|  |
| --- |
|  |
| **Assignment # 1**  **SYSC 5704 – Elements of Computer Systems** |
| Fall 2014  Submitted To  Dr. R. Gregory Franks  By  **Ferhan Jamal (100 953 487)**  Carleton University |

**Q-1: 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 1.0×106 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C, and 20% class D, which implementation is faster?**

**1. What is the global CPI for each implementation?**

**2. Find the clock cycles required in both cases.**

**A-1:** It is given in the question that there are 2 different implementations of the same instruction set architecture. The instructions are divided into 4 classes (Class A, Class B, Class C, and Class D) according to their CPI.

For Processor P1:

Clock Rate=2.5 GHz

Values of CPI for Processor P1 are 1, 2, 3 and 2.

For Processor P2:

Values of CPI for Processor P2 are 2, 2, 2, and 2.

It is also given in the question that the value of dynamic instruction count is 1.0x10^6 instructions which is further divided into 4 classes: 10% Class A, 20% Class B, 50% Class C, and 20% Class D.

*[ CPU Time ( P1 Processor ) = ( Instruction Count x CPI ) / Clock Rate ]*

Value of CPI ( P1 Processor ) =0.1x1 + 0.2x2 + 0.5x3 + 0.2x3= 2.6

Putting the value of CPI ( P1 Processor ) in CPU Time ( P1 Processor ) formula in italic above.

CPU Time ( P1 Processor )=(10^6 x 2.6)/2.5x10^9

**CPU Time ( P1 Processor )= 1.04 milliseconds**

*[ CPU Time ( P2 Processor ) = ( Instruction Count x CPI ) / Clock Rate ]*

Value of CPI ( P2 Processor ) = 2x0.1 + 2x0.2 + 2x0.5 + 2x0.2= 2

Putting the value of CPI ( P2 Processor ) in CPU Time ( P2 Processor ) formula in italic above.

CPU Time ( P2 Processor )= ( 10^6 x 2 )/ 3x10^9

**CPU Time ( P2 Processor )= 0.67 milliseconds ( approximately )**

[The value of CPU Time for P1 Processor is 1.04 milliseconds and the value of CPU Time for P2 Processor is 0.67 milliseconds so we conclude that Processor P2 implementation is faster as compared to Processor P1 ] Answer

Now, we have to find the Global CPI for each implementation and the value of clock cycles required in both cases.

Global CPI (P1 Processor) = [Clock Rate x CPU Time ( P1 Processor)] / Instruction Count

Global CPI (P1 Processor) = [ 2.5 x 10^9 x1.04 x 10^-3 ]/10^6

**Global CPI (P1 Processor) = 2.6**

Global CPI (P2 Processor) = [Clock Rate x CPU Time ( P2 Processor)] / Instruction Count

Global CPI (P2 Processor) = (3 x 10^9 x 0.67 x 10^-3)/10^6

**Global CPI (P2 Processor) = 2.01**

**The values of Global CP1 for P1 and P2 Processor are 2.6 and 2.01**

Clock Cycles (P1 Processor) = Instruction Count x CP1 (P1)

**Clock Cycles (P1 Processor) = 10^6 x 2.6**

Clock Cycles (P2 Processor) = Instruction Count x CP1 (P2)

**Clock Cycles (P2 Processor) = 10^6 x 2.01**

**Q-2: Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56 × 10^9 arithmetic instructions, 1.28×10^9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency. Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.**

**1. Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.**

**2. If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4, or 8 processors?**

**3. To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?**

**A-2:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Instructions Count Per Processor | | | CPI | | |
| Processors | Arithmetic | Load/Store | Branch | Arithmetic | Load/store | Branch |
| 1 | 2.56 x 10^9 | 1.28x10^9 | 2.56x10^8 | 1 | 12 | 5 |
| 2 | 1.8285x10^9 | 0.9142x10^9 | 2.56x10^8 | 1 | 12 | 5 |
| 4 | 0.9142x10^9 | 0.4571x10^9 | 2.56x10^8 | 1 | 12 | 5 |
| 8 | 0.4571x10^9 | 0.2285x10^9 | 2.56x10^8 | 1 | 12 | 5 |

