2GA3 Tutorial #1

DATE: September 17th, 2021
TA: Jatin Chowdhary
TOPIC: Introduction

Quick Note

- These slides were made after tutorial
- If you notice anything unusual or wrong, please let me know
- If something is unclear, please do not hesitate to inform me
- If you have a suggestion, I would love to hear it!

Tutorial Question #1

- **Question:** Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.
  - a. Which processor has the highest performance expressed in instructions per second?
  - b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
  - c. We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

Tutorial Precursor #1

- **Precursor:**
  - There are 3 different processors: P1, P2, & P3
    - The instruction set across all 3 processors is the same
  - P1 has a clock rate of 3.0 GHz and a CPI of 1.5
  - P2 has a clock rate of 2.5 GHz and a CPI of 1.0
  - P3 has a clock rate of 4.0 GHz and a CPI of 2.2
  - Clock rate is "speed"
    - Technically, it is the frequency at which the clock generator of a processor can generate pulses
  - CPI stands for **C**ycles **P**er **I**nstruction (Or **C**lock **C**ycles **P**er **I**nstruction)
    - How many cycles does it take to execute an instruction?
    - More cycles = Slower performance

Tutorial Answer #1A

- **Question:** Which processor has the highest performance expressed in instructions per second?
- **Answer (1A):**
  - To determine which processor has the highest performance, we need to use the clock rate and CPI
    - The clock rate tells us how "fast" the processor is, and CPI is about efficiency. Hence, we need to divide CPI by clock rate
      - i.e. Performance (P) = Clock Rate ÷ CPI
  - *Next slide*

Tutorial Answer #1A

- **Question:** Which processor has the highest performance expressed in instructions per second?
- **Answer (1A):**
  - Performance Of P1 = 3.0 GHz ÷ 1.5 = 2.0
  - Performance Of P2 = 2.5 GHz ÷ 1.0 = 2.5
  - Performance Of P3 = 4.0 GHz ÷ 2.2 = 1.8

Tutorial Answer #1A

- **Question:** Which processor has the highest performance expressed in instructions per second?
- **Answer (1A):**
  - But wait, the textbook multiplied the clock rate by 10^9 and then calculated the performance.
    - If you want to do that, you can. But it won't change the outcome of the answer, which is P2 has the best performance
    - 10^9 is a constant. When you multiply 3.0 GHz with 10^9, you are converting it into MHz
      - GHz = Gigahertz
      - MHz = Megahertz
  - Note: Whether it is GHz or MHz, both are ways to express the "speed" of a processor. Likewise, we use kilometers per hour and meters per second to express the speed of a moving object

Tutorial Answer #1A

- **Question:** Which processor has the highest performance expressed in instructions per second?
- **Answer (1A):**
  - If you wanted to convert the clock rate from GHz to MHz, and then perform the calculation, it would look something like this:
    - Performance Of P1 = $(3.0 \times 10^9) \div 1.5 = 2.0 \times 10^9$
    - Performance Of P2 = $(2.5 \times 10^9) \div 1.0 = 2.5 \times 10^9$
    - Performance Of P3 = $(4.0 \times 10^9) \div 2.2 = 1.8 \times 10^9$
  - As you can see, the answer remains the same, regardless of whether you convert from GHz to MHz, because P2 is still the better processor with respect to performance
    - The only difference is: *($N \times 10^9$)*

Tutorial Answer #1A

- **Question:** Which processor has the highest performance expressed in instructions per second?
- **Answer (1A):**
  - “Should I convert from GHz to MHz?”
    - If the subsequent question requires you to use the value from before, then yes. Otherwise, you do not need to. This is somewhat a shortcut. But for the next question you will see a different case.
  - If you do convert from GHz to MHz, then you can say that the answer is in *Instructions Per Second*
  - On a multiple choice test, it is helpful to skip the conversion, because it saves precious time

