The University of Michigan - Department of EECS

EECS578
Computer-Aided Design Verification of Digital Systems

Midterm exam – April 2, 2014

Name: ______________________________________

This exam is **CLOSED BOOKS, CLOSED NOTES**. You can only have pencil/pen and eraser. No electronic device is allowed. The exam has **7** questions. Questions vary in difficulty; it is strongly recommended that you do not spend too much time on any one question. For questions where a box or a line for your answer is provided, please put your final answer in the box.

IMPORTANT: always show your work in deriving your answers in questions 3-7. **Correct answers without showing your work will receive no credit.**

Question	**Points**
1 – True/false and short questions	**/28**
2 – BDDs, cofactors and sets	**/14**
3 – Research papers	**/20**
4 – SAT solvers and CNF	**/12**
5 – Event-driven simulation	**/10**
6 – CTL-based model checking	**/20**
7 – Reachability analysis	**/18**
TOTAL	**/122**

The rules of the Honor Code of the University of Michigan - College of Engineering apply for this exam:

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

Signature: _________________________________
(Exams without a signed pledge will not be graded)

1. True/False and short questions and answers – 28 points

This question spans several of the topics covered in class.
Determine if the statements in A-E below are TRUE or FALSE and circle the correct option.

A. [1pt] An escaped bug is a design bug that is not detected during pre-silicon validation, and it is only caught during post-silicon validation.

TRUE / FALSE

B. [1pt] A multicore processor is a microprocessor with multiple memory modules controlled by one core's load/store unit.

TRUE / FALSE

C. [1pt] Formal verification encompasses all techniques that leverage mathematical reasoning and proofs to determine the correctness of a silicon design.

TRUE / FALSE

D. [1pt] In "DACOTA" [Deo09], consistency bugs can be detected by constructing a graph based on the activity log, and then checking if the graph is connected (if the graph is indeed connected, no bug is present).

TRUE / FALSE

E. [1pt] "Reversi" [Wag08] is a solution for post-silicon validation of micro-processors where the instructions of a test program are executed in reverse order (from last to first), instead of the normal order, so that bugs occurring late in the program can be detected faster.

TRUE / FALSE

Please answer the questions below, **WRITING NEATLY**:

F. [2 pts] In the context of memory protocols, what does (simply expand the acronym):

SC stand for? ________________________________

TSO stand for? ________________________________

G. [2 pts] What mechanism(s) does RMO provide to enforce ordering restrictions?

(<6 words) ____

H. [2 pts] Below is the mapping of the mask bits in the MEMBAR instruction of a chip multiprocessor:
bit 0: if bit is set, a **load** cannot be reordered before/after other **loads** around it,
bit 1: if bit is set, a **store** cannot be reordered before/after other **loads** around it,
bit 2: if bit is set, a **load** cannot be reordered before/after other **stores** around it,
bit 3: if bit is set, a **store** cannot be reordered before/after other **stores** around it,
How should these bits be set if we wanted to emulate an SC protocol by using MEMBAR instructions? Circle the correct value for each bit.

bit0 0 / 1 bit1 0 / 1 bit2 0 / 1 bit3 0 / 1

I. [2 pts] How does wormhole flow control work in NoCs?

(<10 words) ____

J. [4 pts] Post-silicon vs. runtime validation: which of the following are critical concerns in designing solutions for each? Performance impact, silicon area, power consumption, availability of a bug-recovery mechanism.

- Critical concerns in post-silicon solutions:

  ____

- Critical concerns in runtime solutions:

  ____

K. [3 pts] Name three of the coverage metrics that we discussed in class

1. (<4 words)________________________________
2. (<4 words) ________________________________
3. (<4 words) ________________________________

L. [1 pt] What is the underlying engine in classic Bounded Model Checking?

(<4 words)________________________________

M. [4 pts] Which of the following techniques can be used to prove a general SAFETY PROPERTY (not BOUNDED)? Circle all that applies:

- Sequential equivalence checking
- Symbolic simulation
- Reachability analysis
- Logic simulation
- CTL Model checking

