2008
YEAR
PRESENTATION
The University of Auckland | New Zealand

Computer Science 703

Advance Computer Architecture

2006 Semester 1

Lecture Notes 5

20Mar08

Atomic RMW Primitives

James Goodman

Department
of
Computer Science

Atomic Read-Modify-Write Instructions

- Test&Set
- Test&Test&Set
- Load&Clear
- Compare&Swap
- Load_Linked/Store_Conditional
- Oklahoma Update
- Lock (Hardware prefix)
- Fetch&Increment

Test & Set

Instruction form

```
tas       $Result, EA
```

Execute two instructions atomically:

```
ld        $Result, EA
st        0x1, EA
```

`$Result` might be a testable flag:

```
tas       EA
```

Compare & Swap

Instruction form

```
            cas    $Result, $CMP, EA
```

Execute atomically:

```
            ld     $Result, EA
            cmp    $Result, $CMP
            bneq   Cont
            swap   $Result, EA
Cont:       …
```

Acquiring a Lock

Using T&S

```
tas     $t1, Lock       / Set Lock to HELD ($t1)
bnez    $t1, Failure    / Was Lock already HELD?
b       Acquired
```

Using T&S

```
cas     $0, $t1, Lock   / If Lock is FREE, set to $t1
bne     $0, $t1, Failure  / Was Lock already HELD?
b       Acquired
```

Comparing T&S, CAS

- Are the equivalent?
- Which is easier to implement?
- What can be accomplished by CAS that cannot be accomplished by T&S?