Tutorial Answer #1B

- **Question:** If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
- **Answer (1B):**
  - To find the number of cycles in 10 seconds, multiply the clock rate (in MHz) by 10 seconds
    - Here, you need to convert the clock rate from GHz to MHz, because 1 GHz is 1 billion cycles per second. Hence you multiply by 10^9.
  - To find the number of instructions, take your answer from before and divide it by CPI
    - Once you know how many cycles are executed in 10 seconds, you need to divide it by CPI (clock cycles per instruction), to get how many instructions are executed in 10 seconds
      - Remember: CPI is about efficiency. Some processors do not execute one instruction every cycle; it may take more than 1 clock cycle to complete an instruction

Tutorial Answer #1B

- **Question:** If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
- **Answer (1B):**
  - Number Of Cycles For P1 = $(3.0 \times 10^9) \times 10 = \underline{30 \times 10^9} = 3.0 \times 10^{10}$
  - Number Of Cycles For P2 = $(2.5 \times 10^9) \times 10 = \underline{25 \times 10^9} = 2.5 \times 10^{10}$
  - Number Of Cycles For P3 = $(4.0 \times 10^9) \times 10 = \underline{40 \times 10^9} = 4.0 \times 10^{10}$
  - Number Of Instructions For P1 = $(\underline{30 \times 10^9}) \div 1.5 = 20 \times 10^9 = 2.0 \times 10^{10}$
  - Number Of Instructions For P2 = $(\underline{25 \times 10^9}) \div 1.0 = 25 \times 10^9 = 2.5 \times 10^{10}$
  - Number Of Instructions For P3 = $(\underline{40 \times 10^9}) \div 2.2 = 18.18 \times 10^9 = 1.818 \times 10^{10}$

Tutorial Question #1C

- **Question:** We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?
- **Answer (1C):**
  - The question does not say this but 1C is an extension of 1B
    - The execution time in 1B is 10 seconds, and we want to reduce it by 30%. So, we are trying to achieve an execution time of 7 seconds.
  - In our attempt to reduce the execution time, the CPI increases by 20%
    - This can be modeled via the following equation: $CPI_{new} = CPI_{old} \times 1.2$
  - We need to find the clock rate (*f*) that will yield the time reduction (30%) we are seeking; we also need to factor in the new CPI
    - The following equation will be used: f = ((Number of Instructions) × CPI) ÷ (Execution Time)

Tutorial Question #1C

- **Question:** We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?
- **Answer (1C):**
  - First, let’s calculate the new CPI:
    - New CPI Of P1 = 1.5 × 1.2 = 1.80
    - New CPI Of P2 = 1.0 × 1.2 = 1.20
    - New CPI Of P3 = 2.2 × 1.2 = 2.64
  - Next, we use the formula for clock rate to calculate it:
    - f = ((Number of Instructions) × CPI) ÷ (Execution Time)
      - *Next slide*

Tutorial Question #1C

- **Question:** We are trying to reduce the execution time by 30%, but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?
- **Answer (1C):**
  - Calculate the clock rate (f) of each processor:
    - Clock Rate of P1 = $(20 \times 10^9 \times 1.8) \div 7 = 5142857142.85$ MHz
      - 5142857142.85 MHz = 5.14 GHz
    - Clock Rate of P2 = $(25 \times 10^9 \times 1.2) \div 7 = 4285714285.71$ MHz
      - 4285714285.71 MHz = 4.28 GHz
    - Clock Rate of P3 = $(18.18 \times 10^9 \times 2.6) \div 7 = 6752571428.57$ MHz
      - 6752571428.57 MHz = 6.75 GHz

Tutorial Question #2

- **Question:** Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.
  - a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
  - b. Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
  - c. A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?

