The University of Michigan
Department of EECS

EECS 573 – Microarchitecture
Prof. Todd Austin

Midterm Exam
November 30th, 2016
10:40pm-12:00pm
Open Book, Computer and Notes (no Internet or communication!)

Name: _____

Problem 1 – What this course is all about	_____/15
Problem 2 – Reliable system design	_____/10
Problem 3 – Secure and safe systems	_____/10
Problem 4 – Application-specific processors	_____/10
Problem 5 – Papers and presentations	_____/12
TOTAL	_____/57

Attempt to solve **all** the problems in this exam. Use your time wisely, and pay attention to the point distribution. Think before you plunge, if you are spending too much time on one part, move on to another one. All problems are divided in multiple parts; each part is independent from the others.

If you need more space to work out some the problems, use the backside of the exam sheets using a statement like "Go to back of page 6". When in doubt, state any assumption you make. Show all your work, you will get partial credit for partial answers. Good luck!

HONOR PLEDGE:
"I have neither given not received aid on this exam, nor have I concealed any violations of the Honor Code."
Signature:
_____

1. What this course is all about – 15 points

[Hint: Don't write an essay. You should answer with just a few sentences per question.]

1A. (3 points) The "bathtub curve" represents the probability of failure for transistors over the lifetime of a design. (i) Why is probability of failure high immediately after a transistor is manufactured? (ii) What do manufacturers do to prevent most early transistor failures from happening in the field? (iii) As transistors scale to smaller sizes, are they more reliable or less reliable? Explain your answer.

1B. (3 points) While dual modular redundancy (DMR) is widely used for tolerating transient faults, it is not widely used for tolerating permanent faults. (i) Why doesn't DMR work well for tolerating permanent faults? (ii) Describe how a system could utilize DMR to tolerate transient faults?

1C. (3 points) Process variation is a property of modern transistors where their dimensions and chemical makeup are statistical in nature. (i) Why does high process variation lead to sub-optimal designs? (ii) Why is process variation on the rise? (iii) Who worries about process variation more, ***and why***? a) The architect of a super-pipelined high performance microprocessor with few levels of logic per stage, or b) the architect of a low-power low-frequency embedded microcontroller with many levels of logic per stage?

1D. (3 points) (i) What is a buffer overflow attack? (ii) Describe two ways that they can be prevented.

1E. (3 points) Near-memory computing has received much attention in the architecture community over the last 20 years, and it is again the focus of much research. (i) Give one reason why near-memory computing has not "caught on" in industry as of yet? (ii) What recent development in silicon processing has caused a resurgence of interest in near-memory computing, and why does this development enable near-memory computing? (iii) Which of the following algorithms would likely benefit most from a near-memory computing architecture, **and why**? a) an AES encryption kernel, b) a database sort operation, or c) a video decoder?

2. Reliable System Design – 10 points

The following questions are related to reliable system design techniques from the papers we read this semester.

2A. (5 points) The RelaxFault design allows the last-level cache to help repair broken DRAMs. (i) In what ways can a DRAM break, please give two examples? (ii) How does FreeFault repair DRAM with the last-level cache? (iii) How does RelaxFault improve upon the capabilities of FreeFault? (iv) How does RelaxFault know where to find the last-level cache block used to repair an access to a faulty DRAM address?

2B. (5 points) Perturbation-based fault screening attempts to lower the cost of finding and fixing transient faults. (i) What is a "perturbation" and how does it relate to transient faults? (ii) Describe two types of fault screeners. (iii) What is a "false positive", and how do they affect the ***correctness*** and ***performance*** of perturbation-based fault screening?

3. Secure and Safe Design – 10 points

The following questions are related to secure and safe design techniques from the papers we read this semester.

3A. (5 points) The Rowhammer vulnerability allows attackers to manipulate memory in an insecure manner. (i) What characteristic(s) of DRAMs allow the Rowhammer vulnerability to exist? (ii) What inherent protection mechanism exists that prevents normal programs from experiencing the Rowhammer vulnerability? (iii) List two ways that attackers defeat this inherent protection mechanism? (v) The authors proposed a lightweight protection scheme called PARA. Briefly explain how PARA works and how it can be highly effective at low cost.

3B. (5 points) A code gadgeting attack is a class of security exploit that has received much attention from the research community due to its stealthy nature. (i) Why are code gadgeting attacks difficult to detect? (ii) What is the primary challenge in implementing a code gadgeting attack? (iii) Kayaalp's work attempts to detect this type of attack using a signature-based defense mechanism to look for possible jump-oriented programming (JOP) attacks. According this work, what is the signature of JOP, and why do these attacks exhibit this type of pattern?

4. Application-Specific Processor Design - 10 points

The following questions are related to application-specific design techniques from the papers we read this semester.

4A. (5 points) Loops are very common in program executions. (i) In Campanoni's work on automatic parallelization of irregular programs, which type of loop (small/large) does the proposed design target and why? (ii) What is the biggest bottleneck for the targeted parallelization and what is the cause of the said bottleneck? (iii) How do the authors propose to use hardware/software co-design to overcome this bottleneck? What hardware is introduced and how does it help mitigate the problem?

4B. (5 points) The Convolution Engine works to speed up the processing of convolution-like data-flow, such as those found in computational photography, image processing, and video processing applications. (i) What is a 2D register file, and why does it work well for the algorithms running on the Convolution Engine? (ii) List two ways that the graph fusion unit can save power, compared to executing the algorithm on a CPU? (iii) In the results, the authors compare designs based on "Ops/mm^2". What is the meaning of this metric, and why is it particularly relevant for applications that run on the Convolution Engine?

5. Papers and Presentations – 12 points

5A. (6 points) Possessing a well-developed skill for presentation will serve you in any walk of life. Please give three pitfalls to avoid when presenting technical research.

5B. (6 points) One of the goals of EECS 573 is to expose you to the research **process**. Please list three things you learned about conducting research? (Specifically, list here what you learned about "conducting" research, not presenting it or writing it up.)