Assignment 1
Due: Thursday, October 3, 2013

- There are 12 questions and you are required to answer 10 questions. Last two questions are exercise purpose only.
- **Show your work clearly and legibly for all your answers.**
- The numbers under [ ] indicate the marks allotted to each question.
- The assignment (written or printed) must be stapled and put in a folder
- You may hand in your assignment
  - to the instructor on due date by end of the class or
  - deposit it on hand-in lockers on 4th floor of the E2 building besides the elevator by 430 pm on the due date
- For questions that require on-line searching (for example, cpu-world.com), you must provide the source link (URL).
- A hardcopy "Honesty Declaration" (available in the course web site) appropriately filled and signed must accompany your submission to receive marks for your assignments.

---

1. [2]
   a. M1 performance = 1/10 = 0.1; M2 performance =1/15=0.067
   b. Speed up of M1 over M2= Performance of M1 / Performance of M2= ex. Time of M2 /Ex. Time of M1= 15/10=1.5; M1 is 1.5 faster than M2.

---

2. [2] Machine M2 executes a program in 15s.
   a. If machine M1 is 1.3 times faster than machine M2, what is the execution time on machine M1?
   b. If machine M3 is 1.1 times slower than machine M2, what is the execution time on machine M3?

   a. Speedup (M1 is 1.3 times faster than M2) = Ex. Time on machine M2/Ex. Time of M1;

      That is, 1.5= 10/(ex. Time on M1) ;

      Therefore, Ex. Time on M1 = 15/1.3=11.54 sec

   b. M3 is 1.1 times slower than M2

Ex. Time on M3= 1.1*15=16.5 sec

---

3. [2] If a program P1 takes 5s on a 2 GHz machine,
   a. how many cycles will it take to complete?
   b. What clock rate would be needed to achieve a 1.1 times speedup (hint: assume clock cycles remains same)

   P1 takes 5 s on 2 GHz machine;

   a. Number of cycles P1 takes: =5 * 2 GHz = 5 * $2*10^9$ cycles = 10 billion cycles

   b.
   $$1.1 \text{ speedup} = \frac{\text{Ex.time on slower machine}}{\text{Ex. time on faster machine}} = \frac{\text{Cycles / (Clock rate) in slower}}{\text{Cycles/(Clock rate) in faster}}$$

   $$= \frac{\text{Clock rate in faster machine}}{\text{Clock rate in slower machine}}$$

   Clock rate in faster machine = 1.1 * clock rate in slower machine
   = 1.1 * 2* 10^9 = 2.2 GHz

---

4. [2] Machine M1 runs at 500MHz and Machine M2 runs at 650MHz. A program (P1) requires 100 million cycles on machine M1 and 1.3 times as many on Machine M2
   a. Which machine is faster?
   b. By how much?

   a. Ex. Time in M1=$100*10^6$ / ($500* 10^6$) = 0.2 sec
      Ex. Time in M2 =$130*10^6$ / ($650* 10^6$) = 0.2 sec
      M1 and M2 run at same speed.
   b. 0

---

5. [2] Assume 3 types of instructions in a program
   i. Arithmetic takes 3 cycles
   ii. Conditional takes 2 cycles
   iii. Input/Output takes 4 cycles

```
Xin >>num1;
Xout >> num2;
Num3=num1+num2
If (num3 > 10)
    Xout << "yes";
Else
    Xout<< "no";
```

   b. Total Cycles required 17
   c. CPI (Average) =17/5 = 3.4

---

6. [15] Consider different processors P1, P2 and P3 executing the same instruction set with the clock rates and CPIs given in the following table:
   a. Which processor has the highest performance?
   b. If the processors each execute a program in 7 seconds, find the number of clock cycles and the number of instructions
   c. We are trying to reduce the time by 25% but this leads to an increase in 20% in t

Processor	Clock rate	CPI
P1	2.5 GHz	2.5
P2	1.9 GHz	1.5
P3	3 GHz	2.5

   What clock rate should we have to get this time reduction?

