Jose Franco Baquera

February 7, 2019

CS 473

CS 473 – Architectural Concepts I: Assignment #1

1. The eight great ideas in computer architecture are similar to ideas from other fields. Match the eight ideas from computer architecture, "Design for Moore's Law", "Use Abstraction to Simplify Design", "Make the Common Case Fast", "Performance via Parallelism", "Performance via Pipelining", "Performance via Prediction", "Hierarchy of Memories", and "Dependability via Redundancy" to the following ideas from other fields:

   a. Assembly lines in automobile manufacturing – ***Performance via Pipelining***

   b. Suspension bridge cables – ***Performance via Parallelism***

   c. Aircraft and marine navigation systems that incorporate wind information –***Performance via Prediction***

   d. Express elevators in buildings – ***Make the Common Case Fast***

   e. Library reserve desk – ***Hierarchy of Memories***

   f. Increasing the gate area on a CMOS transistor to decrease its switching time – ***Design for Moore's Law***

   g. Adding electromagnetic aircraft catapults (which are electrically-powered as opposed to current steampowered models), allowed by the increased power generation offered by the new reactor technology – ***Dependability via Redundancy***

   h. Building self-driving cars whose control systems partially rely on existing sensor systems already installed into the base vehicle, such as lane departure systems and smart cruise control systems – ***Use Abstraction to Simplify Design***

2. 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?

      We note the following equation:

$$Performance = \frac{1}{Execution\ Time} = \frac{1}{\frac{Instruction\ Count * CPI}{Clock\ Rate}} = \frac{Clock\ Rate}{Instruction\ Count * CPI}$$

However, since all three processors are executing the same instruction set, we ignore the variable *instruction count.* Therefore, the performance of each processor expressed in instructions per second is the following:

- $Performance(P1) = \frac{\frac{3*10^9 cycles}{second}}{\frac{1.5\ cycles}{instruction}} = \frac{3*10^9 \cancel{cycles}}{second} * \frac{instruction}{1.5\ \cancel{cycles}} = \mathbf{2 * 10^9\ \frac{instructions}{second}}$
- $Performance(P2) = \frac{\frac{2.5*10^9 cycles}{second}}{\frac{1.0\ cycles}{instruction}} = \frac{2.5*10^9 \cancel{cycles}}{second} * \frac{instruction}{1.0\ \cancel{cycles}} = \mathbf{2.5 * 10^9\ \frac{instructions}{second}}$
- $Performance(P3) = \frac{\frac{4*10^9 cycles}{second}}{\frac{2.2\ cycles}{instruction}} = \frac{4*10^9 \cancel{cycles}}{second} * \frac{instruction}{2.2\ \cancel{cycles}} = \mathbf{1.82 * 10^9\ \frac{instructions}{second}}$

**Answer: Processor P2 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.

To find the number of cycles and the number of instructions, we use the following two equations:

1. $CPU\ Clock\ Cycles\ For\ a\ Program = Instruction\ Count * CPI$
2. $Instruction\ Count = \frac{CPU\ Time * Clock\ Rate}{CPI}$

*(NOTE: We could have also used the equation Clock Rate * 10 seconds to find the CPU Clock Cycles for the program.)*

- Processor P1

$$Instruction\ Count = \frac{10\ secs * \frac{3*10^9 cycles}{secs}}{\frac{1.5\ cycles}{instruction}} = \frac{3*10^{10} \cancel{secs}\ \cancel{cycles}}{\cancel{secs}} * \frac{instructions}{1.5\ \cancel{cycles}} = \mathbf{2 * 10^{10}\ instructions}$$

$$Total\ Cycles = 2 * 10^{10} \cancel{instruction} * \frac{1.5\ cycles}{\cancel{instruction}} = \mathbf{3 * 10^{10}\ cycles}$$

**Answer: Processor P1 had a total of $2 * 10^{10}$ *instructions* and $3 * 10^{10}$ *cycles*.**

- Processor P2

$$Instruction\ Count = \frac{10\ secs * \frac{2.5*10^9 cycles}{secs}}{\frac{1.0\ cycles}{instruction}} = \frac{2.5*10^{10} \cancel{secs}\ \cancel{cycles}}{\cancel{secs}} * \frac{instructions}{1.0\ \cancel{cycles}} = \mathbf{2.5 * 10^{10}\ instructions}$$

$$Total\ Cycles = 2.5 * 10^{10} \cancel{instruction} * \frac{1.0\ cycles}{\cancel{instruction}} = \mathbf{2.5 * 10^{10}\ cycles}$$