**1.** In the 1st part of the question, we have to calculate the total execution time for the above program on 1, 2, 4 and 8 processors. The total execution time for one processor is calculated by the formula:

CPU Time (One Processor) = (Instructions Count x CPI)/Clock Rate

Instructions Count (one processor) x CPI = (2.56 x 10^9) x 1 + (1.28x10^9) x 12 + (2.56x10^8) x 5 = (2.56x10^9) + (15.36 x 10^9) + (1.28x10^9)

CPU Time (One Processor) = (2.56x10^9) + (15.36 x 10^9) + (1.28x10^9)/2x10^9

CPU Time (One Processor) **=** 19.2/2

**CPU Time (One Processor) = 9.6 seconds**

CPU Time (Two Processors) = (Instructions Count x CPI)/Clock Rate

Instructions Count (Two processors) x CPI = (1.8285x10^9 x 1) + (0.9142x10^9 x 12) + (2.56x10^8 x 5)

Instructions Count (Two processors) x CPI = (1.8285x10^9 x 1) + (0.9142x10^9 x 12) + (2.56x10^8 x 5)

= (1.8285x10^9 x 1) + (0.9142x10^9 x 12) + (2.56x10^8 x 5)

= (1.8285x10^9) + (10.9704x10^9) + (12.8x10^8)

CPU Time (Two Processors) = (Instructions Count x CPI)/Clock Rate

= (1.8285x10^9) + (10.9704x10^9) + (1.28x10^9)/ 2x10^9

= 7.03 seconds

**CPU Time (Two Processors) = 7.03 seconds**

CPU Time (Four Processors) = (Instructions Count x CPI)/Clock Rate

(Instructions Count x CPI) = (0.9142+ 5.4852+ 1.28) x 10^9

CPU Time (Four Processors) = (Instructions Count x CPI)/Clock Rate

= (0.91+ 5.4+ 1.28) x 10^9 / 2 x 10^9

**CPU Time (Four Processors) = 3.8397 seconds**

CPU Time (Eight Processors) = (Instructions Count x CPI)/Clock Rate

(Instructions Count x CPI) = (0.4571+2.64+1.28) x10^9

CPU Time (Eight Processors) = (0.4571+2.742+1.28) x 10^9 / 2x10^9

CPU Time (Eight Processors) = (0.4571+2.742+1.28) x 10^9 / 2x10^9

**CPU Time (Eight Processors) =2.23955 seconds**

Now we have to calculate the relative speed up of the 2, 4 and 8 processors result relative to the single processor result. It is calculated as follows:

Speed up (S) is calculated by the formula = Execution Time (Old) / Execution Time (New)

In the 1st case:

Speed up (S) for 2 processors = CPU Time (One Processor) / CPU Time (Two Processors)

S = 9.6/7.03

S = 1.3655

In the 2nd case:

Speed up (S) for 4 processors = CPU Time (One Processor) / CPU Time (Four Processors)

S = 9.6/3.8397

S = 2.5001

In the 3rd case:

Speed up (S) for 8 processors = CPU Time (One Processor) / CPU Time (Eight Processors)

S = 9.6/2.23955

S = 4.2865

**2.** In the 2nd part of the question, if the CPI of the arithmetic instructions was doubled, the table will look like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Instructions Count Per Processor | | | CPI | | |
| Processors | Arithmetic | Load/Store | Branch | Arithmetic | Load/store | Branch |
| 1 | 2.56x10^9 | 1.28x10^9 | 2.56x10^8 | 2 | 12 | 5 |
| 2 | 1.8285x10^9 | 0.9142x10^9 | 2.56x10^8 | 2 | 12 | 5 |
| 4 | 0.9142x10^9 | 0.4571x10^9 | 2.56x10^8 | 2 | 12 | 5 |
| 8 | 0.4571x10^9 | 0.2285x10^9 | 2.56x10^8 | 2 | 12 | 5 |

