IC220 – Course Paper Cover Sheet

Your Last Name: Your First Name: Section:

The maximum amount of credit this assignment is eligible for (check one):

- ☐ Full credit (submitted promptly at start of class on the due date)
- ☐ 20% off (submitted before 0800 on the following business day)
- ☐ No credit (submitted after that. All assignments must be submitted to possibly earn passing grade)

On the back is the grading rubric that will be used for your paper. Also, attached is an example of a previously submitted strong paper – you may find it helpful to see what it was like. Notice some of its strong features:

- Good summary of the problem
- Doesn't just rehash what others say, but adds some personal assessment and analysis
- Considers multiple points of view
- Relates the specific problem to computing in general

This paper assignment was previously announced. Attached is a copy of the instructions.

Some reminders:

1. **Please do not place your name on the paper – only on this coversheet, stapled to your paper.**
2. **Be sure to cite your sources (you can use footnotes or parenthetical citations)**
3. **Use your own words. Use quotations where necessary, but this should be infrequent and brief.**

Feedback – please answer for credit (10 pts) but will not otherwise affect your grade

1. How long did this assignment take you to complete? (circle one)
   0.5 hr 1 hr 2 hrs 3 hrs
   4 hrs 5 hrs 6 hrs Other:
2. What was the best part about this assignment?

3. The worst?

4. How helpful was the in-class peer review to improving your paper?
   Very helpful Helpful Okay Not useful Total waste of time

5. Was this assignment useful to your learning?

6. Was this assignment: (circle one)
   Way too easy Little too easy About right Little too hard Way too hard

7. Is there anything that I (the instructor) could have done better? Or other comments?

WRITTEN PAPER RUBRIC

Name: _____ Date: _____

Course:_____ Title/Topic: _____

Thesis Statement/ Introduction	Clearly and concisely states the paper's purpose. The introduction is engaging, states the main topic and previews the structure of the paper. ☐	Clearly states the paper's purpose. The introduction states the main topic and previews the structure of the paper. ☐	States the paper's purpose. The introduction states the main topic but does not adequately preview the structure of the paper. ☐	Incomplete and/or unfocused. There is no clear introduction or main topic and the structure of the paper is missing. ☐	N/A ☐
Body and Content	Each paragraph has thoughtful supporting detail sentences that develop the main idea. Writer demonstrates logical sequencing of ideas through well-developed paragraphs; transitions are used to enhance organization. ☐	Each paragraph has sufficient supporting detail sentences that develop the main idea. Paragraph development present but not perfected. ☐	Each paragraph lacks supporting detail sentences. Logical organization; organization of ideas not fully developed. ☐	Each paragraph fails to develop the main idea. No evidence of structure or organization. ☐	N/A ☐
Conclusion	The conclusion is engaging and restates the thesis. Conclusions are strongly supported by the paper. ☐	The conclusion restates the thesis. Conclusions are supported by the paper. ☐	The conclusion does not adequately restate the thesis. Some conclusions are not supported by the paper. ☐	Incomplete and/or unfocused. Little indication of synthesis or drawing of conclusions. ☐	N/A ☐
Mechanics and Usage	No errors in punctuation, capitalization and spelling. No errors in sentence structure and word usage. ☐	Almost no errors in punctuation, capitalization and spelling. Almost no errors in sentence structure and word usage ☐	Many errors in punctuation, capitalization and spelling. Many errors in sentence structure and word usage. ☐	Numerous errors in punctuation, capitalization and spelling. Numerous errors in sentence structure and word usage. ☐	N/A ☐
Citation and Bibliography	All cited works are noted in the correct format with no errors. The paper includes, if applicable, more than the specified number of proper references, such as journal articles and/or books. ☐	Some cited works are noted in the correct format. Inconsistencies evident. The paper includes, if applicable, the minimum specified number of proper references, such as journal articles and/or books. ☐	Few cited works are presented in the correct format. The paper includes, if applicable, less than the minimum specified number of proper references, such as journal articles and/or books. ☐	Absent. Includes few proper references, such as journal articles and/or books. ☐	N/A ☐

Instructions (as previously distributed)

Assignment: Research and write a paper discussing a current computer architectural topic/issue. Specific requirements:

1. **Originality:**This topic and paper must be researched and written this semester for this class(i.e. a paper from a previous class is not allowed to be used).
2. **Sources:** You must have at least three sources. Wikipedia may be used, but counts for at most ONE source.
3. **Bibliography:** Append a list of resources/references used to the end of your paper. You can pick the format.
4. **Length:** 3 to 5 pages.
5. **Content:** Your paper should have some analysis, not just a rehashing of the sources that you found. For instance, if discussing a problem, how serious a problem is it and are the solutions you discovered likely to really work?
6. **No name!** Please **do not** place your name on the paper itself -- only on the coversheet that will be provided and stapled to your paper.
7. **Quotations:** Use your own words. Use quotations where necessary, but this should be infrequent and brief.
8. **Format:** Your paper must be typed.

Specific deliverables:

1. **Paper description:** Submit your description to your instructor (see calendar for deadline) via plain text e-mail (do NOT send a Microsoft Word document etc.). Include the following:
   - A **topic**. See topic suggestions below. You may also propose your own. One restriction: you may **not use the Pentium division bug as a topic!** (later you will get a sample paper with this topic).
   - A **summary**. In a few sentences, describe the issue, your sources, and, if applicable, the particular solution(s) that you will consider for this problem.
2. **TWO copies of Paper v1.0**. See calendar for due date. You will submit a complete, well-written draft of your paper (see calendar for deadline). Very soon thereafter, we will devote one day of class to peer review: everyone will read and give constructive feedback on two papers. You will receive this feedback, and will use it to revise the paper.
3. **Final paper submission.** See calendar for due date. Be sure that you have revised the paper based on the comments you received. Turn in:
   - Paper coversheet (to be provided)
   - Your final, polished paper. Staple to coversheet but do NOT put your name on the paper itself.
   - The two copies of Paper v1.0, with comment sheets. Do not staple these to the full paper (you will turn in via a separate pile).