**Answer: Processor P2 had a total of $2.5 * 10^{10}$ *instructions* and $2.5 * 10^{10}$ *cycles*.**

- Processor P3

$$Instruction\ Count = \frac{10\ secs * \frac{4.0*10^9 cycles}{secs}}{\frac{2.2\ cycles}{instruction}} = \frac{4.0*10^{10} \cancel{secs}\ \cancel{cycles}}{\cancel{secs}} * \frac{instructions}{2.2\ \cancel{cycles}} = \mathbf{1.82 * 10^{10}\ instructions}$$

$$Total\ Cycles = 1.8181818181 * 10^{10} \cancel{instruction} * \frac{2.2\ cycles}{\cancel{instruction}} = \mathbf{4 * 10^{10}\ cycles}$$

**Answer: Processor P3 had a total of $1.82 * 10^{10}$ *instructions* and $4 * 10^{10}$ *cycles*.**

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?

To find the new clock rate, we use the following equation:

$$(CPU\ Time * 0.70) = \frac{Instruction\ Count * (1.20 * CPI)}{Clock\ Rate}$$

If we simplify this equation, we get the following:

$$New\ Clock\ Rate = \frac{Instruction\ Count * (1.20 * CPI)}{(CPU\ Time * 0.70)}$$

- Processor P1

$$New\ Clock\ Rate = \frac{Instruction\ Count * (1.20 * CPI)}{(CPU\ Time * 0.70)}$$

$$= \frac{2*10^{10}\cancel{instructions}*(1.20*\frac{1.5\ cycles}{\cancel{instruction}})}{10\ secs*0.70}$$

$$= 5142857143\frac{cycles}{secs} = 5.142857143 * 10^9\frac{cycles}{secs}$$

$$= \mathbf{5.14\ GHz}$$

**Answer: For processor P1, we need a new clock rate of 5.14 GHz.**

- Processor P2

$$New\ Clock\ Rate = \frac{Instruction\ Count * (1.20 * CPI)}{(CPU\ Time * 0.70)}$$

$$= \frac{2.5*10^{10}\cancel{instructions}*(1.20*\frac{1.0\ cycles}{\cancel{instruction}})}{10\ secs*0.70}$$

$$= 4285714286\frac{cycles}{secs} = 4.285714286 * 10^9\frac{cycles}{secs}$$

$$= \mathbf{4.2857\ GHz}$$

**Answer: For processor P2, we need a new clock rate of 4.2857 GHz.**

- Processor P3

$$New\ Clock\ Rate = \frac{Instruction\ Count * (1.20 * CPI)}{(CPU\ Time * 0.70)}$$

$$= \frac{1.8181818181*10^{10}\cancel{instructions}*(1.20*\frac{2.2\ cycles}{\cancel{instruction}})}{10\ secs*0.70}$$

$$= 6857142856\frac{cycles}{secs} = 6.857142856 * 10^9\frac{cycles}{secs}$$

$$= \mathbf{6.857\ GHz}$$

**Answer: For processor P3, we need a new clock rate of 6.857 GHz.**

3. Assume a 15 cm diameter wafer (one) has a cost of 12, contains 84 dies, and has 0.020 defects/cm^2. Assume a 20 cm diameter wafer (two) has a cost of 15, contains 100 dies, and has 0.031 defects/cm^2.

   a. Find the yield for both wafers.

      To find the yield for both wafers, we first need to find the die area. We note that the equation $Dies\ per\ wafer \approx \frac{Wafer\ Area}{Die\ Area}$ can be re-written as the following: $Die\ Area \approx \frac{Wafer\ Area}{Dies\ per\ wafer}$. Then, we can use the yield formula to find the yield for both wafers.