Tutorial Precursor #2

- **Precursor:**
  - Dealing with 2 compilers:
    - Compiler A
    - Compiler B
  - Compiler A has a dynamic instruction count of 1.0×10^9
    - And an execution time of 1.1 seconds
  - Compiler B has a dynamic instruction count of 1.2×10^9
    - And an execution time of 1.5 seconds

Tutorial Answer #2A

- **Question:** Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
- **Answer(A):**
  - We need to calculate the CPIs of a program that is compiled with different compilers, A and B
    - This results in 2 *different* programs
      - Note: The programs are not inherently different
  - We will use 2 formulas:
    - Clock Cycle Time = 1 ÷ Clock Rate
    - CPI = Execution Time × ((Clock Rate) ÷ (Number Of Instructions))
  - *Next slide*

Tutorial Answer #2A

- **Question:** Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
- **Answer(A):**
  - The first thing we need to do is calculate the Clock Rate from the Clock Cycle Time
    - We can rearrange the following formula:

      Clock Cycle Time = 1 ÷ Clock Rate

      (rearrange)

      Clock Rate = 1 ÷ Clock Cycle Time
  - *Next slide*

Tutorial Answer #2A

- **Question:** Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
- **Answer(A):**
  - The Clock Rate, in GHz, is:
    - Clock Rate (GHz) = 1 ÷ Clock Cycle Time
    - Clock Rate (GHz) = 1 ÷ 1 *ns*
    - Clock Rate = 1 GHz = 10^9 MHz

Tutorial Answer #2A

- **Question:** Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
- **Answer(A):**
  - Now that we have the Clock Rate of the processor, we can calculate the CPI of each program when compiled using a different compiler
  - First, we calculate the CPI of the program when compiled via *Compiler_A*
    - $CPI_{Compiler\_A}$ = Execution Time × ((Clock Rate) ÷ (Number Of Instructions))
    - $CPI_{Compiler\_A} = 1.1 \times (10^9 \div (1 \times 10^9))$
    - $CPI_{Compiler\_A} = 1.1 \times (10^9 \div 10^9)$
    - $CPI_{Compiler\_A} = 1.1 \times 1$
    - $CPI_{Compiler\_A} = 1.1$

Tutorial Answer #2A

- **Question:** Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
- **Answer(A):**
  - Then, we calculate the CPI of the program when compiled via *Compiler_B*
    - $CPI_{Compiler\_B}$ = Execution Time × ((Clock Rate) ÷ (Number Of Instructions))
    - $CPI_{Compiler\_B} = 1.5 \times (10^9 \div (1.2 \times 10^9))$
    - $CPI_{Compiler\_B} = 1.5 \times (0.83333333333)$
    - $CPI_{Compiler\_B} = 1.25$
  - Our final answer is:
    - $CPI_{Compiler\_A} = 1.1$
    - $CPI_{Compiler\_B} = 1.25$

Tutorial Question #2B

- **Question:** Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
- **Answer (B):**
  - The question does not state this, but we will use the values from part (A) to calculate the speed up. Plus, we will use the equation from part (A), it is:
    - CPI = Execution Time × ((Clock Rate) ÷ (Number Of Instructions))
  - However, we need to rearrange the equation above and solve for Clock Rate. This is because we need to find out how much faster is one compiler over the other compiler, assuming execution time is the same across both processors.
    - Hence, we arrange the equation above for Clock Rate because this is what we are solving for. We already have CPI and Number of Instructions, and Execution Time is the same.
      - Next slide

Tutorial Question #2B

- **Question:** Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
- **Answer (B):**
  - Rearranging the formula for Clock Rate is as follows:
    - CPI = Execution Time × ((Clock Rate) ÷ (Number Of Instructions))
    - ((Clock Rate) ÷ (Number Of Instructions)) = (CPI ÷ (Execution Time))
    - (Clock Rate) = (CPI ÷ (Execution Time)) × (Number Of Instructions)
    - (Clock Rate) = (CPI × (Number Of Instructions)) ÷ (Execution Time)
      - This is our new equation; we will use this to solve the question

Tutorial Question #2B

- **Question:** Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
- **Answer (B):**
  - Since we need to find how much faster is one processor (P1) over another (P2), we divide the equations:

$$\frac{(CPI \times (\text{Number Of Instructions})) \div (\text{Execution Time})}{(CPI \times (\text{Number Of Instructions})) \div (\text{Execution Time})}$$

