ECE 753 - FAULT-TOLERANT COMPUTING
(Spring 2013-14)

PROBLEM SET 5

Date: April 3, 2014 Thursday
Due Date: April 10, 2014 Thursday

Please write your answers legibly, be brief, and to the point. Preferably submit a printed copy of your solution. Some of the problems may be same or similar to the past offering of the course and I strongly urge you to do the problems yourself instead of relying on some solution that may be available from past offering. Some of the problems will be graded for submission credit only, but you must do all problems assuming that they will be graded for completeness and correctness.

1. **(10 points)** Watchdog techniques
   
   (a) Explain briefly how the path signature method discussed in class can reduce the number of signatures.

   (b) Explain the working of branch address hashing and how it reduces the memory overhead

2. **(10 points)** Critical value forwarding
   
   Follow the reference to “fingerprint” in the paper *Energy efficient fault tolerance in chip microprocessors using critical value forwarding* by Subramanyan et. al, in IEEE DSN, June 2010, and describe very briefly in your own words, in no more than one page, “fingerprint” and its possible “implementations.”

3. **(20 points)** Consider a system with the simple control flow shown in Figure 1. The worst case execution time for each block is shown inside of each block. Determine statically where should you place checkpoints, for the following two cases, assuming that on occurrence of a failure the process must be able to return to the same point in 8 units of time. You can sub-divide the blocks into smaller units. You should use as few checkpoints as possible.

   (a) One of the checkpoint is taken at the entry point of the block that takes 10 units to execute.

   (b) One of the checkpoint is taken just before the branch point.
4. (20 points) Problem 3 on page 224-225 of the text.

5. (20 points) Problem 7 on page 226 of the text.

6. (20 points) A system of four processors takes un-coordinated checkpoints as shown in Figure 2.

(a) Determine the recovery line if process P2 fails at the point shown. Note that some of the processes do not have to rollback.
(b) Determine a fault in one of the executions, closest to the present time, which will cause all the processes to rollback to at least previous checkpoint. Note that some processes may have to rollback even more. Draw the corresponding recovery line.