EXAMPLE PROBLEMS

A program is run on a 8 GHz processor. The program has 20 million instructions. 50% of the instructions are integer arithmetic instructions that are 4 CPI. 30% are memory load or store instructions that are 5 CPI. 10% are floating point that are 6 CPI. 10% are branch instructions that are 3 CPI.

What is the average CPI?

Avg CPI = 0.5*4 + 0.3*5 + 0.1*6 + 0.1*3 = 4.4 cycles/instruction

A program is run on a 8 GHz processor. The program has 20 million instructions. 50% of the instructions are integer arithmetic instructions that are 4 CPI. 30% are memory load or store instructions that are 5 CPI. 10% are floating point that are 6 CPI. 10% are branch instructions that are 3 CPI.

How long does it take the program to run?

sec/program = 20e6 instructions/program*4.4 cycles/instruction/(8e9 cycles/sec)
= 0.011 sec/program

A program is run on a 8 GHz processor. The program has 20 million instructions. 50% of the instructions are integer arithmetic instructions that are 4 CPI. 30% are memory load or store instructions that are 5 CPI. 10% are floating point that are 6 CPI. 10% are branch instructions that are 3 CPI.

What is the speedup if integer arithmetic instructions are improved to 3 CPI?

Speedup = old time/new time = $(\text{cycle time*instruction count*CPI})_{OLD}$ / $(\text{cycle time*instruction count*CPI})_{NEW}$

$= (\text{CPI})_{OLD} / (\text{CPI})_{NEW}$

$= 4.4 / (0.5*3 + 0.3*5 + 0.1*6 + 0.1*3)$

$= 4.4 / 3.9$

$= 4.4 / 3.9 = 1.13$

A program is run on a 8 GHz processor. The program has 20 million instructions. 40% of the instructions are integer arithmetic instructions that are 4 CPI. 40% are memory load or store instructions that are 5 CPI. 20% are branch instructions that are 3 CPI.

The program is rewritten so there are only half as many branch instructions. (So the number of instructions has been reduced). What is the speedup?

A program is run on a 8 GHz processor. The program has 20 million instructions. 40% of the instructions are integer arithmetic instructions that are 4 CPI. 40% are memory load or store instructions that are 5 CPI. 20% are branch instructions that are 3 CPI.

The program is rewritten so there are only half as many branch instructions. (So the number of instructions has been reduced). What is the speedup?

Speedup = old time/new time = $(\text{cycles/program})_{OLD}$ / $(\text{cycles/program})_{NEW}$

$(\text{cycles/program})_{OLD}$ = 20e6*4*0.4 + 20e6*5*0.4 + 20e6*3*0.2 = 32e6 + 40e6 + 12e6 = 84e6 cycles/program

$(\text{cycles/program})_{NEW}$ = 32e6 + 40e6 + 6e6 = 78e6 cycles/program

Speedup = 84/78 = 1.08

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 of the processor running compiler A's code versus the clock 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?

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 of the processor running compiler A's code versus the clock 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?

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.

CPI_A = 1.1 seconds/program / (1e-9 seconds/cycle * 1e9 instructions/program) = 1.1 cycles/instruction

CPI_B = 1.5 seconds/program / (1e-9 seconds/cycle * 1.2e9 instructions/program) = 1.25 cycles/instruction

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 of the processor running compiler A's code versus the clock of the processor running compiler B's code?

Execution time = cycle time*instruction count*CPI

$(\text{instruction count*CPI})_B$ / $(\text{instruction count*CPI})_A$ = $(\text{cycle time})_A$ /$(\text{cycle time})_B$ = $(\text{clock rate})_B$ /$(\text{clock rate})_A$

1.2e9 * 1.25 / 1e9 * 1.1 = 1.36

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 of the processor running compiler A's code versus the clock 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?

$\text{Speedup} = \text{old time/new time} = (\text{instruction count*CPI})_{OLD} / (\text{instruction count*CPI})_{NEW}$

$\text{Speedup}_A = (1e9*1.1)_{OLD} / (6e8*1.1)_{NEW} = 1.67$

$\text{Speedup}_B = (1.2e9*1.25)_{OLD} / (6e8*1.1)_{NEW} = 2.27$

A program containing twenty million instructions is executed on a system with a fixed clock cycle time of 2 nano-seconds. The divide instruction on this machine requires 8 clock cycles and accounts for 20% of the total number of instructions executed in the program. The other 80% of the instructions in the program require an average of 4 clock cycles per instruction.

What is the average CPI?

A program containing twenty million instructions is executed on a system with a fixed clock cycle time of 2 nano-seconds. The divide instruction on this machine requires 8 clock cycles and accounts for 20% of the total number of instructions executed in the program. The other 80% of the instructions in the program require an average of 4 clock cycles per instruction.

What speedup would be obtained for this program by making the divide instructions twice as fast?

A program containing twenty million instructions is executed on a system with a fixed clock cycle time of 2 nano-seconds. The divide instruction on this machine requires 8 clock cycles and accounts for 20% of the total number of instructions executed in the program. The other 80% of the instructions in the program require an average of 4 clock cycles per instruction.

What speedup would be obtained for this program by making the divide instructions twice as fast?

Speedup = old time / new time

= $(IC*CPI*cycletime)_{old}/(IC*CPI*cycletime)_{new}$

The problem states that the divide is twice as fast. IC didn't change. Cycletime didn't change, or all the instructions would be twice as fast.

So CPI changes to make divide twice as fast. New CPI for divide = 8/2 = 4

So average CPI_{new} = 4

CPI_{old} = 4.8

Speedup = CPI_{old} / CPI_{new} = 4.8/4 = 1.2

A program containing twenty million instructions is executed on a system with a fixed clock cycle time of 2 nano-seconds. The divide instruction on this machine requires 8 clock cycles and accounts for 20% of the total number of instructions executed in the program. The other 80% of the instructions in the program require an average of 4 clock cycles per instruction.

What is the average CPI?

8*0.2 + 4*0.8 = 1.6+3.2 = 4.8 cycles/instruction