  - But wait, since we know that the execution time is the same, we can cancel it right now:

$$\frac{(CPI \times (\text{Number Of Instructions}))}{(CPI \times (\text{Number Of Instructions}))}$$

  - *Next slide*

Tutorial Question #2B

- **Question:** Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
- **Answer (B):**
  - Now we substitute values for P1 & P2 and solve!

    $$\frac{(1.1 \times (1.0 \times 10^9))}{(1.25 \times (1.2 \times 10^9))}$$

  - After doing some simple math we get:

    $$\frac{(1.1 \times 10^9)}{(1.5 \times 10^9)}$$

  - For clarity, we can cancel the 10^9 from both numerator and denominator

    $$\frac{(1.1)}{(1.5)}$$

  - Answer?
    - *Next slide*

Tutorial Question #2B

- **Question:** Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock rate of the processor running compiler A's code versus the clock rate of the processor running compiler B's code?
- **Answer (B):**
  - The answer is: **0.7333333333**
    - What does this mean?
      - This means that P1 is roughly 0.73 times slower than P2
    - If we take the inverse of 0.73, we get 1.3636363636
      - This means that P2 is roughly 1.36 times faster than P1

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - We will use the equation from 1B to compare the speedup, but we need to rearrange it for execution time.
    - This is because all 3 compilers are used on the original processor, and we need to find the speed up, which is the difference in execution time
  - The steps are:
    - i. Rearrange the equation from 1B to solve for ExecutionTime
    - ii. Compare the speedup between CompilerA and CompilerNew
    - iii. Compare the speedup between CompilerB and CompilerNew

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - The equation from 1B is:
    - (Clock Rate) = (CPI × (Number Of Instructions)) ÷ (Execution Time)
  - We rearrange it to solve for Execution Time:
    - (Clock Rate) = (CPI × (Number Of Instructions)) ÷ (Execution Time)
    - (Execution Time) = (CPI × (Number Of Instructions)) ÷ (Clock Rate)
  - But wait, since all 3 compilers are used on the same processor, we can eliminate Clock Rate, and simplify the equation:
    - (Execution Time) = (CPI × (Number Of Instructions))

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - Now, we calculate the speedup for Compiler**New** VS. Compiler**A**:
    - In essence, we are dividing the two values because we are comparing the Execution Time
  - Setting up the equation:

$$\frac{(\text{CPI} \times (\text{Number Of Instructions}))}{(\text{CPI} \times (\text{Number Of Instructions}))}$$

  - Substituting the values:

$$\frac{(1.1 \times (6.0 \times 10^8))}{(1.1 \times (1.0 \times 10^9))}$$

  - Performing arithmetic:

$$\frac{(6.6 \times 10^8)}{(1.1 \times 10^9)}$$

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - After dividing the fraction, the answer is: **0.6**
    - This means that CompilerA is 0.6x slower than CompilerNew
  - If we take the inverse of the answer, we get: **1.666666**
    - This means that CompilerNew is 1.66 times faster than CompilerA

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - Finally, we calculate the speedup for Compiler**New** VS. Compiler**B**, the math is identical from before:
    - In essence, we are dividing the two values because we are comparing the Execution Time
  - Setting up the equation:

$$\frac{(\text{CPI} \times (\text{Number Of Instructions}))}{(\text{CPI} \times (\text{Number Of Instructions}))}$$

  - Substituting the values:

$$\frac{(1.1 \times (6.0 \times 10^{8}))}{(1.25 \times (1.2 \times 10^{9}))}$$

  - Performing arithmetic:

$$\frac{(6.6 \times 10^{8})}{(1.5 \times 10^{9})}$$

Tutorial Question #2C

- **Question:** A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
- **Answer (C):**
  - After dividing the fraction, the answer is: **0.44**
    - This means that CompilerB is 0.44x slower than CompilerNew
  - If we take the inverse of the answer, we get: **2.2727272**
    - This means that CompilerNew is 2.27 times faster than CompilerB

Tutorial Question #3

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.1] Find the CPI if the clock cycle time is 0.333 ns.
  - [1.12.2] Find the SPEC ratio.
  - [1.12.3] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.
  - [1.12.4] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.
  - [1.12.5] Find the change in the SPEC ratio for this change.
  - [1.12.6] Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPECratio is 13.7. Find the new CPI.

