

Home Quiz #2

Assigned: 11/04/2015 – Due: 11/09/2015 6PM

Name: _____ unickname: _____

How to submit this assignment:

- (1) email your answer sheet (softcopy) to doowon@umich.edu, or
- (2) bring your answer sheet (hardcopy) to BBB 2765.

Paper: “DIVA: a reliable substrate for deep submicron microarchitecture design” by T. Austin, MICRO '99

Question 1 (2pts each). For each sentence, write “T” if it is true, or “F” if false.

- (a) ____: DIVA is a **design-time verification** methodology that checks the correctness of a microprocessor before its product release, but it is not a runtime verification methodology.
- (b) ____: DIVA can be used to validate **electrical characteristics** such as voltage and temperature dependence.
- (c) ____: In the DIVA checker, a watchdog timer can detect deadlock situations (*i.e.*, situations where no instruction completes for a long period of time).
- (d) ____: The DIVA checker is **tightly coupled** with its core under verification, so it does not stall the core even when there is **only a small buffer** between the core and the DIVA checker.

Question 2 (1pt each). Which of the following statements below is correct about the proposed DIVA checker architecture? Choose all that apply.

- a. The CHKcomp pipeline in the checker verifies the integrity of functional units.
- b. The CHKcomm pipeline in the checker reads again the register file and the data cache to detect incorrect input operands.
- c. The CHKcomp can detect a malfunctioning bypass logic in the pipeline.
- d. When recovering from an error, the execution result from a DIVA checker is used instead of its core under verification.

Question 3 (1pt each). For each sentence, choose the most appropriate term among those listed in the square brackets.

- (a) The DIVA checker verifies the execution result at the [**fetch / decode / issue / execution / commit**] pipeline stage.
- (b) The DIVA checker is implemented usually with an [**in-order / out-of-order**] pipeline.
- (c) If there is no read port in array structures (*i.e.*, both register files and caches) dedicated to the DIVA checker, DIVA’s performance overhead is [**more than 10% / less than 10%**].
- (d) When a DIVA checker is operating 4 times slower than its core under verification, its performance overhead is [**more than 5% / less than 5%**].

Paper: “Ultra low-cost defect protection for microprocessor pipelines” by S. Shyam, K. Constantinides, S. Phadke, V. Bertacco and T. Austin, ASPLOS '06

Question 1 (2pts each). For each sentence, write “T” if it is true, or “F” if false.

- (a) ____: The proposed solution adopts testing mechanisms from built-in self-test (BIST).
(b) ____: The proposed solution targets only transient faults, but not permanent faults.

Question 2 (1pt each). Connect each pipeline component on the left with one of the testing solutions on the right with a line.

- | [left side] | [right side] |
|---------------------|--|
| (a) Decoder · | · using multi-cycle execution of a smaller bit-width component |
| (b) Register file · | · using random test inputs |
| (c) ALU · | · using a majority circuit |
| (d) Multiplier · | · using modulo operations |
| (e) Cache line · | · using a parity bit |
| | · comparing signatures |

Question 3 (1pt each). Which of the following statements below is correct about the proposed checkpointing mechanism? Choose all that apply.

- a. The proposed solution records any register and memory updates not only in the last epoch, but also the epoch before the last one. (*i.e.*, it records updates in the last two epochs.)
 b. At the beginning of each computation epoch, it always copies **all data** in the original register file into a backup register file, entailing some performance penalty.
 c. Each cache line is augmented with a volatile bit indicating whether its data is speculative.

Question 4 (1pt each). For each sentence, choose the most appropriate term among those listed in the square brackets.

- (a) [**A victim cache / Writeback buffers**] can provide longer epochs.
(b) The test clock is set to be [**slower / faster**] than the main clock to detect wearout-related failures.
(c) The performance overhead due to defect-testing is [**less than 1% / more than 1%**] in the experimental results.
(d) The proposed testing technique for register files [**interrupts / does not interrupt**] their normal operation.

Paper: “Post-silicon validation of multiprocessor memory consistency” by B. Mammo, V. Bertacco, A. DeOrio and I. Wagner, TCAD ’15

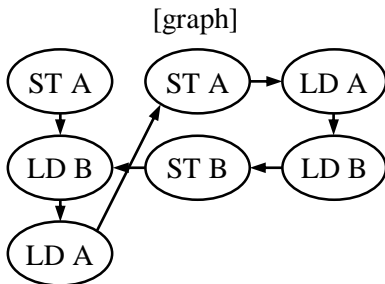
Question 1 (2pts each). For each sentence, write “T” if it is true, or “F” if false.

