Full Name:
Andrew Id:

15-418/618
Exercise 4
Answer Sheet

Problem 1: Lock Implementations

A. Direct.

CAS direct always tries to write the lock and invalidates the cache in other processors.

B. Test.

CAS test only tries to write and invalidates after it detects the lock is 0. Before that, it only reads and don't invalidates. So it would be faster than CAS direct.

C. Backoff.

When a thread is in the delay, it won't even try to read. Thus it can be faster than CAS test.

Problem 2: Memory Consistency

A. Explain why these two fence operations have such different relative performance.

As the number of threads grows, the contention over the global variable becomes more severe. However, the contends over the local variable still don't exist. So the differece between the two Implementations grows larger.

B. Explain why the local version has performance that improves with the number of threads.

The operation over local variable can all be paralleled. So with more threads, the time shared by per operation is less. The graph has an almost perfect speedup ($7.62\times$ for 8 threads). But the graph is not linear. The speedup is linear only means the proportion between two time is linear.

C. For each of the five functions, determine where fence operations must be inserted to guarantee the ordering properties. You should insert only a minimum set of fences. Justify why these particular fences are required and why no others are.

A fence should be used after the `cas_lock_init` to ensure no write has happened before the write has done.
A fence should be used before the `cas_lock_unlock` to ensure all write operations has finished before unlocking the lock.
No need for fences in the lock functions.