Tutorial Question #3

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.7] This CPI value is larger than obtained in 1.12.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?
  - [1.12.8] By how much has the CPU time been reduced?
  - [1.12.9] For a second benchmark, libquantum, assume an execution time of 960 s, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.
  - [1.12.10] Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.

Tutorial Precursor #3

- **Precursor:**
  - *SPEC* stands for Standard Performance Evaluation Corporation
    - The job of the SPEC is to create a standardized set of performance benchmarks for computers.
  - *bzip2* is a file compression program
    - Similar to WinRAR, 7zip, etc.
  - Benchmarking is used to measure performance
    - i.e. How fast something is?
    - We use benchmarking to compare things with each other
      - Applies to all facets of life, not just computers
  - AMD Barcelona is a processor
    - The processor/CPU is the brain of the computer
      - The CPU is where computation/arithmetic occurs

Tutorial Precursor #3

- **Precursor:**
  - The SPEC benchmark results for running bzip2 (a program) on the AMD Barcelona (the processor) are:
    - Instruction Count = 2.389×10^{12}
    - Execution Time = 750 s
    - Reference Time = 9650 s

Tutorial Question #3 [1.12.1]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.1] Find the CPI if the clock cycle time is 0.333 ns.
- **Answer:**
  - The CPI can be calculated via:
    - CPI = Clock Rate × (CPU Time ÷ Instruction Count)
  - The only thing we are missing is the Clock Rate, but we can calculate it via:
    - Clock Rate = 1 ÷ Cycle Time
      - The answer will be in GHz
  - *Next slide*

Tutorial Question #3 [1.12.1]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.1] Find the CPI if the clock cycle time is 0.333 ns.
- **Answer:**
  - Clock Rate calculation:
    - Clock Rate = (1 ÷ Cycle Time)
    - Clock Rate = (1 ÷ 0.333333333)
    - Clock Rate = 3 GHz
    - **Clock Rate = 3 × 10^9 MHz**
  - CPI Calculation:
    - *Next slide*

Tutorial Question #3 [1.12.1]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.1] Find the CPI if the clock cycle time is 0.333 ns.
- **Answer:**
  - CPI Calculation:
    - CPI = Clock Rate × (CPU Time ÷ Instruction Count)
    - $CPI = (3 \times 10^9) \times (750 \div (2.389 \times 10^{12}))$
    - $CPI = (3 \times 10^9) \times (3.139388866 \times 10^{-10})$
    - **CPI = 0.941816659**
  - The CPI for bzip2, if the clock cycle time is 0.333 ns is: **0.941816659**

Tutorial Question #3 [1.12.2]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.2] Find the SPEC ratio.
- **Answer:**
  - The SPEC ratio can be calculated via the following formula:
    - $SPEC_{ratio}$ = (Reference Time / Execution Time)
    - $SPEC_{ratio}$ = (9650s / 750s)
    - **$SPEC_{ratio}$ = 12.866666666**
  - The SPEC ratio for *bzip2* is (roughly): **~12.86**

Tutorial Question #3 [1.12.3]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.3] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.
- **Answer:**
  - The answer is quite trivial. If the CPI and Clock Rate do not change, then the increase in CPU time is proportional to the increase in the number of instructions, which is 10%
    - Therefore, the CPU time increases by 10%
      - Note: CPU time and Execution time are the same
  - Just for fun, let's do the math
    - Next slide

