Exercise 1

2012-09-15

Pls. mail your homework (converted into **PDF File**) to coa.2012.assignment@gmail.com.

File name: ID_YourName_HW_Sequence.pdf, ex: **5042119123_Obama_HW_01.pdf**（You could save the doc/docx as PDF in word 2007 or later, using 'Save as…'）

文件名不符合要求导致作业丢失后果自负。。。

Exercise 1.1

Find the word or phrase from the list below that best matches the description in the following questions. Use the numbers to the left of the words in the answer. Each answer should be used only once.

1.	virtual worlds	**14.**	operating system
2.	desktop computers	**15.**	compiler
3.	servers	**16.**	bit
4.	low-end servers	**17.**	instruction
5.	supercomputers	**18.**	assembly language
6.	terabyte	**19.**	machine language
7.	petabyte	**20.**	C
8.	datacenters	**21.**	assembler
9.	embedded computers	**22.**	high-level language
10.	multicore processors	**23.**	system software
11.	VHDL	**24.**	application software
12.	RAM	**25.**	cobol
13.	CPU	**26.**	fortran

Questions:

1.1.1 [2] <1.1> Computer used to run large problems and usually accessed via a network

1.1.2 [2] <1.1> 10^{15} or 2^{50} bytes

1.1.3 [2] <1.1> Computer composed of hundreds to thousands of processors and terabytes of memory

1.1.4 [2] <1.1> Today's science fiction application that probably will be available in near future

1.1.5 [2] <1.1> A kind of memory called random access memory

1.1.6 [2] <1.1> Part of a computer called central processor unit

1.1.7 [2] <1.1> Thousands of processors forming a large cluster

1.1.8 [2] <1.1> A microprocessor containing several processors in the same chip

1.1.9 [2] <1.1> Desktop computer without screen or keyboard usually accessed via a network

1.1.10 [2] <1.1> Currently the largest class of computer that runs one application or one set of related applications

1.1.11 [2] <1.1> Special language used to describe hardware components

1.1.12 [2] <1.1> Personal computer delivering good performance to single users at low cost

1.1.13 [2] <1.2> Program that translates statements in high-level language to assembly language

1.1.14 [2] <1.2> Program that translates symbolic instructions to binary instructions

1.1.15 [2] <1.2> High-level language for business data processing

1.1.16 [2] <1.2> Binary language that the processor can understand

1.1.17 [2] <1.2> Commands that the processors understand

1.1.18 [2]<1.2> High-level language for scientific computation

1.1.19 [2] <1.2> Symbolic representation of machine instructions

1.1.20 [2] <1.2> Interface between user's program and hardware providing a variety of services and supervision functions

1.1.21 [2] <1.2> Software/programs developed by the users

1.1.22 [2] <1.2> Binary digit (value 0 or 1)

1.1.23 [2] <1.2> Software layer between the application software and the hardware that includes the operating system and the compilers

1.1.24 [2] <1.2> High-level language used to write application and system software

1.1.25 [2] <1.2> Portable language composed of words and algebraic expressions that must be translated into assembly language before run in a computer

1.1.26 [2] <1.2> 10^{12} or 2^{40} bytes

Exercise 1.3

Consider three different processors P1, P2, and P3 executing the same instruction set with the clock rates and CPIs given in the following table.

Processor	Clock rate	CPI
P1	2 GHz	1.5
P2	1.5 GHz	1.0
P3	3 GHz	2.5

1.3.1 [5] <1.4> Which processor has the highest performance?

1.3.2 [5] <1.4> If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

1.3.3 [10] <1.4> We are trying to reduce the 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?

For problems below, use the information in the following table.

Processor	Clock rate	No. instructions	Time
P1	2 GHz	20×10^9	7 s
P2	1.5 GHz	30×10^9	10 s
P3	3 GHz	90×10^9	9 s

1.3.4 [10] <1.4> Find the IPC (instructions per cycle) for each processor.

1.3.5 [5] <1.4> Find the clock rate for P2 that reduces its execution time to that of P1.

1.3.6 [5] <1.4> Find the number of instructions for P2 that reduces its execution time to that of P3.

Exercise 1.4

Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

	Clock rate	CPI Class A	CPI Class B	CPI Class C	CPI Class D
P1	1.5 GHz	1	2	3	4
P2	2 GHz	2	2	2	2

1.4.1 [10] <1.4> Given a program with 10^6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C and 20% class D, which implementation is faster?

1.4.2 [5] <1.4> What is the global CPI for each implementation?

1.4.3 [5] <1.4> Find the clock cycles required in both cases.