      - Wafer One Yield

        $Die\ Area \approx \frac{Wafer\ Area}{Dies\ per\ Wafer} \approx \frac{\pi * r^2}{84} \approx \frac{\pi * (7.5cm)^2}{84} \approx 2.103745081\ cm^2$

        We can now find the wafer one's yield:

        $$Yield = \frac{1}{\left(1 + \left(Defects\ per\ area * \frac{Die\ Area}{2}\right)\right)^2}$$

        $$= \frac{1}{\left(1 + \left(\frac{0.020}{\cancel{cm^2}} * \frac{2.103745081\ \cancel{cm^2}}{2}\right)\right)^2} = 0.959216543 \approx \mathbf{0.959}$$

        **Answer: The yield of wafer one is about 0.959 = 95.9%.**

      - Wafer Two Yield

        $Die\ Area \approx \frac{Wafer\ Area}{Dies\ per\ Wafer} \approx \frac{\pi * r^2}{84} \approx \frac{\pi * (10cm)^2}{100} \approx 3.141592654\ cm^2$

        We can now find the wafer two's yield:

        $$Yield = \frac{1}{\left(1 + \left(Defects\ per\ area * \frac{Die\ Area}{2}\right)\right)^2}$$

        $$= \frac{1}{\left(1 + \left(\frac{0.031}{\cancel{cm^2}} * \frac{3.141592654\ \cancel{cm^2}}{2}\right)\right)^2} = 0.9092888489 \approx \mathbf{0.909}$$

        **Answer: The yield of wafer two is about 0.909 = 90.9%.**

   b. Find the cost per die for both wafers.

      To find the cost per die for both wafers, we can use the following formula:

      $$Cost\ per\ Die = \frac{Cost\ Per\ Wafer}{Dies\ Per\ Wafer * Yield}$$

      - Wafer One

        $$Cost\ per\ Die = \frac{12}{84 * .959} = 0.1489646964 \approx \mathbf{0.149}$$

        **Answer: The cost per die for wafer one is about 0.149.**

- Wafer Two

$$Cost\ per\ Die = \frac{15}{100 * .909} = .1650165017 \approx \mathbf{0.165}$$

**Answer: The cost per die for wafer two is about 0.165.**

c. If the number of dies per wafer is increased by 10% and the defects per area unit increases by 15%, find the die area and yield.

- Wafer One

$$New\ Number\ of\ Dies\ per\ Wafer = 84 + 84 * .10 = 92.4$$

$$New\ Die\ Area \approx \frac{Wafer\ Area}{Dies\ per\ Wafer} \approx \frac{\pi * r^2}{92.4} \approx \frac{\pi * (7.5cm)^2}{92.4} \approx 1.912495528\ cm^2$$

$$New\ Number\ of\ Defects\ per\ Area = 0.020 + 0.020 * .15 = 0.023$$

$$New\ Yield = \frac{1}{\left(1 + \left(Defects\ per\ area * \frac{Die\ Area}{2}\right)\right)^2}$$

$$= \frac{1}{\left(1 + \left(\frac{0.023}{\cancel{cm^2}} * \frac{1.912495528\ \cancel{cm^2}}{2}\right)\right)^2} = 0.9574223555 \approx 0.957$$

**Answer: The new die area of wafer one would be about $\mathbf{1.91\ cm^2}$ while the new yield would be around 0.957 = 95.7%.**

- Wafer Two

$$New\ Number\ of\ Dies\ per\ Wafer = 100 + 100 * .10 = 110$$

$$New\ Die\ Area \approx \frac{Wafer\ Area}{Dies\ per\ Wafer} \approx \frac{\pi * r^2}{110} \approx \frac{\pi * (10cm)^2}{110} \approx 2.855993321\ cm^2$$

$$New\ Number\ of\ Defects\ per\ Area = 0.031 + 0.031 * .15 = 0.03565$$

$$New\ Yield = \frac{1}{\left(1 + \left(Defects\ per\ area * \frac{Die\ Area}{2}\right)\right)^2}$$

$$= \frac{1}{\left(1 + \left(\frac{0.03565}{\cancel{cm^2}} * \frac{2.855993321\ \cancel{cm^2}}{2}\right)\right)^2} = 0.9054626421 \approx 0.905$$

**Answer: The new die area of wafer two would be about $\mathbf{2.86\ cm^2}$ while the new yield would be around 0.905 = 90.5%.**

4. 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.