Tutorial Question #3 [1.12.3]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.3] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.
- **Answer:**
  - First of all, we take the equation from [1.12.1] and rearrange it for CPU Time:
    - CPI = Clock Rate × (CPU Time ÷ Instruction Count)
    - (CPI ÷ Clock Rate) = (CPU Time ÷ Instruction Count)
    - (CPU Time) = ((CPI ÷ Clock Rate) ÷ Instruction Count)
    - (CPU Time) = ((CPI × Instruction Count) ÷ Clock Rate)
  - Now, we take this equation and solve for CPU Time
    - Next slide

Tutorial Question #3 [1.12.3]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.3] Find the increase in CPU time if the number of instructions of the benchmark is increased by **10%** without affecting the CPI.
- **Answer:**
  - Now, we take the equation and solve for CPU/Execution Time:
    - (CPU Time) = ((CPI × Instruction Count) ÷ Clock Rate)
    - (CPU Time) = ((0.941816659 × (1.1 × (2.389 × 10^{12}))) ÷ (3 × 10^{9}))
      - Note: The CPU time is taken from [1.12.1]
    - (CPU Time) = (2.474999998 × 10^{12}) ÷ (3 × 10^{9})
    - **(CPU Time) = 824.99999993 s**

Tutorial Question #3 [1.12.3]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.3] Find the increase in CPU time if the number of instructions of the benchmark is increased by **10%** without affecting the CPI.
- **Answer:**
  - The new CPU time is: **824.99999993 s**
  - If we multiply 750 s by 1.1, we get:
    - $750 \times 1.1 =$ **825 s**
  - Therefore, if the number of instructions of the benchmark is increased by 10%, then the CPU time will also increase by 10%, assuming that CPI is unaffected.
    - Note: Clock Rate needs to remain the same as well

Tutorial Question #3 [1.12.4]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.4] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.
- **Answer:**
  - The logic for this question [1.12.4] is the same as the previous question [1.12.3]
  - We need to use the following formula:
    - (CPU Time) = ((CPI × Instruction Count) ÷ Clock Rate)
  - We know that the number of instructions increases by 10% and the CPI is increased by 5%. The formula can be adjusted as follows:
    - (CPU Time) = (((1.05 × CPI) × (1.1 × Instruction Count)) ÷ Clock Rate)

Tutorial Question #3 [1.12.4]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.4] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.
- **Answer:**
  - Since multiplication is associative, we can skip doing the math and conclude the following:
    - CPU Time increases by a factor of: (1.05 × 1.1) **= 1.155**
  - We can show that the above statement holds true by doing the math; although it is not necessary
    - (CPU Time) = (((1.05 × CPI) × (1.1 × Instruction Count)) ÷ Clock Rate)
    - (CPU Time) = (((1.05 × 0.941816659) × (1.1 × (2.389 × 10^{12}))) ÷ (3 × 10^{9}))
      - Note: The CPI and Clock Rate are from [1.12.1]
    - (CPU Time) = ((0.988907492 × (2.6279 × 10^{12})) ÷ (3 × 10^{9}))
    - (CPU Time) = ((2.59875 × 10^{12}) ÷ (3 × 10^{9}))
    - **(CPU Time) = 866.2499994 s**

Tutorial Question #3 [1.12.4]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.4] Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%.
- **Answer:**
  - If we multiply 750 with 1.155 we get:
    - $750 \text{ s} \times 1.155 = 866.25 \text{ s}$
  - Therefore, we can conclude that if the number of instructions of the benchmark is increased by 10%, and the CPI is increasd by 5%, then the **CPU time (or execution time) increases by a factor of 1.155**

Tutorial Question #3 [1.12.5]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.5] Find the change in the SPEC ratio for this change.
- **Answer:**
  - From [1.12.2] we know that the equation for $SPEC_{ratio}$ is:
    - $SPEC_{ratio}$ = (Reference Time ÷ Execution Time)
  - To find the change in $SPEC_{ratio}$, we need to do:
    - (New-$SPEC_{ratio}$) ÷ (Old-$SPEC_{ratio}$)
  - After substitution we get:
    - $(\text{Reference Time} \div \text{Execution Time})_{NEW} \div (\text{Reference Time} \div \text{Execution Time})_{OLD}$
  - Since the Reference Time is the same, we can cancel it out:
    - $(\text{Execution Time})_{NEW} \div (\text{Execution Time})_{OLD}$
  - *Next slide*

