Predictable On-Chip Interconnects

A) Discuss the key characteristics of on-chip interconnects (both for general purpose systems, as well as for real-time systems).
-- What are the advantages and disadvantages between buses, crossbars and NoC architectures?
-- What are the key differences between off-chip and on-chip networks? How do they affect on-chip interconnect design?
-- How does wormhole routing work? Why it is beneficial?

B) Consider the three systems described below. For each system, discuss which on-chip interconnect architecture and arbitration scheme you feel would be most appropriate, based on considerations of performance, predictability, cost and/or power. Refer to the examples seen during Lecture 5 as needed.
-- 1. A multicore avionics system based on Integrated Modular Avionics. The system comprises multiple cores, each with private last level (LVL2) cache, and a unique main memory device with a single communication port to the interconnect. Cache coherency is not required; each software partition (i.e., a real-time virtual machine) runs on a single core and does not communicate with other partitions. It is desirable to verify and certify partitions in isolations (i.e., without running the other partitions at the same time), hence interference among partitions should be minimized.
-- 2. A heterogeneous multicore system used to implement a complex data flow application (an example could be a software defined radio, or any other data and process-intensive real-time application, such as real-time video processing). The platform has multiple I/O devices and memory devices. Nodes in the data flow represent tasks that are statically assigned to cores, and communicate with I/O devices and with each other through memory. The data flow is statically scheduled, i.e., each task is time-triggered such that it starts after previous tasks in the data flow have finished executing and sending their data. The system designer wishes to be able to easily implement the static system schedule and verify its correctness.
-- 3. A soft real-time system running on a large multicore under a modified version of Linux. The system workload includes soft real-time applications that are sensitive to delay (ex: video encoding/decoding, teleconference), and other non-real-time, background applications that are insensitive to delay, but still require a given average bandwidth to progress smoothly. All applications are I/O and data intensive, and the platform supports multiple I/O devices and memory controllers. We wish to provide differentiated quality-of-service to different applications, such that the user experience is improved but background applications still perform useful work.

C.1) **(C.1 and C.2 are exclusive. You might receive a similar question if your focus is on hardware/architecture)**. Consider the following NoC using

wormhole routing: the NoC is composed of switches with 5 input and 5 output ports each, arranged in a mesh configuration; 4 ports are used to communicate with neightbouring switches, and one port with the core attached to the switch. All links between switches are uni-directional; a 64-bits flit is transmitted every clock cycle. Whenever a source core wants to send a packet to a destination core, it encodes the packet route in the header flit of the packet. As the header flit propagates through the network, each switch processes it and uses the information within the header to set up the encoded route (connection). The last flit in the packet is used to tear down the connection. For simplicity, assume that some other mechanism (ex: synchronization and agreement among cores) is used to ensure that the network is contentionless, hence you do not have to worry about arbitration for access to output ports or virtual channels.
-- Sketch a possible bit format for the header flit and a block diagram for the switch hardware. Feel free to draw only a single input and output port, but make sure to include required cross-bars and/or multiplexers/demultiplexers.

C.2) **(C.1 and C.2 are exclusive. You might receive a similar question if your focus is on software/analysis)**. Consider a mesh NoC with bidirectional links (i.e., each link between two switches is shared in the two directions). Which of the four end-to-end analyses we saw in class (Network Calculus, Holystic Analyis, Delay Calculus, Flow-Level Analysis) can be applied to compute delays on the NoC in this case? Which cannot? How would they perform in the case of packets moving on the same route but in opposite directions? Clearly motivate your answer.

Evaluation: You will be evaluated based on the technical correctness, insight and depth of your answers. There is no space limit; try to be concise, but make sure to touch on all key points covered in class/reading list and relevant to the questions. The exam is designed to be completed in 4-6 hours + additional research time. If you have questions, please email the instructor (best during the morning to ensure a quick reply). You are free to use whatever material you deem useful, but you must refrain from discussing the problem with other people. Please send your answers by email in pdf format to the instructor by 11:59PM tonight.