N. [3 pts] Look at the powerpoint slide below. Suggest three improvements that you would make if it were your slide.

A. (<10 words) ____

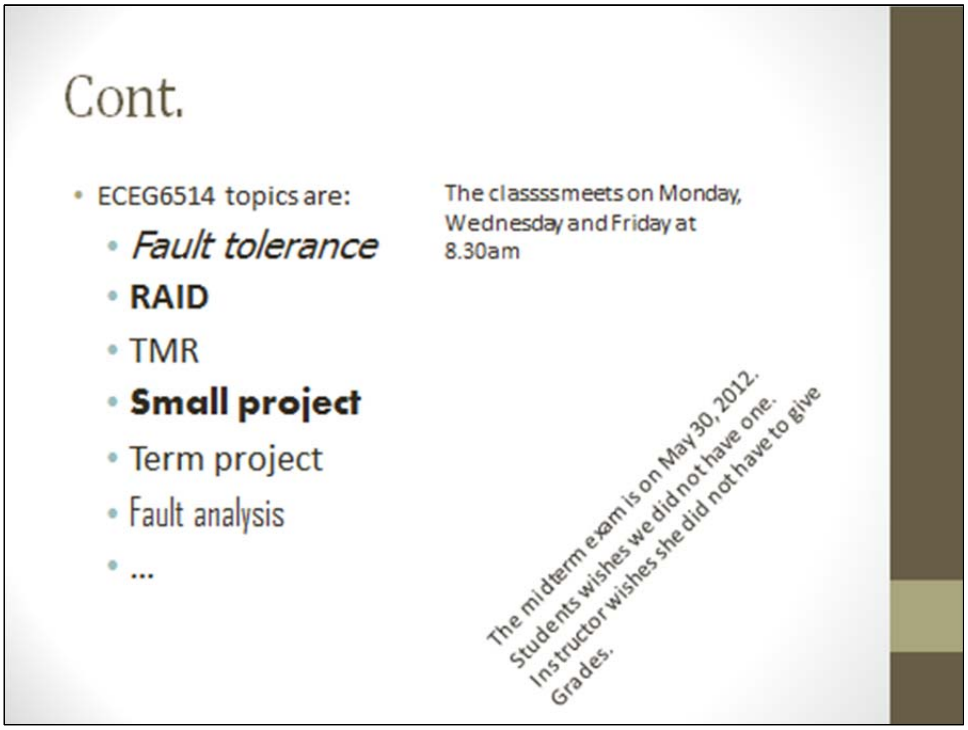
____

B. (<10 words) ____

____

C. (<10 words) ____

____



Cont.

- ECEG6514 topics are:
  - *Fault tolerance*
  - **RAID**
  - TMR
  - **Small project**
  - Term project
  - Fault analysis
  - ...

The classssmeets on Monday, Wednesday and Friday at 8.30am

The midterm exam is on May 30, 2012.
Students wishes we did not have one.
Instructor wishes she did not have to give
Grades.

2. BDDs, cofactors and set representation – 14 points

Consider the function below:

f = c b + b (a + d) + a’d’

For the questions below, use variable order (from top to bottom): a,b,c,d.

A. [4 pts] What are the cofactors of F w.r.t. a? Provide a simplified expression with the minimum number of literals:
Show your work to receive credit for this question

$F_{a=0}$ = ____

$F_{a=1}$ = ____

B. [4 pts] Draw the BDD for the function **f**. Remember to indicate the root by drawing an arrow that points to the root node. Remember also to mark the edge with ‘0’ or ‘1’ to indicate 0 and 1 cofactors. Please draw the ‘0’ edges on the left and the ‘1’ edges on the right.

C. [6 pts] This question is unrelated to the previous parts: consider the following vector set over the Boolean variables <a,b,c,d>:
{ <0010>, <0011>, <1100>, <1110> }.
Draw the BDD to represent this set. Remember to mark the edge with ‘0’ or ‘1’ to indicate 0 and 1 cofactors. Please draw the ‘0’ edges on the left and the ‘1’ edges on the right.

