Rohan Jhaveri
CSU Id: 830962238

ASSIGNMENT 6

Q1 A

Lu Simulations,

Configurations	Core 0	Core 1	Core 2	Core 3
L1_instruction cache: size = 4, associativity = 1 L1_data cache: size = 4, associativity = 1 L2 cache: size = 32, associativity = 4 (base configuration)	Time (ns) = 92806161	Time (ns) = 92804812	Time (ns) = 92804860	Time (ns) = 92804812
L1_instruction cache: size = 4, associativity = 2 L1_data cache: size = 8, associativity = 4 L2 cache: size = 2048, associativity = 4 (random configuration)	Time (ns) = 92761838	Time (ns) = 92760468	Time (ns) = 92760468	Time (ns) = 92760468
L1_instruction cache: size = 32, associativity = 4 L1_data cache: size = 32, associativity = 4 L2 cache: size = 2048, associativity = 8 (maximum configuration)	Time (ns) = 92798890	Time (ns) = 92797536	Time (ns) = 92797509	Time (ns) = 92797536
L1_instruction cache: size = 16, associativity = 2 L1_data cache: size = 32, associativity = 2 L2 cache: size = 2048, associativity = 4 (best configuration)	Time (ns) = 89236693	Time (ns) = 89235396	Time (ns) = 89235442	Time (ns) = 89235396

Speedup= 92806161/90986432 = 1.04

Rohan Jhaveri
CSU Id: 830962238

Q1 B

Cholesky Simulations,

Configurations	Core 0	Core 1	Core 2	Core 3
L1_instruction cache: size = 4, associativity = 1 L1_data cache: size = 4, associativity = 1 L2 cache: size = 32, associativity = 4 (base configuration)	Time (ns) = 190842565	Time (ns) = 190841054	Time (ns) = 190841054	Time (ns) = 190841054
L1_instruction cache: size = 8, associativity = 2 L1_data cache: size = 8, associativity = 2 L2 cache: size = 512, associativity = 8 (random configuration)	Time (ns) = 187100554	Time (ns) = 187098794	Time (ns) = 187098665	Time (ns) = 187098665
L1_instruction cache: size = 32, associativity = 4 L1_data cache: size = 32, associativity = 4 L2 cache: size = 2048, associativity = 8 (maximum configuration)	Time (ns) = 190707072	Time (ns) = 190705544	Time (ns) = 190705598	Time (ns) = 190705544
L1_instruction cache: size = 16, associativity = 1 L1_data cache: size = 32, associativity = 4 L2 cache: size = 512, associativity = 4 (best configuration)	Time (ns) = 182974654	Time (ns) = 182973206	Time (ns) = 182973206	Time (ns) = 182973206

Speedup = 190842565/182974654 = 1.043

Rohan Jhaveri
CSU Id: 830962238

Q2 A

Cholesky Simulation for following configuration,

L1_instruction cache: size = 32, associativity = 4

L1_data cache: size = 32, associativity = 4

L2 cache: size = 256, associativity = 8

Type of Execution	Power (W)	Energy (J)	IPC Core 0 |Core1 | Core 2 | Core 3	Execution Time (ns) Core 0 |Core1 | Core 2 | Core 3
In-Order Execution	15.75	3.0	0.46 | 0.45 | 0.44 | 0.46	190784478 | 190782902 | 190782902 | 190782902
Out-of-Order Execution	30.65	4.04	2.60 | 2.58 | 2.51 | 2.55	131811218 | 131808258 | 131808258 | 131808258

Lu Simulation for following configuration,

L1_instruction cache: size = 32, associativity = 4

L1_data cache: size = 32, associativity = 4

L2 cache: size = 256, associativity = 8

Type of Execution	Power (W)	Energy (J)	IPC Core 0 |Core1 | Core 2 | Core 3	Execution Time (ns) Core 0 |Core1 | Core 2 | Core 3
In-Order Execution	15.11	1.40	0.49 | 0.47 | 0.45 | 0.47	92785989 | 92784604 | 92784750 | 92784604
Out-of-Order Execution	28.60	2.20	2.29 | 2.19 | 2.09 | 2.18	79191672 | 79189268 | 79190424 | 79189268

In both Cholesky & Lu simulations, the Out-of-Order execution has higher power and energy consumption, higher IPC and lower execution time compared In-Order execution.

Power and Energy Consumption:

Out-of-Order execution has additional hardware used for re-ordering instructions which results in the increase in the power and energy consumption.

IPC:

IPC increases because now there will be more number of instructions executed as at a given point of time where dependency is encountered, in which the out-of-order processor will execute another instruction instead of sitting idle and waiting for the dependency to be executed.

Rohan Jhaveri
CSU Id: 830962238

Execution Time:

In In-Order execution the instructions are executed in sequence, so when a dependency comes through, the processor will halt and wait for the dependent instruction is execution not proceed further. Whereas in Out-of-Order execution, the processor will execute all independent instructions in the sequence while it waits for a dependency to come through in the current instruction. This results in reduction of execution time for Out-of-Order execution.

Conclusion:

Hence, we can conclude that the out of order execution is good but the trade-off here is that the power and energy consumption increases by a larger margin.

Rohan Jhaveri
CSU Id: 830962238

Q2 B

IPC depends on cache configurations like cache size, associativity issue width etc. It also inversely depends on frequency. If we decrease the frequency that means we increase the numbers of instructions per cycle which is IPC. Hence, we need to decrease the frequency of Cholesky and Lu to optimize their IPC.

Best Configuration for Cholesky Simulations,

L1_instruction cache: size = 16, associativity = 1

L1_data cache: size = 32, associativity = 4

L2 cache: size = 512, associativity = 4

IPC at frequency 0.5: Core 0	Core 1	Core 2	Core 3
2.56	2.58	2.51	2.60

IPC at frequency 0.2: Core 0	Core 1	Core 2	Core 3
2.63	2.61	2.54	2.58

Decreasing frequency from 0.5 to 0.2 results in increase in IPC. This is because when we decrease the frequency, we are essentially increasing the number of instruction per unit time as discussed above.

Best Configuration for Lu Simulations,

L1_instruction cache: size = 16, associativity = 2

L1_data cache: size = 32, associativity = 2

L2 cache: size = 2048, associativity = 4

IPC at frequency 0.5: Core 0	Core 1	Core 2	Core 3
2.28	2.29	2.23	2.30

IPC at frequency 0.2: Core 0	Core 1	Core 2	Core 3
2.30	2.32	2.28	2.35

The same thing is applicable for Lu as well, but just the margin of increment is less as compared to Cholesky.