We note that we can find the average CPI by using the following equation:

$$CPI = \frac{CPU\ Time}{Instruction\ Count * Clock\ Cycle\ Time}$$

Furthermore, we note that $1\ ns = 1 * 10^{-9} secs$.

- Compiler A

$$CPI = \frac{1.1\ \cancel{secs}}{1.0E9\ instructions * \frac{1 * 10^{-9}\cancel{secs}}{cycles}} = \mathbf{1.1}\ \frac{\boldsymbol{instructions}}{\boldsymbol{cycle}}$$

**Answer: Compiler A has a CPI of 1.1 instructions per cycle.**

- Compiler B

$$CPI = \frac{1.5\ \cancel{secs}}{1.2E9\ instructions * \frac{1*10^{-9}\cancel{secs}}{cycles}} = \mathbf{1.25}\ \frac{\boldsymbol{instructions}}{\boldsymbol{cycle}}$$

**Answer: Compiler B has a CPI of 1.25 instructions per cycle.**

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?

We note the following equation:

$$CPI = \frac{CPU\ Time}{Instruction\ Count * Clock\ Cycle\ Time}$$

The previous equation can be rewritten to the following:

$$Clock\ Cycle\ Time = \frac{CPU\ Time}{Instruction\ Count * CPI}$$

We note that the CPU times are the same. Therefore, we can use the following equation to find how much faster the clock of the processor running compiler A's code is:

$$\frac{Clock\ Cycle\ Time_A}{Clock\ Cycle\ Time_B} = \frac{\frac{CPU\ Time_A}{Instruction\ Count_A * CPI_A}}{\frac{CPU\ Time_B}{Instruction\ Count_B * CPI_B}} = \frac{\cancel{CPU\ Time_A}}{Instruction\ Count_A * CPI_A} * \frac{CPI_B * Instruction\ Count_B}{\cancel{CPU\ Time_B}}$$

$$= \frac{\cancel{CPU\ Time_B}}{1.0E9 * 1.1} * \frac{1.25 * 1.2E9}{\cancel{CPU\ Time_A}} = \mathbf{1.3636}$$

We note the following: **Clock Cycle Time A = Clock Cycle Time B * 1.3636**

**Answer: We conclude that the processor running compiler A's code is about 1.3636 times slower than the processor running compiler B. In other words, P1 is about 36% slower.**

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?

Let us call this new compiler *Compiler C*. We note that *Compiler C* has 6.0E8 instructions, an average CPI of 1.1., and a clock cycle time of 1ns = 1.0E-9.

Knowing this, we can use the following formula:

$$CPU\ Time = Instruction\ Count * CPI * Clock\ Cycle\ Time$$

$$\text{Therefore: } CPU\ Time = 6.0E8\ instructions * 1.1 \frac{cycles}{instruction} * 1.0 * 10^{-9} \frac{seconds}{cycle}$$
$$= 6.0E8\ \cancel{instructions} * 1.1 \frac{\cancel{cycles}}{\cancel{instruction}} * 1.0 * 10^{-9} \frac{seconds}{\cancel{cycle}}$$
$$= 0.66\ seconds$$

$$\frac{Performance_C}{Performance_A} = \frac{CPU\ Time_A}{CPU\ Time_C} = \frac{1.1\ \cancel{secs}}{0.66\ \cancel{secs}} = \mathbf{1.67}$$

$$\frac{Performance_C}{Performance_B} = \frac{CPU\ Time_B}{CPU\ Time_C} = \frac{1.5\ \cancel{secs}}{0.66\ \cancel{secs}} = \mathbf{2.27}$$

**Answer: The speed up of using the new compiler instead of compiler A is 1.67. The speed up of using the new compiler instead of compiler B is 2.27.**

5. Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C, and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3, and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2, and 2. Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?