3. Research papers – 20 points

This semester we studied 21 research papers in class:

[Mar98] “An overview of backtrack search satisfiability algorithms”

[Mof13] “Place and route for massively parallel hardware-accelerated functional verification”

[Che13] “Optimized out-of-order parallel discrete event simulation using predictions”

[Fan12] “Transformer: a functional-driven cycle-accurate multicore simulator”

[Mad11] “Litmus tests for comparing memory consistency models: how long do they need to be?”

[Qin12] “Automated generation of directed tests for transition coverage in cache coherence protocols”

[Li13] “A hybrid approach for fast and accurate trace signal selection for post-silicon debug”

[Hon10] “QED: Quick Error Detection tests for effective post-silicon validation”

[Adi11] “Threadmill: a post-silicon exerciser for multi-threaded processors”

[Mei07] “Argus: low-cost, comprehensive error detection in simple cores”

[Fou11] “Accelerating microprocessor silicon validation by exposing ISA diversity”

[Zha10] “Fractal coherence: scalably verifiable cache coherence”

[Wil13] “Towards a generic verification methodology for system models”

[Cha13] “Hybrid checking for microarchitectural validation of processor designs on acceleration platform

[Wag07] “Engineering trust with semantic guardians”

[Aus99] “DIVA: a reliable substrate for deep submicron microarchitecture design”

[Abd11] “Functional correctness for CMP interconnects” (SafeNoC)

[Par11] “Formally enhanced runtime verification to ensure NoC functional correctness”

[Deo09] “Dacota: post-silicon validation of the memory subsystem in multi-core designs”

[Wag08] “Reversi: Post-Silicon Validation System for Modern Microprocessors”

[Abd12] “Functional Post-Silicon Diagnosis and Debug for Networks-on-Chip”

A. [4 pts] Which of the papers above are runtime verification solutions? (simply list the acronym, e.g. [Ber13]):

Papers: ____

B. [1 pt] Think of the paper that you presented in class. Which is it? _____________

C. [3 pts] What problem does the paper you presented strive to solve? Describe the problem in 30 words or less. Below is an example for [Wag07]:
"[Wag07] strives to eliminate bugs escaped into microprocessors released to customers, both escapes known by the manufacturer and those that are yet to be discovered"

Problem your paper strives to solve: **WRITE NEATLY**

D. [6 pts] Connect each paper one the left with one idea on the right by drawing a line (this list only includes papers presented by students):

[Mar98]	4 checkers: control flow, data flow, memory & computation
[Mof13]	verifiable cache design by self-similarity
[Che13]	hypercubes to verify caches
[Fan12]	signal restoration ratio
[Mad11]	equivalent instructions in ISAs
[Qin12]	the Awan compiler
[Li13]	small concurrent programs to analyze memory consistency
[Hon10]	Boolean constant propagation
[Adi11]	on-platform test generation for post-silicon
[Mei07]	reducing error detection latency in post-silicon
[Fou11]	simulating ESL on multicore hosts
[Zha10]	transforming description to UML for verification
[Wil13]	coupling timing and functional models for simulation

E. [6 pts] Summarize the main idea for one of the papers, choosing among [Zha10], [Mei07] or [Fou11]. If you presented one of these papers, you cannot choose that one. Be careful in choosing your words, you need to convey the key idea in 100 words or less.
Below is an example for "Engineering Trust with Semantic Guardians":
"[Wag07]: This is a runtime verification solution where the design states that have not been verified in pre-silicon are encoded in a circuit block (the Semantic Guardian) taped-out alongside the design. At runtime, when an unverified state occurs, the Semantic Guardian triggers a degraded mode of operation for the processor that is guaranteed to operate correctly because it has been formally verified."

Paper you chose: _________

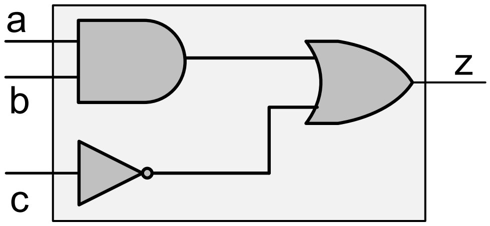
Summary: **WRITE NEATLY**

4. SAT solvers and CNF - 12 points

A. [6 pts] Your company developed a new logic gate for sorting algorithms that they can implement with very little silicon area. As a result, you encounter many of these gates in your design's verification effort. To limit the size of your CNF instances you want to develop a set of clauses that represent this gate as a monolithic block (not as three separate gates).