a. CPU Time = CPI * Clock Cycles Time
   i. CPU Time (P1) = 1.5 / 2 *10^9;
      Perf(P1) = 1 / CPU Time(P1) = 2.5 * 10^9 / 2.5 = 1 * 10^9 inst/sec
   ii. Perf(P2) = 1.9 x 10^9/1.5 = 1.267 x 10^9
   iii. Perf (P3) = 3x10^9/2.5 = 1.2 x 10^9

b. CPU Time = 7 sec (Assume number of instructions to be " I ")
   CPU Time (P1) = I x CPI x clock cycle Time
   7 sec = I (P1) x 2.5 / 2 .5* 10^9
   That is, I (P1)= 17.5 / 2.5*10^9 = 7 * 10^9
   Similarly.
   I(P2) = 7 *1.5 /1.9 * 10^9 = 5.526 * 10^9
   I(P3) = 7 * 2.5 / 3 *10^9 = 5.833 x 10^9

c. CPU Time (new) = 7- (7 * 0.25) = 5.25 sec
   CPI(P1) = 2.5 * 0.2 + 2.5 = 3.0
   CPI(P2) = 1.5 * 0.2 + 1.5 = 1.8
   CPI(P3) = 2.5 * 0.2 + 2.5 = 3.0

   Clock Rate = (I * CPI)/CPU Time

   Clock Rate(P1) = (7 * 10^9) (3.0) / 5.25 = 4 GHz
   Clock Rate(P2) = (5.526 * 10^9)(1.8) / 5.25 = 1.8946 GHz
   Clock Rate(P3) = (5.833 * 10^9) (3) / 5.25 = 3.3331 GHz

7. [25] Given the information about the 3 CPUs in the following table answer the questions below explaining briefly how you got your answers:

Feature	CPU-1	CPU-2	CPU-3
Clock rate	2.6 GHz	1.9 GHz	3.6 GHz
Average CPI	2.8	2.8	4.3
Number of cores	2	4	1
Cost	$225	$385	$310

a. [1] How long does it take to run a $30*10^9$ instruction program on CPU-3?
b. [1] How long does it take to run 30 such programs on CPU-1?
c. [1] How long does it take to run 30 such programs on CPU-2?
d. [5] Which CPU offers the best throughput? Show your work clearly.
e. [3] How much faster is CPU-2 than CPU-1? Consider both cases: single job executed on a processor and stream of jobs executed on a processor.
f. [2] Which CPU will run a single job the fastest?
g. [5] Which CPU is most cost effective in terms of throughput?
h. [3] Which CPU is most likely to appeal to a computational scientist? Assume that a computational scientist is likely interested in doing a lot of computation to solve a single large problem and has a large grant to pay for computing equipment. Assume cost is not an issue and that the program does not make use of multiple cores in the CPU.
i. [4] Which CPU is most likely to appeal to a business executive? Assume that a business executive is a "low-end multi-tasker". That is, they are not terribly demanding in terms of computation for any single task but they may need to have several interactive programs running efficiently at once (e.g. spreadsheet, financial planning software, web browser, etc.). Assume cost is an issue.
Note: since actual number of jobs is not provided for this part of the question, this is a subjective question and could lead to multiple answers depending on the scenario you are considering. Discuss your conclusions.

a. [1] CPU-3 Time = IC * CPI * Clock cycle Time
$= (30 * 10^9 * 4.3) / (3.6 * 10^9) = 35.833$ sec

b. [1] CPU-1 Time = $(30 * 10^9 * 2.8) / (2.6 * 10^9) = 32.31$ sec

30 programs, with 2 cores it takes, $30/2 * 32.31 = 484.615$ sec

c. [1] CPU-2 Time = $(30 * 10^9 * 2.8)/ (1.9 * 10^9) = 44.21$ sec
30 programs, 4 cores, it takes, 30/4=7.5; that is 7 programs could be running simultaneously.
It will take 4 cores to finish 28 jobs, and 2 cores to finish the last 2 jobs
$\lceil 30/4 \rceil * 44.21 = 353.68 \text{sec}$

d. [5] Throughput measures the amount of work that can be done per unit time. Normally we assume an "infinite" supply of work is available when computing throughput so we are never waiting for work. Throughput could be measured in jobs per unit time, instructions per unit time, etc. I will choose *jobs per second*.