Problem 3: Interconnection Networks

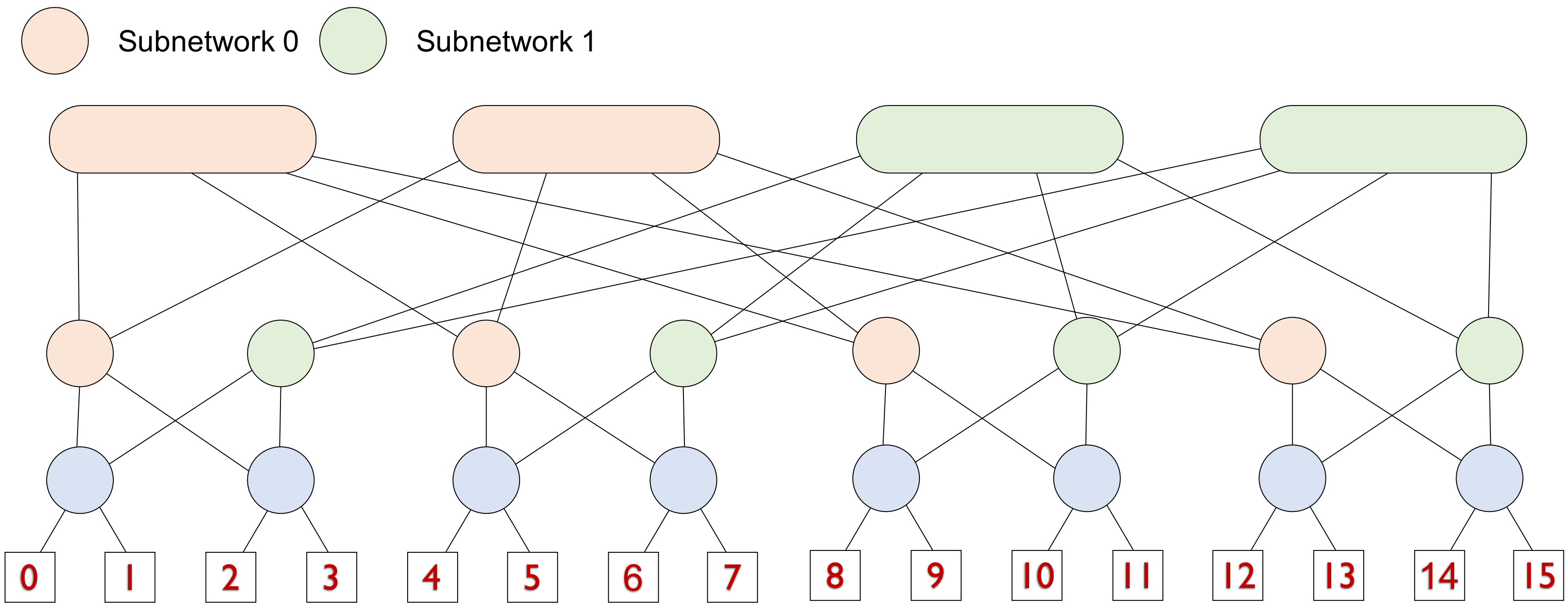

Figure 1: Fat-tree network, showing the recursive structure

A. Identify the $k/2$ subnetworks of type $N(k, 2)$ for $k = 4$ in Figure 1. You can do this by modifying the diagram in Figure 1. Use different colors for the switches to indicate the different subnetworks and the additional switches.

B. Derive a closed-form formula for $P(k, l)$.

$P(k, 1) = k$
$P(k, l) = P(k, l - 1) \cdot \frac{k}{2}$
Hence, $P(k, l) = P(k, 1) \cdot (\frac{k}{2})^{(l-1)} = 2(\frac{k}{2})^l$

C. Show that you could set up the eight links forming a *mirror permutation*, mapping port i to port $N-i-1$ for $0 \leq i < N/2$. You can do this by modifying the diagram in Figure 2. Use different colors to illustrate the different links.

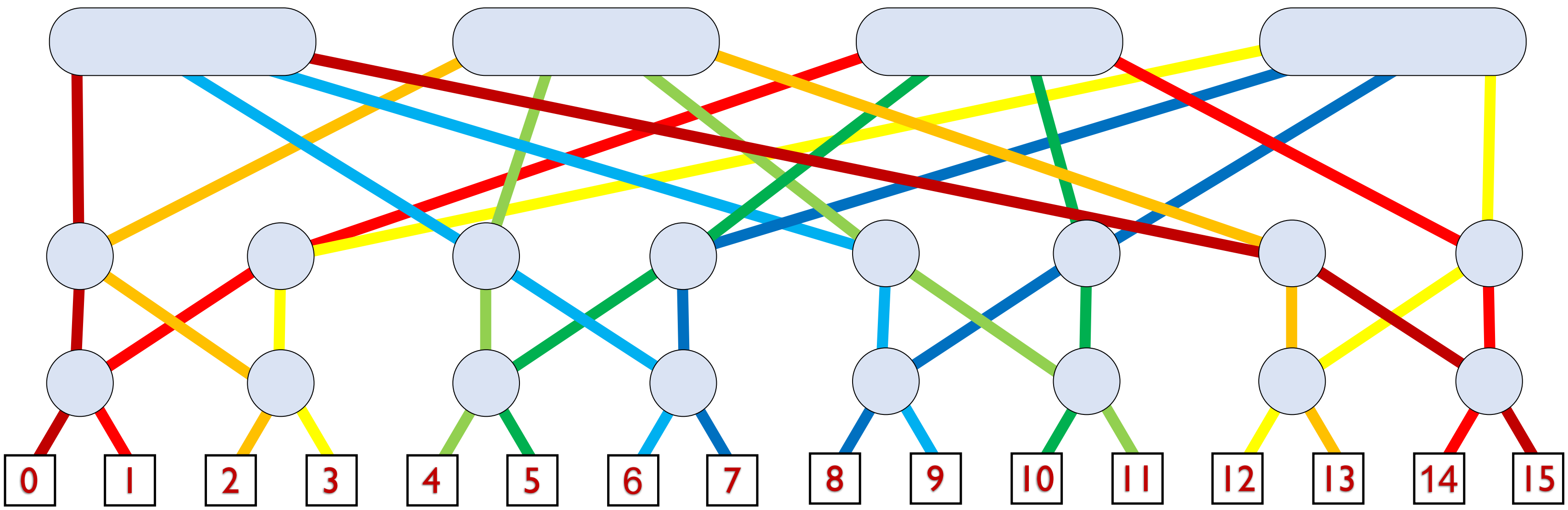

Figure 2: Fat-tree network with links for mirror permutation