Tutorial Question #3 [1.12.5]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.5] Find the change in the SPEC ratio for this change.
- **Answer:**
  - From [1.12.4] we know that the *New Execution Time* is: 866.25 s
  - From the question, we know that the *Old Execution Time* is: 750 s
  - Calculation
    - $(866.25)_{NEW} \div (750)_{OLD}$ **= 1.155**
  - The *new time* is 1.155x slower than the *old time*
    - If we take the inverse of 1.155, we get: 0.865800865 (~0.86)
      - 100 – 86 = 14%
  - Therefore, we can conclude that the $SPEC_{ratio}$ decreasd by 14%.

Tutorial Question #3 [1.12.6]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.6] Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPECratio is 13.7. Find the new CPI.
- **Answer:**
  - To calculate the CPI, we take the equation from [1.12.1] and plug the numbers in. The equation is:
    - CPI = ((Clock Rate × CPU Time) ÷ Instruction Count)
  - The numbers are:
    - Clock Rate = 4 GHz **= 4×10^9 MHz**
    - CPU Time = 700 s
      - Note: CPU Time is the same as Execution Time
    - Instruction Count = $(0.85 \times 2.389 \times 10^{12})$ **= (2.03065×10^{12})**
  - *Next slide*

Tutorial Question #3 [1.12.6]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.6] Suppose that we are developing a new version of the AMD Barcelona processor with a 4 GHz clock rate. We have added some additional instructions to the instruction set in such a way that the number of instructions has been reduced by 15%. The execution time is reduced to 700 s and the new SPECratio is 13.7. Find the new CPI.
- **Answer:**
  - We take the equation from before, and we plug in the values:
    - CPI = ((Clock Rate × CPU Time) ÷ Instruction Count)
    - CPI = $(((4 \times 10^{9}) \times 700) \div (2.03065 \times 10^{12}))$
    - CPI = $((2.8 \times 10^{12}) \div (2.03065 \times 10^{12}))$
    - **CPI = 1.378868835**
  - Therefore, the CPI of the new version of the AMD Barcelona processor is: **1.378868835**

Tutorial Question #3 [1.12.7]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.7] This CPI value is larger than obtained in 1.12.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?
- **Answer:**
  - For this question, we are comparing the ratio of Clock Rate and CPI to see if they are the same or not
  - Values from [1.12.1]
    - Instruction Count = 2.389×10^9
    - Clock Rate = 3.0 GHz
    - CPI = 0.94
  - Values From [1.12.6]
    - Instruction Count = 2.03065×10^9
    - Clock Rate = 4.0 GHz
    - CPI = 1.37
  - *Next slide*

Tutorial Question #3 [1.12.7]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.7] This CPI value is larger than obtained in 1.12.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?
- **Answer:**
  - Divide Clock Rates:
    - $(\text{Clock Rate})_{New} \div (\text{Clock Rate})_{Old}$
    - $(4.0)_{New} \div (3.0)_{Old}$
    - 1.3333333333
  - Divide CPIs:
    - $(\text{CPI})_{New} \div (\text{CPI})_{Old}$
    - $(1.37)_{New} \div (0.94)_{Old}$
    - 1.457446809
  - *Next slide*

Tutorial Question #3 [1.12.7]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.7] This CPI value is larger than obtained in 1.12.1 as the clock rate was increased from 3 GHz to 4 GHz. Determine whether the increase in the CPI is similar to that of the clock rate. If they are dissimilar, why?
- **Answer:**
  - When comparing the ratios, the increase in CPI is different from the Clock Rate because both instruction count and CPU time (aka execution time) have been reduced.
    - CPU time (aka execution time) is reduced by a lower percentage than instruction count