CPU Time (One Processor) = [Instructions Count x CPI (new)]/Clock Rate

[Instructions Count x CPI (new)] = (5.12+15.36+1.28) x 10^9 = (21.76) x 10^9

CPU Time (One Processor) = [Instructions Count x CPI (new)]/Clock Rate

CPU Time (One Processor) = (21.76) x 10^9 / 2x10^9

**CPU Time (One Processor) = 10.88 seconds**

CPU Time (Two Processors) = [Instructions Count x CPI (new)]/Clock Rate

CPU Time (Two Processors) = (3.657+10.9704+1.28)x10^9/2x10^9 = (15.9074) x10^9/2x10^9

CPU Time (Two Processors) = (15.9074) x10^9/2x10^9 = 7.9537 seconds

**CPU Time (Two Processors) = 7.9537 seconds**

CPU Time (Four Processors) = [Instructions Count x CPI (new)]/Clock Rate

Instructions Count x CPI (new) = (1.8284+5.4852+1.28) x 10^9 = (8.5936) x 10^9

CPU Time (Four Processors) = (8.5936) x 10^9/ 2x10^9

**CPU Time (Four Processors) = 4.2968 seconds**

CPU Time (Eight Processors) = [Instructions Count x CPI (new)]/Clock Rate

Instructions Count x CPI (new) = (0.9142+2.742+1.28) x 10^9 = (4.9362) x 10^9

CPU Time (Eight Processors) = (4.9362) x 10^9 / 2 x 10^9

**CPU Time (Eight Processors) = 2.4681 seconds**

**3.** In the 3rd part of the question, we have to calculate that to what amount we should reduce the CPI value of load/store instructions in order for a single processor to match the value of 4 processors using original CPI values. So,

CPU Time (One Processor) = (Instructions Count x CPI)/Clock Rate

(Instructions Count x CPI) = (2.56x10^9x1+ 1.28x10^9x12 + 2.56x10^8x5)

CPU Time (One Processor) = (2.56x10^9) + (15.36 x 10^9) + (1.28x10^9)/2x10^9

[2.56 x 10^9 + 1.28 x 10^9 x (12-m) + (1.28x10^9) ] / 2x10^9= 3.8397 seconds

2.56 + 1.28 x (12 -m) + 1.28 = 3.8397

2.56 + 15.36 -1.28m +1.28 = 3.8397x2

2.56 + 15.36 -1.28m +1.28 = 7.6794

m=9.0004

So, CPI should be reduced to: (12-9.0004) = 2.9996 in order for a single processor to match the value of 4 processors using original CPI values.

**Q-3: The results of the SPEC CPU2006 bzip2 benchmark running on an AMD Barcelona has an instruction count of 2.389 × 10^12, an execution time of 750 s, and a reference time of 9650 s.**

**1. Find the CPI if the clock cycle time is 0.333 ns.**

**2. Find the SPECratio.**

**3. Find the increase in CPU time if the number of instructions of the benchmark is increased by 10% without affecting the CPI.**

**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%.**

**5. Find the change in the SPECratio for this change.**

**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.**

**7. This CPI value is larger than obtained in 1.11.1 as the dock 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?**

**8. By how much has the CPU time been reduced?**

**9. For a second benchmark, libquantum, assume an execution time of 960 ns, CPI of 1.61, and clock rate of 3 GHz. If the execution time is reduced by an additional 10% without affecting to the CPI and with a clock rate of 4 GHz, determine the number of instructions.**

**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.**

**11. Determine the clock rate if the CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged.**

**A-3:**

It is given in the question that the value of Instruction Count is 2.389 x 10^12, execution time is 750 s and reference time is 9650 s.

Instruction Count = 2.389 x 10^12

Execution time = 750 seconds

Reference time = 9650seconds

**(1)**

Clock Cycle time = 0.333ns= 0.333 x 10^-9 seconds

Clock Rate = 1/( Clock Cycle time )

Clock Rate = 1/0.333 x 10^-9= 3GHz approximately

The value of execution time i.e. value of CPU Time is 750 seconds

CPI = (CPU Time x Clock Rate) / Instruction Count

CPI = (750 x 3 x 10^9) / 2.389 x 10^12

**CPI = .9418**

**(2)**

The SPECratio is calculate by the formula:

SPECratio = Reference Time / Execution Time = 9650/750

SPECratio = 12.8666

**(3)**

The CPU Time is calculated by the formula:

CPU Time = ( CPI x Instruction Count ) / Clock Rate

If the number of instructions of the benchmark is increased by 10% without the effecting the CPI and also the clock rate **then the CPU Time will also be increased by 10%.**

**(4)**

The CPU Time is calculated by the formula:

CPU Time = (CPI x Instruction Count ) / Clock Rate

It is the given in the question that the number of instructions of the benchmark is increased by 10% and the CPI is increased by 5%

Suppose the value of instruction count is x and the value of CPI is y. It is given in the question that value of the instruction is increased by 10% and the value of CPI is increased by 5% therefore new values of instruction count and CPI are:

Instruction Count (New Value) = x + .1x = 1.1x

CPI (New Value) = y + 0.05y= 1.05y

CPU Time (New Value) = [CPI (New Value) x Instruction Count (New Value)] / Clock Rate

= [(1.05y) x (1.1x)]/Clock Rate

= 1.155xy / Clock Rate

Increase in CPU Time =

(Change in the Value of CPU Time / Original Value of CPU Time) x 100

= 0.155 x 100

= 15.5 %

**(5)**

SPECratio (Old Value) = Reference Time / Execution Time

= 9650/750

=12.86

SPECratio(New Value) = Reference Time/ Execution Time

= 9650 / [(15.5\*750/100) + 750)

= 9650 / (866.25)

= 11.1399

The Change in SPECratio is calculated by the formula

= [SPECratio (Old Value) - SPECratio (New Value) / SPECratio(Old Value) ] x 100

= [(12.86-11.1399)/12.86)] x 100

= 13.3755 %

SPECratio is decreased by 13.3755%

**(6)**

In the 6th part, a new version of the AMD Barcelona is developed with a 4 GHZ clock rate and the instruction count is reduced by 15%. The execution time is reduced to 700 seconds and the new SPECratio is 13.7

Clock Rate = 4 GHZ

Instruction count is reduced by 15% so the new value of instruction count is:

Instruction Count (new) = 2.389 x 10^12 – (15% x 2.389x10^12)

Instruction Count (new) = 2.0306 x 10^12

Execution Time i.e. CPU Time= 700 seconds

SPECratio (new) = 13.7

CPI = ( CPU Time x Clock Rate ) / Instruction Count

CPI = (700 x 4 x 10^9 ) / 2.0306 x 10^12

CPI = 1378.9027 x 10^-3

CPI = 1.3789

**(7)**

It is given in the question that the CPI value is larger as the dock rate was increased from 3 GHz to 4 GHz. The value of CPI at 3GHZ and 4GHZ are:

Value of CPI (3GHZ) = .9418 ( Part 1 of the question )

Value of CPI (4GHZ) = 1.3789 ( Part 6 of the question )

Increase in CPI % = (Change in the Value of CPI / Original Value of CPI) x 100;

= (0.4371/.9418) x 100

= 46.4111 %

Execution Time (Old) = 750

Execution Time (New) = 700

Decrease ( Execution Time ) = [(750-700)/750] \*100

Decrease ( Execution Time ) = 6.67 %

Increase in Clock Rate % = (Change in the value of Clock Rate / Original Value of Clock Rate) x 100

= (1/3) x 100

= 33.3333 %

No, they are dissimilar as CPU time has been decrement by a lower percentage, even though the value of instructions count has been decrement by 15%, the value of clock rate was increased by 33.3333 % and the execution time by 6.67 %. This implies that every single instruction in the instructions count is now taking more time as usual due to the increase in the value of CPI which is 46.411 %.