Looking at the results from *parts a-c*, and assuming we have many jobs to run, we see that CPU-3 can run 1 job in 35.83 seconds, CPU-1 can run 2 jobs in 32.31 seconds. Normalizing this for throughput computation, we might say that CPU-1 can run 1 job in 16.16 seconds (though, of course, in terms of latency this is not so). Similarly, CPU-2 can run 4 jobs in 44.21 seconds or normalized for throughput, we could say 1 job in 11.05 seconds. Thus, taking the reciprocals of the three machines' execution times we get the number of jobs executed per second as follows:

CPU-1 – 1/16.16 = 0.0619 jobs per second,
CPU-2 – 1/11.05 = 0. 0905 jobs per second, and
CPU-3 – 1/35.83 = 0.0279 jobs per second.
Hence, CPU-2 has the best ("biggest") throughput.

e. [3] The answer to this question depends on whether you consider a single job or a large stream of jobs.
For a single job, CPU-1 takes 32.31 seconds (using only a single core) and CPU-2 takes 44.21 seconds (again using only a single core) so in this case CPU-1 is faster. If we have multiple jobs available to run, as in the throughput assessment, then this changes. Consider 4 available jobs.
On CPU-1 we run 2 jobs at a time so it takes 2 * 32.31= 64.62 seconds.
On CPU-2 we can run all 4 jobs at once so it takes 44.21 seconds.
In this case CPU-2 is faster.

f. [2] Having multiple cores does not help in running a single job any faster.
CPU-1 takes 32.31 seconds
CPU-2 takes 44.21 seconds and
CPU-3 takes 35.833 seconds
Therefore, CPU-1 is fastest.

g. [5] Cost effectiveness is a measure of cost for work done. Recaling from part d (for convenience only) we have:
CPU-1 – 0.0619 * 60 = 3.714 jobs/**minute**,
CPU-2 – 0.0905 * 60 = 5.43 jobs/**minute**, and
CPU-3 – 0.0279 * 60 = 1.674 jobs/**minute**.
The costs of the various CPUs are: CPU-1: $225, CPU-2: $385, CPU-3: $310. The work done per dollar for each CPU is:
CPU-1: 3.714 / 225 = 0.0165 jobs per minute per dollar;
CPU-2: 5.43 / 385 = 0.0141 jobs per minute per dollar;
CPU-3: 1.674 /310 = 0.0054 jobs per minute per dollar;
Thus CPU-2 is the most cost effective in terms of throughput. That is, it costs us the least to get an equivalent amount of work done.

h. [3] Since we can ignore the cost of computing, we want to focus on run time for a single job. Assuming also that the scientist's program cannot exploit multiple cores, then we look at *part f* above and choose CPU-1.

i. To support many computationally non-demanding but concurrent tasks, a multi-core system is an obvious choice.
Therefore, CPU-3 is not attractive.
CPU-1 and CPU-2 have the same CPI but B has a slower clock rate.
Since none of the tasks are demanding computationally but we want interactive performance for them all (i.e. concurrently running), CPU-2 is likely the attractive choice when large number of small jobs is to be executed.
Since cost is an issue, and CPU-1 is cheaper, depending on the number of jobs at any given time CPU-1 could be a choice as well.

8. [10] Given the table below. Assume that computes take 1 cycle, loads and store instructions take 10 cycles and branches take 3 cycles. Find the execution time of each program on a 3GHz MIPS processor.

