit takes 2 extra clock cycle on a unified cache if there is only one cache port to satisfy two simultaneous requests. Assume write-through caches and no stalls for a write buffer.
- Q1: Which has the lower miss rate?
- Q2: What is the average memory access time in each case?

answer: A1

- **1) Formula:** Miss rate = #misses/#memory access
  = misses per 1000 instructions / ((memory accesses per instruction)*1000)

Miss rate (16K I$)=3.82/(1*(1000))=0.004

Miss rate (16K D$)=40.9/(0.32*(1000))=0.128

Miss rate(32K unified)=43.3/(1.32*1000)=0.0328

2) **Formula:** Miss rate(16K I$+16K D$) = % instruction * Miss rate (16K I$)+% data *Miss rate (16K D$)

% instruction=1/(1+0.32)=76% % data = 1-76%=24%

Miss rate(16K I$+16K D$) = 0.03376

Therefore, Unified design has a lower miss rate

answer: A2

- **Formula:** AMAT= % instruction *(HitTime+InstructionMissRate*MissPenalty)+ % dadta *(HitTime+dataMissRate*MissPenalty)
- $AMAT_{split}$=76%(1+0.004*100)+24%(1+0.128*100)=4.376 cycles
- $AMAT_{unified}$=76%(1+0.0326*100)+24%(1+2+0.326*100)=5.22 cycles
- Therefore, split design has a lower AMAT.

Homework3

- 2.Given: Assume that the cache miss penalty is 160 clock cycles, and all instructions normally take 1.0 clock cycles (ignoring memory stalls). Assume that the average miss rate is 3%, there is an average of 1.5 memory references per instruction.
- Q: What is the impact on performance when behavior of the cache is included? (speedup of the architecture with cache over without cache)

Answer

- **Formula:** CPU time=IC*(CPI$_{execution}$+MissRate * MemoryAccessesPerInstructon *MissPenalty)*ClockCycleTime
- 1) with cache

  CPU time = IC*(1+3%*1.5*160)*ClockCycleTime

  = IC*8.2*ClockCycleTime
- 2)without cache

  CPU time = IC*(1+100%*1.5*160)*ClockCycleTime

  = IC*241*ClockCycleTime

Speedup = 241/8.2=29.39

Homework3

- 3. Given: Suppose that in 1200 memory references there are 50 misses in the first-level cache and 30 misses in the second-level cache. Assume the miss penalty from the L2 cache to memory is 180 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit time of L1 is 1 clock cycle, and there are 1.4 memory references per instruction. Ignore the impact of writes.
- Q1: What are the various miss rates?
- Q2: What is the average memory access time?
- Q3: What is the average stall cycles per instruction?

Answer: A1

- Formula: Local miss rate = #misses / # cache related accesses

  Miss rate L1= 50/1200=4%

  Miss rate L2=30/50=60%

- Formula: global miss rate = #misses/#CPU related accesses

  Miss rate L1= 50/1200=4.17%

  Miss rate L2=30/1200=2.5%

Answer: A2

A2: Formula: $AMAT_{2\text{-level}}$

$= HitTime_{L1} + MissRate_{L1} *$
$(HitTime_{L2} + MissRate_{L2} * MissPenalty_{L2})$

$= 1 + 4.17\% * (10 + 60\% * 180)$

$= 5.92$ cycles

Answer: A2 & A3

Formula: AverageStallCyclesPerInstruction

$=MissPerInstruction_{L1}*HitTime_{L2}+MissPerInstruction_{L2}*MissPenalty_{L2}$

$=MissRate_{L1}*MemoryAccessesPerInstruction*HitTime_{L2}+MissRate_{L2}*MemoryAccessesPerInstruction*MissPenalty_{L2}$

=(4.17%*1.4)*10+(2.5%*1.4)*180=6.88 CyclesPerInstruction