Tutorial Question #3 [1.12.8]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.8] By how much has the CPU time been reduced?
- **Answer:**
  - The old CPU time is: 750 s
  - The new CPU time is: 700 s
  - If we divide the new CPU time by the old CPU time, we get:
    - Ratio = (700) ÷ (750)
    - Ratio = (0.9333333)
  - We can conclude that the CPU time reduction is: **~6.7%**
    - 100 – 93.3333333 **= 6.66666667%**

Tutorial Question #3 [1.12.9]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.9] For a second benchmark, libquantum, assume an execution time of 960 s, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.
- **Answer:**
  - We take the equation we've been using all along, and rearrange it to solve for *instruction count*:
    - CPI = (Clock Rate) × (CPU Time ÷ Instruction Count)
    - CPI = (Clock Rate) × (CPU Time ÷ Instruction Count)
    - (CPI ÷ Clock Rate) = (CPU Time ÷ Instruction Count)
    - (Clock Rate ÷ CPI) = (Instruction Count ÷ CPU Time)
      - Note: In this step, we flipped the numerator and denominator on both sides
    - (Instruction Count ÷ CPU Time) = (Clock Rate ÷ CPI)
    - **(Instruction Count) = ((CPU Time × Clock Rate) ÷ CPI)**

Tutorial Question #3 [1.12.9]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.9] For a second benchmark, libquantum, assume an execution time of 960 s, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.
- **Answer:**
  - With our rearranged equation, we can calculate the new instruction count:
    - The values we will use are:
      - CPU Time = 0.9 × 960 **= 864**
      - Clock Rate = 4 GHz **= 4.0×10^9 MHz**
      - CPI = 1.61
  - Calculation:
    - We will use the following formula:
      - (Instruction Count) = ((CPU Time × Clock Rate) ÷ CPI)
    - *Next slide*

Tutorial Question #3 [1.12.9]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.9] For a second benchmark, libquantum, assume an execution time of 960 s, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.
- **Answer:**
  - Calculation:
    - (Instruction Count) = ((CPU Time × Clock Rate) ÷ CPI)
    - (Instruction Count) = $((864 \times (4.0 \times 10^9)) \div 1.61)$
    - (Instruction Count) = $((3.456 \times 10^{12}) \div 1.61)$
    - **(Instruction Count) = (2.1465×10^{12})**
  - Therefore, the instruction count is: **2.1465×10^{12}**

Tutorial Question #3 [1.12.10]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.10] Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.
- **Answer:**
  - We take the formula we've been using all along, and rearrage it to solve for Clock Rate:
    - CPI = (Clock Rate) × (CPU Time ÷ Instruction Count)
    - (CPI ÷ Clock Rate) = (CPU Time ÷ Instruction Count)
    - (Clock Rate ÷ CPI) = (Instruction Count ÷ CPU Time)
    - **(Clock Rate) = ((Instruction Count ÷ CPU Time) × CPI)**
  - Calculation
    - *Next slide*

Tutorial Question #3 [1.12.10]

- **Question:** The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389E12, an execution time of 750 s, and a reference time of 9650 s.
  - [1.12.10] Determine the clock rate required to give a further 10% reduction in CPU time while maintaining the number of instructions and with the CPI unchanged.
- **Answer:**
  - Calculation:
    - (Clock Rate) = ((Instruction Count ÷ CPU Time) × CPI)
    - (Clock Rate) = $((2.1465 \times 10^{12} \div (0.9 \times 960)) \times 1.61)$
    - (Clock Rate) = $((2.1465 \times 10^{12} \div (864)) \times 1.61)$
    - (Clock Rate) = $((2.484375 \times 10^{9}) \times 1.61)$
    - (Clock Rate) = (3.999843×10^{9}) MHz
    - **(Clock Rate) ~ (4.0 GHz)**
  - Therefore, we need a clock rate of about 4 GHz to achieve a 10% reduction in CPU time (or execution time), without affecting the number of instructions or CPI

THE END