	Compute	Load	Store	Branch	Total
Program 1	1000	400	100	50	1550
Program 2	1500	300	100	100	1750

CPU Clock Cycles = $\sum IxCPI$

a. CPU Clock Cycle (Program 1) = 1000x1 + 400 x 10 + 100 x 10 + 50 x 3 = 6150 cycles

CPU Time = 6150/(3x10^9)=2.05 μs

b. CPU Clock Cycle (Program 2) = 1500+300(10)+100(10)+100(3) = 5800 cycles
CPU Time = 5800/(3x10^9)=1.933 μs

9. [5]
   a. [1] The set of instructions understood by a computer is known as _____ _____
   b. [2] Name 2 architectures that are RISC architectures and two architectures that are CISC architectures
   c. [2] Name 4 different devices from 4 different companies that use RISC architecture

   a. [1] The set of instructions understood by a computer is known as machine language
   b. [2]
      i. RISC architectures (examples): MIPS; PlayStation 2; Other architectures include the list in answer for (c) below. Consider giving marks for other answers, if they are RISC architectures. First verify yourself for correctness.
      ii. CISC architectures: Intel x86; Motorola 68000 series. Give marks to other answers, first verify yourself for correctness before giving marks.
   c. [2]
      iii. ARM — The ARM architecture dominates the market for low power and low cost embedded systems (typically 100–500 MHz in 2008). ARM Ltd., which licenses intellectual property rather than manufacturing chips, reported that 10 billion licensed chips had been shipped as of early 2008.[16] The various generations, variants and implementations of the ARM core are deployed in over 90% of mobile electronics devices, including almost all modern mobile phones, mp3 players and portable video players. Some high profile examples are:
         - Apple iPods
         - Apple iPhone and iPod Touch
         - Apple iPad
         - Palm and PocketPC PDAs and smartphones
         - RIM BlackBerry email devices, smartphones
         - Microsoft Windows Mobile
         - Nintendo Game Boy Advance
         - Nintendo DS
         - Sony Network Walkman (Sony in-house ARM based chip)
         - Android smartphones, tablets

iv. MIPS's MIPS line, found in most SGI computers and the PlayStation, PlayStation 2, Nintendo 64, PlayStation Portable game consoles, and residential gateways like Linksys WRT54G series.
v. IBM's and Freescale's (formerly Motorola SPS) Power Architecture, used in all of IBM's supercomputers, midrange servers and workstations, in Apple's PowerPC-based Macintosh computers (discontinued), in Nintendo's Gamecube and Wii, Microsoft's Xbox 360 and Sony's PlayStation 3 game consoles, EMC's DMX range of the Symmetrix SAN, and in many embedded applications like printers and cars.
vi. SPARC, by Oracle (formerly Sun Microsystems), and Fujitsu
vii. Hewlett-Packard's PA-RISC, also known as HP-PA, discontinued December 31, 2008.
viii. Alpha, used in single-board computers, workstations, servers and supercomputers from Digital Equipment Corporation, Compaq and HP, discontinued as of 2007.
ix. XAP processor used in many low-power wireless (Bluetooth, Wi-Fi) chips from CSR.
x. Hitachi's SuperH, originally in wide use in the Sega Super 32X, Saturn and Dreamcast, now at the heart of many consumer electronics devices. The SuperH is the base platform for the Mitsubishi–Hitachi joint semiconductor group. The two groups merged in 2002, dropping Mitsubishi's own RISC architecture, the M32R.
xi. Atmel AVR used in a variety of products ranging from Xbox handheld controllers to BMW cars.

10. [10] Read the article by David Kanter (that compares between Intel's Sandy Bridge (SB) architecture and its previous Nehalem (NH) architecture; http://www.realworldtech.com/page.cfm?ArticleID=RWT091810191937):
    a. How many single precision FLOPs/cycle can SB sustain?
    b. How does this compare to NH's capabilities?
    c. Next, what is SB's speedup in total L1 cache bandwidth over NH and how is it achieved?
    d. What is the relative performance of SB's L2 cache's load-to-use latency compared to NH?
    e. What about the speedup of SB's L3 cache (take the best numbers) over NH's L3 cache?

a. 16 Single Precision FLOPs/cycle, double the capabilities of NH
b. 50% more due to the addition of an additional 128-bit load (the 128-bit isn't necessary)
c. 10 cycles (NH) / 12 cycles (SB) = 0.833 as fast.
d. SB's performance is actually worse than NH's so its relative performance is only .833 of NH. Would also accept 1.2 times **slower**.
e. 35 cycles (NH) / 26 cycles (SB) = 1.346 times faster than NH.