Grading:

- 5%: Paper description, on-time. A topic alone is not a description.
- 15%: Paper v1.0 submitted on-time and complete.
- 5%: Providing good feedback to other students' papers.
- 75%: Final paper.

Sample paper (spacing compressed to fit on fewer pages)

October 30, 1994 a professor of mathematics at Lynchburg College named Dr. Thomas R. Nicely made a public announcement that he had discovered a flaw in the Pentium I chipset released by Intel. He noted that the floating point unit would return erroneous values for particular division operations. This incorrect result could be reproduced on any machine that contained the processor with the bug in any number of programs. This means that it was possible to test for the error in compiled code, any ordinary spreadsheet program such as Excel, or the basic operating system calculator. Nicely had discovered this flaw in June 2004 but did not feel confident in diagnosing the problem until he had conducted further testing. On October 24, 1994 he contacted Intel with the information about the problem. Nicely says in his public memo, "The contact person later reported that the bug was observed on a 66-MHz system at Intel, but had no further information or explanation, other than the fact that no such bug had been previously reported or observed."

The best example of when the chip would produce a very significant error can be found in the division of two large numbers. In the formula z=x-(x/y)*y where 'x' equals 4195835 and 'y' equals 3145727 the correct answer should be exactly zero. The affected Pentium chip would produce 256 as the correct answer. While this did happen, this was an extreme case and most erroneous results were only off by a few bits. Intel's initial claim was that after months of testing it was determined this error would only occur once in about nine billion division operations. Intel had tested the occurrences of the bug by using completely random numbers. In the first article actually covering the incident, Intel explains that this would not affect the common computer user at all in their work. An Intel spokesman said of Nicely "He's the most extreme user. He spends round-the-clock time calculating reciprocals. What he observed after running this for months is an instance where we have eight decimal points correct, and the ninth not showing up correctly. So you get an error in the ninth decimal digit to the right of the mantissa. I think even if you're an engineer, you're not going to see this." Intel did make an immediate fix to all processors prior to their shipping after the bug was made public but they did not correct it when they discovered small errors in their labs prior to Nicely's memo.

Intel attributed the bug to a very subtle problem in a specific algorithm that closely resembles long division. The SRT algorithm is the one used for floating point division on the Pentium chip. Instead of using base ten decimals it uses base 4. This serves for very quick performance during large division computations. A triangular array with stored values is used to accomplish its calculations. The problem occurred when the SRT algorithm was implemented in silicon. Five values out of 1066 in the table were dropped, resulting in empty cells. These five were supposed to contain the value positive two but since there was no entry in those spaces any time an attempt to fetch information from them would be made zero would be returned. This caused an issue where the remainder may have been calculated slightly wrong but due to that value being carried forward it may increase the rate of error in the computation.

Intel's initial assessment using completely random numbers was accurate in producing only one error in nine billion division operations. However, their claim that it would only affect very specific users may not be entirely accurate. A study done by IBM produced results quite differently from those that Intel had released. Using their logic (which seems much more likely than Intel's) a problem occurs on the average once in one hundred million division operations. Not only was this error more likely but IBM also questioned Intel's assumption that the average user only did 1000 divides a day. They cited independent studies that showed a spreadsheet user completing common tasks doing 5000 divides a second. In only 15 minutes that is 4.2 million divides. These odds affect the average user much more than Intel initially speculated. It was also noted that numbers and calculations used commonly during financial calculations produced erroneous results more often than random numbers. This means many common users will encounter the problem several times a day. This would have a tremendously adverse affect on everyday life given the need of these programs by common users.

The accuracy of the calculations is no doubt a problem that required a resolution immediately. One of the major mistakes that Intel made was that they did not confront this problem when they had initially discovered it. They waited until a big deal was made about it and they completed a misleading study to avoid replacing many faulty processors. They initially decided to treat every claim on a case by case basis while denying common users a replacement. This is a bad business practice to sell a faulty piece of equipment to a user and then refuse to fix it unless they only did complex mathematical operations on a daily basis. They did not change this policy until IBM decided to quit selling this particular chip. Intel claimed that IBM was overreacting with their study but conceded to replace any customers chip if asked.

This type of error may not seem like a big deal if finances are miscalculated but the magnitude of the problems

it can cause are much greater than one might notice. On June 4, 1996 the Ariane 5 rocket exploded 37 seconds after liftoff due to an inaccuracy in a number. When a floating point was converted to an integer it was too large to be stored in the allocated space. The computer cleared its memory and instructions were given to its rocket nozzles, which caused an explosion.

Design in the computer world is very important but there will undoubtedly be mistakes. Intel's mistake was not so much in making the mistake since it was almost impossible to look for it beforehand. The mistake made was downplaying the bug to the average user. Computer architects have a responsibility to their customers to produce quality products. When a bug is discovered it should be standard practice to repair the problem without justification. These problems may not be a big deal to some users but to others it could mean years of research out the window or even human death.

Works Cited

Pentium Study. 12 Dec. 1994. IBM. 12 Dec. 1994 <http://www.ibm.com/Features/pentium.html>. (This source was accessed through an archive titled Pentium Division Bug Documents available at http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=1666&objectType=file)

Peterson, Ivars. Pentium Bug Revisited. 12 May 1997. Mathematical Association of America. 27 Apr. 2005 <http://www.maa.org/mathland/mathland_5_12.html>.

Wolfe, Alexander. "Intel Fixes A Pentium FPU Glitch." EE Times (1994)