Deadlock & Paper Discussions

Tuesday, February 23, 2010
10:23

Deadlock

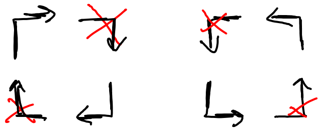
→ cyclical dependency of queue waiting

⇒ unable to make forward progress

→ we can use virtual channels to help, but only if we restrict their usage and leave them open to help w/ deadlock

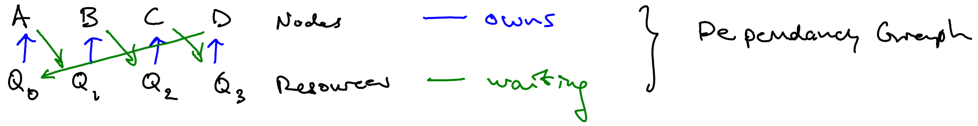
→ dimension-order routing is deadlock-free, but unnecessarily restrictive in its routes

⇒ we can use the "Turn Model" to prevent deadlock instead, which only removes the necessary links, but it's way more complicated

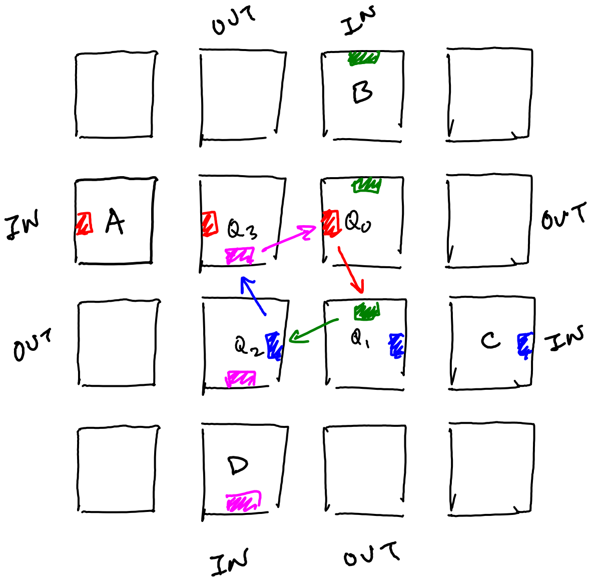


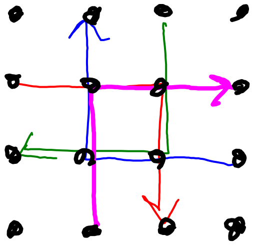
need to break the cycles by removing one or more links.

Example



Desired Source/Destination Routing Paths





↖ Note the cycle in the middle of the network.

Each message is waiting for someone else's queue.

The dependency graph (shown above) shows a cyclical dependency

⇒ Deadlock!

OITURN Routing

- → OITURN is similar to ROMM and DOR
  - → more flexible than DOR
  - → less flexible than ROMM

  - → They are better in worst-case

- → OITURN makes sense
  - → passed the intuition test
    - → If it's intuitive, the analysis should confirm your intuition
    - → If not, it had better be really convincing

  - → Just use it! Unless you can't deal with out of order packet delivery.
    - → due to path diversity, etc.
  - → But

    - → How often does the worst case happen?
    - → Is that the best metric?

    - → They make a big deal about this, but what they should say is they are competitive with ROMM in the average case, with simpler router, and better worst-case performance.

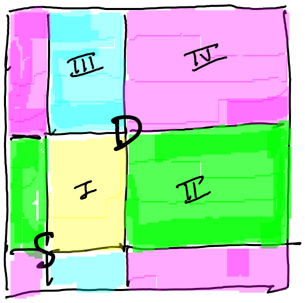
- → OITURN Router Implementation

  - → Use VC's to get deadlock-free behavior like DOR

GOAL Routing Algorithm

→ Globally Oblivious Adaptive Locally

→ Non-minimal route for torus networks.



We know Source & Destination, which divide the 2-dimension torus into 4 quadrants. Weighted-random choose a quadrant and route minimally in that quadrant.

- → Decreases throughput sometimes, as the randomness introduces contention.
  - → plus it's non minimal