COIS 2300 assignment 1

Part A)

A: the minimum size in bytes of the frame buffer to store a frame is

(Resolution) 3840 \* 2160 = 8 294 400

(10 Bits for each) 8 294 400 \* 10 = 82 944 000

82 944 000 bits per frame

82 944 000 /8 = 10 368 000 bytes per frame

There are approximately 10 368 000 bytes per frame.

B: The amount of time it would take to send the frame at 100Mbit/s would be

82 944 000 bits = 82.944 Mbits

82.94/100 = .82944

It would take about .82944 seconds to send 82.94 Mbits at 100 Mbits/s.

Part B

A: the processor that has the highest instructions per second is

Clock rate/ CPI = instructions per second

P1 3.5/1.5 = 2.33 \* 10^9

P2 3 / 1 = 3 \* 10^9

P3 4 / 2.2 = 1.82 \* 10^9

Processor 2 is the fastest and it has a number of instructions per second of (3\*10^9)

B:

Instructions/second \* total time \* CPI = total cycles

P1: 2.333 \* 10^9 instructions / second \* 10 seconds = 23.33 \* 10^9 instructions

1.5 CPI \* 23.33 \* 10^9 instructions = 34.995\* 10^9 cycles

P2: 3.000 \* 10^9 instructions /second \* 10 seconds = 30.00\*10^9 instructions

1.0 CPI \*30.00\*10^9 instructions = 30.00\*10^9 cycles

P3: 1.818 \* 10^9 instructions/ seconds \* 10 seconds = 18.18\*10^9 instructions

2.2 CPI \* 18.18\*10^9 instructions = 39.96\*10^9 cycles

C:

P1:

X = new clock rate

10 \* 0.7 = (3.5 \* 1.2)/ X

X = (3.5 \* 1.2) / .7

X = 6

Processor 1 should have a clock rate of (6 \* 10 ^ 9) to enable a 30% reduced execution time.

P2:

X = new clock rate

.7 = (3 \* 1.2)/X

X = (3 \* 1.2)/.7

X = 5.14

Processor 2 should have a clock rate of (5.14 \* 10 ^ 9) to enable a 30% reduced execution time.

P3:

X = new clock rate

1.818\*2.2/4

.7 = (4 \* 1.2)/ X

X = (4 \* 1.2)/ .7

X = 6.86

Processor 3 should have a clock rate of (6.86 \* 10 ^ 9) to enable a 30% reduced execution time.

Part C)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Clock speed | CPI | Instructions |
| Processor 1 | 4ghz | .9 | 5E09 |
| Processor 2 | 3ghz | .75 | 1E09 |

A) CPU Time = CPI \* instructions/ clock rate

P1 time = .9 \* 5E09 /4

= 1.125 seconds to execute;

Performance = 1/1.125 = 0.888888(relative performance)

P2 time = .75 \* 1E09 / 3

= .25 seconds to execute;

Performance = 1/.25 = 4 (relative performance)

The slower instructions for P1 made it’s performance overall slower than P2 despite being a better processor in the other performance metrics.

B)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Clock rate | CPI | Instructions |
| Processor 1 | 4 | .9 | 1.0E9 |
| Processor 2 | 3 | .75 | ? |

P1 time = .9 \* (1.0 \* 10^6) / 4 \* 10^6

P1 time = 0.225 seconds

X = 0.225 \* 3\* 10^9/ 0.75 = processor 2 instructions

P2 instructions = 0.9E9

The number of instructions that P2 would execute in the time it takes P1 to execute 1.0E9 instructions would be 0.9E9;

C)

MIPS = clock rate / CPI \* 10^6

P1 = MIPS = 4ghz /0.9 \* 10^6

P1 MIPS = 4.44

P2 MIPS = 3ghz / .75 \* 10^6

P2 MIPS = 4

The MIPS rating for P1 was higher than the MIPS rating for P2, but as we learned in the previous questions processor 2 has the faster performance due to the complexity of the instruction set of P1 vs P2, which means that MIPS doesn’t take all performance metrics into account.