Express the function the gate computes in clausal normal form. Use the variables as indicated in the diagram, where a,b,c are inputs and z is an output. Your solution should only use variables a, b, c, z, no additional variables. **Show your work to receive credit – correct answers not showing the work will not be credited**.

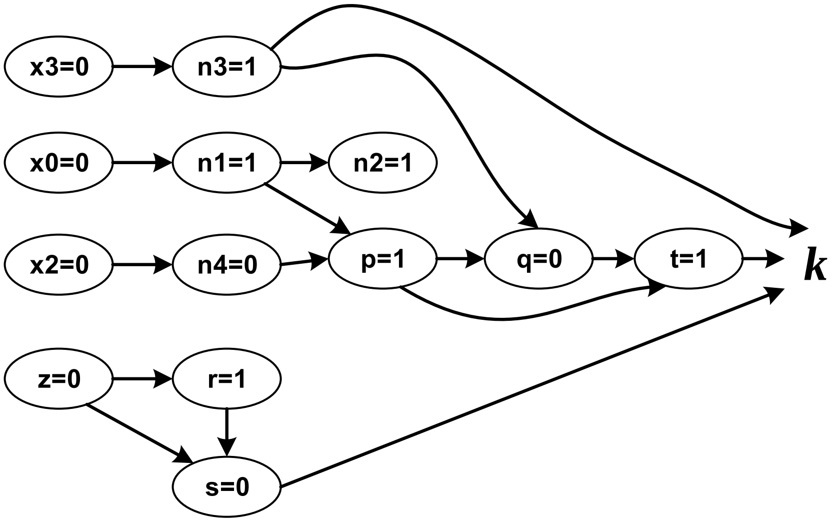
[Be careful in your Boolean logic manipulation! This problem should be quick: use the properties of Boolean algebra!]



Final CNF expression for the special gate:

____

B. [6 pts] Consider the following implication graph generated by a SAT solver in trying to find a satisfiable solution to a CNF instance. In each node we show a decision assignment (omitting the level of the decision) for a variable of the CNF instance.



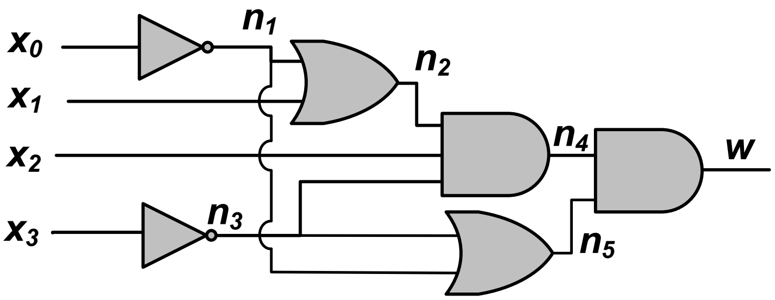
Provide two distinct clauses that could be "learned" from this implication graph. Be careful in choosing the positive or complement version of each literal in your clauses!

Clause 1 ____

Clause 2____

5. Event-driven simulation– 10 points

Consider the circuit below:



At time 0 the input values are all set to 0 and the circuit is stable. The propagation delay of each type of gate is as follows:

