EE 382C/361C: Multicore Computing **Fall 2016**

Lecture 24: November 17th

Lecturer: Vijay Garg *Scribe: Hang GONG*

24.1 Introduction

Some questions for Chapter 5.

Q1

In SRSW multi-value register, how do we get the correct value?

A. do a single forward scan until we find a true bit

B. do a single backward scan until we find a true bit

C. first do a forward scan, then do a backward scan

Answer: C

Q2

why this code is considered safe instead of regular? (In which scenario it is not regular?)

```
1 class SafeBoolean {
2     boolean value ;
3     public boolean getValue () {
4         return value ;
5     }
6     public void setValue (bolean b) {
7         value = b;
8     }
9 }
```

Answer: If setValue and getValue are invoked concurrently, the value being written is the same as the previous value and getValue returns a different value.

Q3

Which register we use a communication matrix for the construction?

A. SRSW multi-value register

B. MRSW multi-value register

C. MRMW multi-value register

Answer: B

Q4

Which register we use a technique like Lamport's Bakery Algorithm for the construction?

A. SRSW multi-value register

B. MRSW multi-value register

C. MRMW multi-value register

Answer: C

Q5

In a *safe* register, if the initial value is 0, and one read operation overlaps with one write(3) (write 3 to the register) operation. Which value can be possibly returned by the read? (multiple choice)

A. 0

B. 3

C. 100

Answer: ABC (if read overlaps with write, a safe register can return **anything**)

Q6

In a *regular* register, if the initial value is 0, and one read operation overlaps with one write(3) (write 3 to the register) operation. Which value can be possibly returned by the read? (multiple choice)

A. 0

B. 3

C. 100

Answer: AB (if read overlaps with write, a *regular* register can return either the value of the most recent write that preceded the read or the value of one of the overlapping writes.)

Q7

When we do a write operation in SRSW multi-valued register, when we do a write operation, how do we update the array bits that preceding it?

A. In a backward direction, update the array entries from here to zero

B. In a forward direction, update the array from zero to here

C. Never mind, either will be OK

Answer: A

References

[1] V.K. Garg, Introduction to Multicore Computing