The following table shows the number of instructions for a program.

Arith	Store	Load	Branch	Total
500	50	100	50	700

1.4.4 [5] <1.4> Assuming that arith instructions take 1 cycle, load and store 5 cycles and branch 2 cycles, what is the execution time of the program in a 2 GHz processor?

1.4.5 [5] <1.4> Find the CPI for the program.

1.4.6 [10] <1.4> If the number of load instructions can be reduced by one-half, what is the speed-up and the CPI?

Exercise 1.12

The following table shows results for SPEC2006 benchmark programs running on an AMD Barcelona.

	Name	Intr. count × 10^9	Execution time (seconds)	Reference time (seconds)
a.	perl	2118	500	9770
b.	mcf	336	1200	9120

1.12.1 [5] <1.7> Find the CPI if the clock cycle time is 0.333 ns.

1.12.2 [5] <1.7> Find the SPEC ratio.

1.12.3 [5] <1.7> For these two benchmarks, find the geometric mean.

The following table shows data for further benchmarks.

	Name	CPI	Clock rate	SPECratio
a.	sjeng	0.96	4 GHz	14.5
b.	omnetpp	2.94	4 GHz	9.1

1.12.4 [5] <1.7> Find the increase in CPU time if the number of instruction of the benchmark is increased by 10% without affecting the CPI.

1.12.5 [5] <1.7> Find the increase in CPU time if the number of instruction of the benchmark is increased by 10% and the CPI is increased by 5%.

1.12.6 [5] <1.7> Find the change in the SPECratio for the change described in 1.12.5.

Exercise 1.14

Section 1.8 cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider the following data for the execution of given instruction sequence of 10^6 instructions in different processors.

Processor	Clock rate	CPI
P1	4 GHz	1.25
P2	3 GHz	0.75

1.14.1 [5] <1.8> One usual fallacy is to consider the computer with the largest clock rate as having the large performance. Check if this is true for P1 and P2.

1.14.2 [10] <1.8> Another fallacy is to consider that the processor executing the largest number of instruction will need a larger CPU time. Considering that processor P1 is executing a sequence of 10^6 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 10^6 instructions.

1.14.3 [10] <1.8> 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.

Another common performance figure is MFLOPS (million of floating-point operations per second), defined as

MFLOPS = No. FP operations/execution time $\times 10^6$

but this figure has the same problems as MIPS. Consider the programs in the following table, running on a processor with clock rate = 3 GHz.

	Instr. count	L/S instr.	FP instr.	Branch Instr.	CPI(L/S)	CPI(FP)	CPI(Branch)
a.	10^6	50%	40%	10%	0.75	1	1.5
b.	3×10^6	40%	40%	20%	1.25	0.70	1.25

1.14.4 [10] <1.8> Find the MFLOPS figures for the programs.

1.14.5 [10] <1.8> Find the MIPS figures for the programs.

1.14.6 [10] <1.8> Find the performance for the programs and compare with MIPS and MFLOPS.

Exercise 1.16

Another pitfall, relating to the execution of programs in multiprocessors systems, is expecting improvement in performance by improving only the execution time of part of the routines. The following table shows the execution time of five routines of a program running on different numbers of processors.

	# Processors	Routine A (ms)	Routine B (ms)	Routine C (ms)	Routine D (ms)	Routine E (ms)
a.	2	20	80	10	70	5
b.	16	4	14	2	12	2

1.16.1 [10] <1.8> Find the total execution time and by how much it is reduced if the time of routines A, C, and E is improved by 15%.

1.16.2 [10] <1.8> By how much is the total time reduced if routine B is improved by 10%?

1.16.3 [10] <1.8> By how much is the total time reduced if routine D is improved by 10%?

Execution time in a multiprocessor system can be split into computing time for the routines plus routing time spent sending data from one processor to another. Consider the execution time and routing time given in the following table. In this case, the routing time is an important component of the total time.

# Processors	Routine A (ms)	Routine B (ms)	Routine C (ms)	Routine D (ms)	Routine E (ms)	Routing (ms)
2	20	78	9	65	4	11
4	12	44	4	34	2	13
8	1	23	3	19	3	17
16	4	13	1	10	2	22
32	2	5	1	5	1	23
64	1	3	0.5	1	1	26

1.16.4 [10] <1.8> For each doubling of the number of processors, determine the ratio of new to old computing time and the ratio of new to old routing time.

1.16.5 [5] <1.8> Using the geometric means of the ratios, extrapolate to find the computing time and routing time in a 128-processor system.

1.16.6 [10] <1.8> Find the computing time and routing time for a system with one processor.