3-input AND	12ns
2-input OR, 2-input AND	10ns
NOT	5ns

A. [2 pts] What are the stable values of the internal nodes?

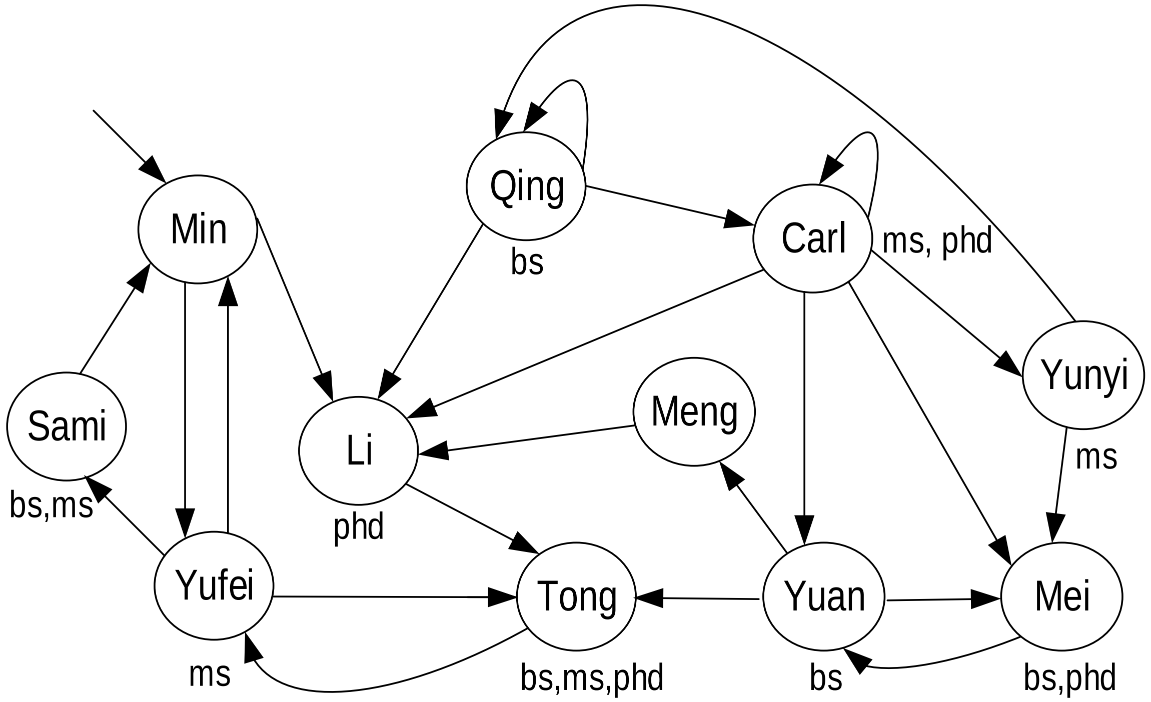
n1 =_____ n2 =_____ n3 =_____ n4 =_____ n5 =_____ w =_____

B. [8 pts] Now simulate the following sequence of events: for each input event indicated, provide the new events that are scheduled and added to the event's queue. Note that you should remove all glitches (see below). Remember that signals may propagate through multiple gates, so one input event may cause adding several events to the queue. Each event should have the form: *signal=value@time*

Glitch = no gate should switch faster than its propagation delay. For instance a NOT gate cannot switch at time 7 and then again at time 10 – since its propagation delay is 5. If a glitch is generated, both the event at time 7 and at time 10 should be removed from the queue.

Input event	Events added to the queue	Events removed from the queue (to cancel glitches)
X2=1@5		
X1=1@8		
X3=1@13		
X0=1@15		
X0=0@16		

6. CTL-based model checking - 20 points



Consider the Kripke structure above where all the atomic propositions are marked next to the states for which they hold. Your goal is to prove or disprove two properties for this structure (indicated in parts D and G). However, we will build up to them by decomposing the CTL formulas into its components and answering questions about them.
[Hint: it may be helpful to track your work by marking the states with the letter corresponding to the subformula on which you are working]

A. [1 pt] List the states for which the property: ***(bs ∧ ms ∧ phd') ∨ (bs ∧ phd)*** holds [Beware: this is a very simple question, but your answer will affect all questions below]

States: ______________________________________

B. [3 pt] List the states for which ***EX ((bs ∧ ms ∧ phd') ∨ (bs ∧ phd))*** holds. [The portion in parenthesis is the same as part A]. For each state in your list, explain why you included the state – group together states with a same explanation.