| Implementation | Clock Rate (Hz) | CPI | | | |
|---|---|---|---|---|---|
| | | A | B | C | D |
| P1 | $2.5 * 10^9$ | 1 | 2 | 3 | 3 |
| P2 | $3.0 * 10^9$ | 2 | 2 | 2 | 2 |

| Total Instructions | Number of Instructions Per Class | | | |
|---|---|---|---|---|
| | A (10%) | B (20%) | C (50%) | D (20%) |
| 1.0E6 | 1.0E5 | 2.0E5 | 5.0E5 | 2.0E5 |

To find which implementation is faster, we can use the following equation:

$$CPU\ Time = \frac{Instruction\ Count * CPI}{Clock\ Rate}$$

$$CPU\ Time_{P1} = \frac{Instruction\ Count * CPI}{Clock\ Rate} = \frac{\cancel{1.0E6\ instructions} * \frac{(10^5 * 1) + (2 * 10^5 * 2) + (5 * 10^5 * 3) + (2 * 10^5 * 3)\ \cancel{cycles}}{\cancel{1.0E6\ instruction}}}{2.5 * 10^9 \frac{\cancel{cycles}}{second}}$$

$$= \frac{2600000\ seconds}{2.5 * 10^9} = 0.00104\ seconds = \mathbf{1.04 * 10^{-3}\ seconds\ (or\ 10.4 * 10^{-4})}$$

$$CPU\ Time_{P2} = \frac{Instruction\ Count * CPI}{Clock\ Rate} = \frac{\cancel{1.0E6\ instructions} * \frac{(10^5 * 2) + (2 * 10^5 * 2) + (5 * 10^5 * 2) + (2 * 10^5 * 2)\ \cancel{cycles}}{\cancel{1.0E6\ instruction}}}{3.0 * 10^9 \frac{\cancel{cycles}}{second}}$$

$$= \frac{2000000\ seconds}{3.0 * 10^9} = \mathbf{6.667 * 10^{-4} seconds}$$

| Implementation | Execution Time (Seconds) | Performance (1/Execution Time) |
|---|---|---|
| P1 | $1.04*10^{-3}$ | 961 |
| P2 | $6.667*10^{-4}$ | 1500 |

**Answer: Implementation P2 is faster than implementation P1 since it had a better performance.**

a. What is the global CPI for each implementation?

$$Global\ CPI(P1) = \frac{(10^5 * 1) + (2 * 10^5 * 2) + (5 * 10^5 * 3) + (2 * 10^5 * 3) cycles}{1.0E6\ Instructions}$$

$$= \frac{2600000\ cycles}{1.0E6\ Instructions} = \mathbf{2.6\ \frac{cycles}{instruction}}$$

$$Global\ CPI(P2) = \frac{(10^5 * 2) + (2 * 10^5 * 2) + (5 * 10^5 * 2) + (2 * 10^5 * 2) cycles}{1.0E6\ Instructions}$$

$$= \frac{2000000\ cycles}{1.0E6\ Instructions} = \mathbf{2.0\ \frac{cycles}{instruction}}$$

**Answer: The global CPI for P1 is 2.6 while the global CPI for P2 is 2.0**

b. Find the clock cycles required in both cases.

To find the total clock cycles, we can use the following equation:

$$Clock\ Cycles = Instruction\ Count * CPI$$

$$Clock\ Cycles(P1) = 1.0E6\ \cancel{Instruction} * \frac{2.6\ Cycles}{\cancel{Instruction}} = 2600000\ Cycles = \mathbf{2.6 * 10^6\ Cycles}$$

$$Clock\ Cycles(P2) = 1.0E6\ \cancel{Instruction} * \frac{2.0\ Cycles}{\cancel{Instruction}} = 2000000\ Cycles = \mathbf{2.0 * 10^6\ Cycles}$$

**Answer: The clock cycles required for P1 is $2.6 * 10^6$ *Cycles* while the clock cycles required for P2 is $2.0 * 10^6$ *Cycles*.**

6. The Pentium 4 Prescott processor, released in 2004, had a clock rate of 3.6 GHz and voltage of 1.25 V. Assume that, on average, it consumed 10 W of static power and 90 W of dynamic power. The Core i5 Ivy Bridge, released in 2012, had a clock rate of 3.4 GHz and voltage of 0.9 V. Assume that, on average, it consumed 30 W of static power and 40 W of dynamic power.

   a. For each processor find the average capacitive loads.