D)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clock rate | CPI | Instructions | MFLOPS (40%) |
| Processor 1 | 4ghz | .9 | 5.0E9 | 2.0E9 |
| Processor 2 | 3ghz | .75 | 1.0E9 | 0.4E9 |

MFLOPS = number of FP operations / execution time \* 1E6

P1 execution time = 1.125 (CPI \* instructions/ clock rate)

P2 execution time = 0.25 (CPI \* instructions/ clock rate)

P1 MFLOPS = 2.0E9 / 1.125 \* 1E6

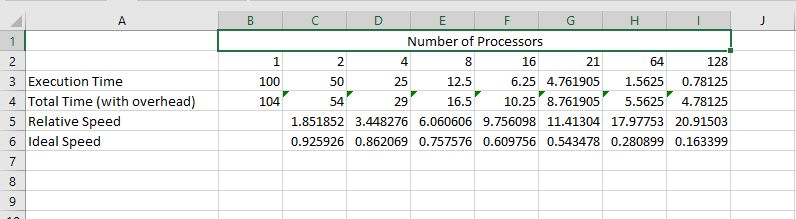
P1 MFLOPS = 1.77E3

P2 MFLOPS = .4E9 / .25 \* 1E6

P2 MFLOPS = 1.6E3

Processor 1 has a higher MFLOPS rating than P2 although similar to the findings of the previous question the MFLOPS are not indicative of a better performance, because we already know processor 2 is faster due to its simplified instructions set.

Part D:



Part E)

This is the byte ordering for Little Endian (left) and big Endian (right);

Little Endian Big Endian

|  |  |  |  |
| --- | --- | --- | --- |
| Address | Data | Address | Data |
| 12 | Ab | 12 | 12 |
| 8 | Cd | 8 | Ef |
| 4 | Ef | 4 | Cd |
| 0 | 12 | 0 | Ab |

Part F:

0xabcdef12 🡪 Decimal;

2\*16^0 + 1\* 16^1 + 15\*16^2 + 14\*16^3 + 13\*16^4 + 12\*16^5 + 11\*16^6 + 10\* 16^7;

2 + 16 + 3840 + 57344 + 851968 + 12582912 + 184549376 + 2684354560

= 2882400018;

0xabcdef12 = 2882400018;

Part G:

A)

600 + 250 + 90 = 940 (old instructions)

(600 \*.75) + 250 + 90 = 790 (new instructions with a 25% decrease in time for arithmetic)

600 \* 1 = 600

250 \* 10 = 2500

90 \* 3 = 270

3370/940 = 3.585 \* 10^6 (original CPI)

450 \* 1 = 450

250 \* 10 = 2500

90 \* 3 = 270

3220/790 = 4.075 \* 10^6 (new CPI)

CPI \* instruction count = execution time

3.585 \* 940 \* clock cycle time \* 10^6 = 3369.9 \* clock cycle time \* 10^6 (old execution time)

4.075 \* 790 \* old clock cycle time \* 10^6 = 3541.18 \* old clock cycle time \* 10^6 = (new execution time)

3541.18/ 3369.9 = 1.051053725140991392104482042149 (this is the amount A is faster than B by)

Making this change would be a bad design choice as the first more complicated instruction set with the faster clock cycle time has a better overall performance than the one with the slowest by 1.05105

B) if we were to double the speed of the arithmetic instructions, the new speed would be

3.585 \* 940 \* clock cycle time \* 10^6 = 3369.9 \* clock cycle time \* 10^6 (old execution time)

3.266 \* 940 \* clock cycle time \* 10 ^6 = 3070 \* clock cycle time \* 10 ^6 (new execution time with doubled speed of the arithmetic instructions)

The relative speedup would be for doubled arithmetic instructions would be (3369.9/3070) = 1.098 \* clock cycle time. Which is about 9% faster.

3.585 \* 940 \* clock cycle time \* 10^6 = 3369.9 \* clock cycle time \* 10^6 (old execution time)

3.011 \* 940 \* clock cycle time \* 10^6 = 2830.34 \* clock cycle time \* 10^6 (new execution time with the speed of arithmetic instructions \* 10)

The relative speedup would be for 10 X arithmetic instructions would be (3369.9/2830.34) = 1.191 \* clock cycle time. Which is about 19% faster.