States: _____________________________________

C. [4 pt] List the states for which ***E((ms ∨phd) U EX ((bs ∧ ms ∧ phd') ∨ (bs ∧ phd)))*** holds. [The right portion of the until statement is the same as part B]. For each state in your list explain why the property holds. You can explain using a few words or by drawing a portion of the computation tree. If a same justification holds for multiple states, group them together.

States: ___________________________________

D. [1 pt] Does the property:
"Li" |= AG (E((ms ∨phd) U EX ((bs ∧ ms ∧ phd') ∨ (bs ∧ phd))))
hold? YES / NO

E. [4 pt] Provide an explanation of why the property in D holds or does not hold. Use <15 words.

Explanation: ____

F. [2 pt] List the states for which ***ms → EX ((bs ∧ ms ∧ phd') ∨ (bs ∧ phd))*** holds. [The portion on the right of the implication is the same as part B]. For each state in your list, explain why you included the state – group together states with a same explanation.

States: ___________________________________

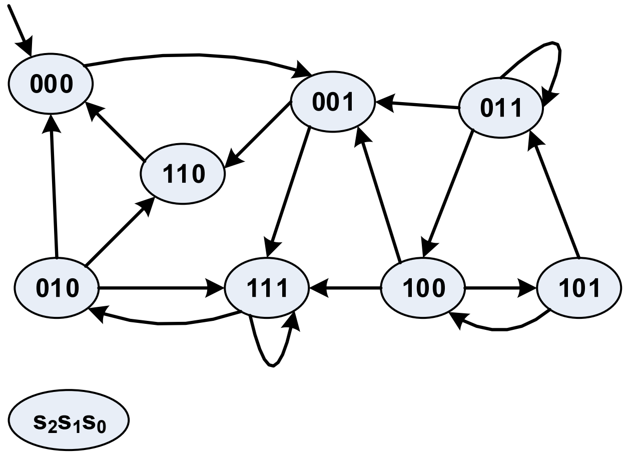
G. [1 pt] Does the property:
"Yuan" |= EG (ms → EX ((bs ∧ ms ∧ phd') ∨ (bs ∧ phd)))
hold? YES / NO

H. [4 pt] Provide an explanation of why the property in G holds or does not hold. Use <15 words.

Explanation: ____

7. Reachability analysis – 18 points

Consider the finite state machine represented in the diagram below: it represents the behavior of a circuit with three flip-flops: s2, s1 and s0. The initial state is <000>.



Below are questions related to a number of sets that are computed in the process of performing reachability analysis on this machine. Please provide your answers by listing each state in the set as <xyz>.

Answer the following questions:

A. [1 pt] What is the Reached set at time 0 (initial state)?

Reached$_{t=0}$=______________________

B. [1 pt] What is the Image set at time 1 (first clock cycle)?

Image$_{t=1}$=________________________

C. [1 pt] What is the Frontier set at time 2?

Frontier$_{t=2}$=_______________________

D. [1 pt] What is the Frontier set at time 3?

Frontier$_{t=3}$=________________________

E. [1 pt] How many steps (cycles) of reachability analysis does it take to reach a fixpoint AND also realize that you have indeed reached a fixpoint?

Steps to fixpoint: _________________________

F. [2 pts] What is the final reached set of this FSM?

Reached$_{final}$=_________________________

G. [2 pts] Below, draw the BDD of Frontier$_{t=3}$. Use the variable order s2, s1, s0 from top to bottom.

H. [1 pt] Is the following property true?
" State 101 can eventually be reached from the initial state"

TRUE / FALSE

Now, assume that this machine is too complex for the reachability analysis to complete. You will need to abstract the machine and do your analysis on this abstract machine:

I. [4 pts] Abstract the flip-flop s2 and draw the new machine. Be careful in this process, it may be helpful for you to cross-out s2 in the original FSM, to visualize the new states.

J. [2 pt] Map the property in H to this new FSM with only two flip-flops (s1 and s0). What should be the new abstracted property description?

"____"

K. [1 pt] What is the Reached set in the abstract machine?

Reached$_{final}$=________________________

L. [1 pt] Does the "abstracted property" hold in the "abstract machine"? (is it true or false)

TRUE / FALSE