      We note the following equation: $Dynamic\ Power = \frac{1}{2} * Cpacity\ Load * Voltage^2 * Frequency\ Switched$

      Therefore, $Capacity\ Load = \frac{2 * Dynamic\ Power}{Voltage^2 * Frequency\ Swtched}$

      - Pentium 4 Prescott Processor

        $$Capacity\ Load = \frac{2 * 90W}{(1.25V)^2 * 3.6 * 10^9 \frac{cycles}{seconds}} = \mathbf{3.2 * 10^{-8}}$$

        **Answer: Pentium 4 Prescott Processor has a capacity load of $\mathbf{3.2 * 10^{-8}}$**

      - Core i5 Ivy Bridge

        $$Capacity\ Load = \frac{2 * 40W}{(0.9V)^2 * 3.4 * 10^9 \frac{cycles}{seconds}} \approx \mathbf{2.90486565 * 10^{-8}}$$

        **Answer: Core i5 Ivy Bridge Processor has a capacity load of $\mathbf{2.90486565 * 10^{-8}}$**

   b. Find the percentage of the total dissipated power comprised by static power and the ratio of static power to dynamic power for each technology.

      - Ratio/Percentage for the Pentium 4 Prescott Processor

        $$Total\ Dissipated\ Power = 10W + 90W = 100W$$

        $$Percentage = \frac{10\cancel{W}}{100\cancel{W}} * 100 = .1 * 100 = \mathbf{10\%.}$$

        $$Ratio = \frac{10\cancel{W}}{90\cancel{W}} = \mathbf{0.11\ \ (i.e.\ 1:9\ ratio)}$$

      - Ratio/Percentage for the Core i5 Ivy Processor

        $$Total\ Dissipated\ Power = 30W + 40W = 70W$$

        $$Percentage = \frac{30\cancel{W}}{70\cancel{W}} * 100 = 0.42857 * 100 = \mathbf{42.857\%}$$

        $$Ratio = \frac{30\cancel{W}}{40\cancel{W}} = \mathbf{0.75\ (i.e\ 3:4\ ratio)}$$

c. If the total dissipated power is to be reduced by 10%, how much should the voltage be reduced to maintain the same leakage current? Note: power is defined as the product of voltage and current.

We note the following equation:

$$Total\ Dissipated\ Power = Voltage * Current$$

- Pentium 4 Prescott Processor

$Total\ Dissipated\ Power = 10W + 90W = 100W$

$100\ W = 1.25V * Current$

This means that $Current = \frac{100\ W}{1.25\ V} = 80\ \frac{W}{V}$

If we reduce the total dissipated power by 10%, then the new dissipated power is $(.90)100\ W = 90\ W$.

Therefore, if we want to maintain the same leakage current, then the new voltage should be:

$$New\ Voltage = \frac{90\ \cancel{W}}{80\ \frac{\cancel{W}}{V}} = 1.125\ V$$

To find how much we need to reduce voltage, we do the following calculation:

$$\frac{1.25 - 1.125}{1.25} = .1\ (or\ 10\%).$$

**Answer: For the Pentium 4 Prescott Processor, we would reduce the voltage by 10% in order to maintain the same leakage current. That is, it would need a new voltage of 1.125V.**

- Core i5 Ivy Processor

$Total\ Dissipated\ Power = 30W + 40W = 70W$

$70\ W = 0.90V * Current$

This means that $Current = \frac{70\ W}{0.90\ V} \approx 77.78\ \frac{W}{V}$

If we reduce the total dissipated power by 10%, then the new dissipated power is $(.90)70\ W = 63\ W$.

Therefore, if we want to maintain the same leakage current, then the new voltage should be:

$$New\ Voltage = \frac{63\ \cancel{W}}{77.78\frac{\cancel{W}}{V}} \approx 0.80998\ V$$

To find how much we need to reduce voltage, we do the following calculation:

$$\frac{0.90 - 0.80998}{0.90} \approx .\mathbf{1}\ (\boldsymbol{or}\ \mathbf{10}\%).$$

**Answer: For the Core i5 Ivy Processor, we would reduce the voltage by 10% in order to maintain the same leakage current. That is, we would need a new voltage of about 0.81V.**