- (a) ____: The proposed validation method can detect a **bug in cache coherence**, as well as one in memory consistency.
- (b) ____: Log analysis is performed in a **distributed manner**, where each core analyzes its own log.
- (c) ____: Each memory operation is tagged with both a sequence ID and a store counter.

Question 2 (1pt each). Which of the following statements below is correct about the proposed memory-access tracking technique? Choose all that apply.

- a. A fence-counter value indicates the **dependence order** of a memory access.
- b. Each core (or thread) uses a **dedicated fence counter** that is not shared with other cores.
- c. It uses a portion of **L1 data cache** as a log storage.
- d. A log-delay buffer can be used to delay logging of load-accesses, until their depending store-access successfully updates its store counter.

Question 3 (3pts). The graph on the left side is derived from the execution logs of the 3-thread program on the right side. For this graph, which set of (A), (B), (C) and (D) values below best describes the logs on the right side? Note that the fences (e.g., ST → LD) are not part of the original format described in the paper, but shown for your reference. Choose one.



	[logs]	
** log 1 **	** log 2 **	** log 3 **
ST A, {0, 0x0}, (A)	ST A, {0, 0x0}, (C)	LD A, {0, 0x0}, (D)
(ST → LD)	ST B, {0, 0x0}, 1	(LD → LD)
LD B, {1, 0x2}, 1		LD B, {1, 0x1}, 0
(LD → LD)		
LD A, {2, 0x1}, (B)		

- a. (A) 0, (B) 1, (C) 1, (D) 2
- b. (A) 1, (B) 0, (C) 2, (D) 2
- c. (A) 1, (B) 1, (C) 2, (D) 2
- d. (A) 1, (B) 2, (C) 2, (D) 3
- e. None of the above

Question 4 (2pts). Does the graph in Question 3 **violate** any memory ordering rule?

- a. Yes.
- b. No.

Question 5 (1pt each). For each sentence, choose the most appropriate term among those listed in the square brackets.

- (a) The execution overhead [**increases / decreases**] as the size of a log-storage increases.
- (b) Log-analysis time varies [**greatly (more than 50%) / minimally (less than 50%)**] depending on the size of a log-storage.

Paper: “PipeCheck: specifying and verifying microarchitectural enforcement of memory consistency models” by D. Lustig, M. Pellauer and M. Martonosi, MICRO ’14

Question 1 (2pts each). For each sentence, write “T” if it is true, or “F” if false.

- (a) ____: PipeCheck takes a formal approach that enumerates all possible execution paths.
- (b) ____: Preserved program order (PPO) is a set of reordering rules specifying which reordering must not occur.
- (c) ____: PipeCheck considers PPO as a proposition to be verified, but not as an assumption that has already been verified.
- (d) ____: Even though an observed memory ordering is forbidden in an architectural-level analysis, it can be permitted in a microarchitectural-level analysis.
- (e) ____: A litmus test fails when an observed memory ordering is forbidden under the rules allowed in the deployed memory consistency model.

Question 2 (1pt each). In the “microarchitecturally happens before (μhb)” graph, which of the following edges is constructed with static information (*i.e.*, without executing a program)? Choose all that apply.

- a. Intra-instruction edge
- b. Intra-location edge
- c. Write-serialization edge
- d. PPO edge

Question 3 (2pts). Which of the following pipeline stages below is the performing location of a store with respect to remote cores? Please follow the pipeline-stage notations described in the paper. Choose one.

- a. Memory stage
- b. Store buffer stage
- c. Memory hierarchy stage

Question 4 (1pt each). For each sentence, choose the most appropriate term among those listed in the square brackets.

- (a) When a load gets its data from a preceding store in the same core, there is a “reads-from” (*rf*) edge starting at the [**memory stage / memory hierarchy stage**] of the store.
- (b) Non-local ordering edges define orderings between different instructions [**at a single microarchitectural location / at different microarchitectural locations**].
- (c) The proposed technique took [**more / less**] than 1 second on average to verify the memory consistency model in the OpenSPARC T2 processor.