**(8)**

CPU Time (Original) = 750

CPU Time (New Value) = 700

CPU Time ( Decrease Percentage ) =

(Change in the Value of CPU Time / Original Value of CPU Time) x 100

= (50/750) x 100= **6.6666 %**

**(9)**

In the 9th part of the question, it is given that for a second benchmark libquantum, the value of execution time i.e. CPU Time is 960 ns, CPI is 1.61, and a clock rate of 3 GHZ. Further it is mentioned in the question that the value of execution time is reduced by 10% without effecting the CPI with a clock rate of 4 GHZ. Now the value of execution will be:

For 3 GHZ Processor:

Instructions Count = [CPU Time x Clock Rate] / CPI

Instructions Count = (960 x 10^-9 x 3 x 10^9) / 1.61

Instructions Count = (960 x 10^-9 x 3 x 10^9) / 1.61

Instructions Count = 1788.8198

For 4 GHZ Processor:

Execution Time (New) i.e. CPU Time (New) = 960-(10% of 960) = 864 x 10^-9 seconds

Instructions Count = [CPU Time (New) x Clock Rate] / CPI = (864 x 10^-9 x 4 x 10^ 9)/1.61

**Instructions Count = 2146.5838**

**(10)**

In the 9th part of the question, it is given that CPU Time is further reduced to 10% while maintaining the values number of instructions and CPI unchanged.

Now, the value of CPU Time will be = 960- (10% of 960)

= 864 ns

The value of Clock Rate will be:

New Clock Rate = [1788.8198 x 1.61] / 864

**New Clock Rate = 3.333 GHZ**

**(11)**

In the 11th part of the question, it is given that CPI is reduced by 15% and the CPU time by 20% while the number of instructions is unchanged:

CPI (New Value) = 1.61 – 15%of1.61

CPI (New Value) = 1.61 – 15%of1.61

CPI (New Value) = 1.3685

CPU Time (New Value) = 960 – 20% of 960= 768 ns

Clock Rate = [Instructions Count x CPI (New Value)] / CPU Time (New Value)

Clock Rate = [1788.8198 x 10^9 x 1.3685] / 768

**Clock Rate = 3.1875 GHZ**

**Q-4: Section 1.10 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0×109 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0 × 109 instructions.**

**1. One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2.**

**2. Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor Pl is executing a sequence of 1.0 × 109 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.09 instructions.**

**3. A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.**

**4. Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as MFLOPS = No. FP operations / (execution time ×16) but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both Pl and P2 are floating-point instructions. Find the MFLOPS figures for the programs.**

**A-4:**

**1.** It is given in 1st part of the question that P1 Processor has a clock rate of 4GHZ, average CPI of 0.9 and requires the execution of 5 x 10^9 instructions.

For P2 Processor, clock rate is 3GHZ, average is CPI 0.75 and requires the execution of 1 x 10^9 instructions.

For P1 Processor:

CPU Time = (Instructions Count x CPI) / Clock Rate

= (5 x 10^9 x 0.9) / 4 x 10^9

= 1.125 seconds

For P2 Processor:

CPU Time = (Instructions Count x CPI) / Clock Rate

= (1 x 10^9 x 0.75)/3 x 10^9

= 0.25 seconds

Value of CPU Time for P2 Processor is less than value of CPU Time for P1 Processor

CPU Time (P2 Processor) < CPU Time (P1 Processor)

So, P2 processor has the largest performance than P1 processor.

It is given in the question that the computer with the largest clock rate has the largest performance but unfortunately this is not true in this case as the performance of P2 processor is much better than P1 processor although P1 has the largest clock rate than P2.

**2.**  It is given in the question that for P1 Processor, the value of instructions count is 1.0 x 10^9, CPI = 0.9, Clock Rate = 4 GHZ.

For P2 Processor, we have to calculate the value of instructions in the same time P1 need to execute P1 instructions. The value of CPI for P2 processor is 0.75 and the Clock Rate = 3GHZ.

**For P1 Processor:**

CPU Time = (Instructions Count x CPI)/Clock Rate

CPU Time= (1 x 10^9 x 0.9)/ 4 x 10^9

CPU Time= 0.225

**For P2 Processor:**

Instructions Count= (CPU Time x Clock Rate)/CPI

Instructions Count= (0.225 x 3 x 10^9)/0.75

Instructions Count= 0.9 x 10^9

**3.** MIPS (Processor P1) = (Clock Rate x 10^-6 ) / CPI

= (4 x 10^9 x 10^-6 ) / 0.9

= 4.44 x 10^3

MIPS (Processor P2) = (Clock Rate x 10^-6 ) / CPI

= (3 x 10^9 x 10^-6)/ 0.75

= 4 x 10^3

As value of MIPS (Processor P1) > value of MIPS (Processor P2) therefore the performance of P2 Processor is much better than P1 Processor.

**4.** In the 4th part of the question, MFLOPS (Millions of floating-point operations per seconds) is calculated by the formula:

MFLOPS= (No. of FP Operations/execution time x 10^6)

For P1 Processor:

MFLOPS= (No. of FP Operations/execution time x 10^6)

No. of floating point operations (Processor P1) = 40% of (5.0 x 10^9)

No. of floating point operations (Processor P1) = 2 x 10^9

Execution Time (P1 Processor) i.e. CPU Time (P1 Processor) = 1.125

MFLOPS= (2 x 10^9/1.125 x 10^6)

MFLOPS= 1.78 x 10^3 (approximately)

For P2 Processor:

MFLOPS= (No. of FP Operations/execution time x 10^6)

No. of floating point operations (Processor P2) = 40% of (1.0 x 10^9)

No. of floating point operations (Processor P2) = .4 x 10^9 = 4 x 10^8

Execution Time (Processor P2) i.e. CPU Time (Processor P2) = 0.25

MFLOPS= (4 x 10^8/0.25 x 10^6) = 16 x 10^2

MFLOPS= 16 x 10^2

**Q-5: Another pitfall cited in Section 1.10 is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250 s, with 70 s spent executing FP instructions, 85 s executed L/S instructions, and 40 s spent executing branch instructions.**

**1. By how much is the total time reduced if the time for FP operations is reduced by 20%?**

**2. By how much is the time for INT operations reduced if the total time is reduced by 20%?**

**3. Can the total time can be reduced by 20% by reducing only the time for branch instructions?**

**A-5:**

**1.** In the 1st part of the question, we have to calculate that by how much time the total time is reduced if the time for floating point operations is reduced by 20%

Original Time (FP Operations) = 70 seconds

New Time (FP Operations) = 70- .2x70

New Time (FP Operations) = 70- .2x70 = 56 seconds

Change in time (FP Operations) = 14 seconds

Total time reduced to for the changes above = 250-14 = 236 seconds

Total Time (New) = 236 seconds

Total Time reduced percentage = (Change in Total Time / Original Total Time) x 100

Total Time reduced percentage = (14/ 250) x 100 = 5.6%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instructions | FP Instructions | L/S Instructions | Branch Instructions | INT Instructions |
| Time | 70 seconds | 85 seconds | 40 seconds | 55 seconds |
|  | Total Time = 250 seconds | | | |

**2.** In the second part of the question, we have to calculate by how much is the time for INT operations is reduced if the total time is reduced by 20%

Original Total Time = 250 seconds

FP Instructions (Time) = 70 seconds

L/S Instructions (Time) = 85 seconds

Branch Instructions (Time) = 40 seconds

So, INT Instructions (Time) will be = 55 seconds ( 250 - 70 - 85 – 40 )

It is mentioned in the question that total time is reduced by 20% then New Total Time will be:

New Total Time = 250 - .2x250 = 200 seconds

FP Instructions (Time) = 70 seconds

L/S Instructions (Time) = 85 seconds

Branch Instructions (Time) = 40 seconds

So, the new value of INT Instructions (Time) will be = 5 seconds

Reduced percentage (INT Instructions) =

(Change in the INT instructions time / Original INT Instructions Time) x 100

= (50/55) x 100

= 90.90 %

**(3)** In the third part of the question, we have to find that whether the total time can be reduced by 20% by reducing only the time for branch instructions.

Total Time = FP Instructions Time + L/S Instructions Time + Branch Instructions Time + INT Instructions Time = 250 seconds

If the total time is reduced by 20% then the value of total time now will be: 250 - .2x250 = 200

Total Time (Leaving Branch Instructions) = 70+85+55 = 210 seconds

The value of Total Time (Leaving Branch Instructions) is 210 seconds meaning if we neglect the value of branch instructions then also the value of Total Time (Leaving Branch Instructions) = 210 seconds. This indicates no matter how much we reduce the value of branch instructions, the value of total time will be greater than 200 seconds.

Note : Here 200 seconds is the value which we obtained above after reducing the value of total time by 20%.