Homework 02 Solutions

ECE 587, Spring 2026

1. *(1 point)*

   A hardware device is required to read a sensor every 500ns and to report the readings immediately over a network to a server. Show that the Gigabit Ethernet is actually too slow for such purpose. (Hint: check `https://en.wikipedia.org/wiki/Ethernet_frame` for minimum Ethernet frame size.)

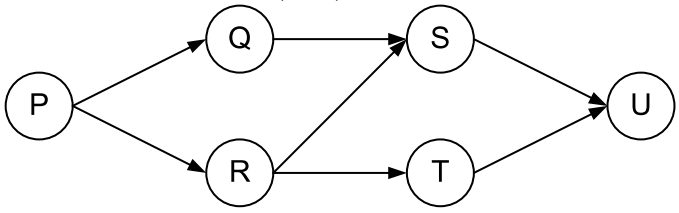
   Answer:

   The hardware device would need to send 1s/500ns=2,000,000 packets per second.

   However, ethernet frame has a minimum size of 64 bytes or 512 bits. Therefore, one can send at most 1,000,000,000/512=1,953,125 packets per second on Gigabit Ethernet, which is slower than what is required.

2. *(2 points)*

   Consider the application modeled by the dataflow graph below where the output is obtained by executing each of the actors once (P, Q, R, S, T, and U) following the data dependency. Assume each activity can be either implemented as a program running on a software processor (SW) or as a hardware component (HW).



   The time to finish each actor (in clock cycles) by HW or SW and the HW cost are listed below:

Actor	SW Time	HW Time	HW Cost
P	10	5	10
Q	100	5	100
R	20	10	10
S	20	10	10
T	100	5	100
U	10	5	10

We further use the following assumptions to simplify the synthesis process.

- There should be exactly one SW processor in the system platform so that the costs of SW implementations could be ignored. The SW processor executes one actor at a time and the scheduling overhead is ignored.
- HW components do NOT share resources so that their costs add up directly.
- Performance and cost overheads of communications are ignored.

1) If all the actors are mapped to the SW processor, how many cycles are required to complete the computation?
2) If all the activities are mapped to HW components, how many cycles are required to complete the computation? What is the cost of the system?

Answer:

For 1), actors need to executor sequentially on the SW processor. So we need 10+100+20+20+100+10=260 cycles.

For 2), actors can run parallelly as long as there is no dependency. So the performance is limited by the critical path PRSU and we need 30 cycles. The cost is 10+100+10+10+100+10=240.

3. *(2 points)*

Once upon a time a farmer went to a market and purchased a fox, a goose, and a bag of beans. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the fox, the goose, or the bag of beans.

If left unattended together, the fox would eat the goose, or the goose would eat the beans. Will the farmer be able to carry himself and his purchases to the far bank of the river, leaving each purchase intact?

While you may easily find the answer to this problem online, we are interested in a method that solves this and similar river crossing puzzles. Show that you can model the problem with a FSM and then apply model checking concepts to solve it.

Answer:

First of all, the states of the problem can be encoded by 4 bits, one each for the farmer, the fox, the goose, and the bag of beans. For each bit, 0 represents one bank and 1 represents the other.

Therefore, the initial state is 0000 where all are at one bank and the final state is 1111 where all are at the other bank.

The state transitions represent the action of the farmer crossing the river with at most one of his purchases. So if the farmer first crosses the river with the beans, the state transition is from 0000 to 1001.

However, the state 1001 is not safe because the fox will eat the goose. Therefore, our goal is to find a path from 0000 to 1111 where all states along the path are safe, or to prove such a path does not exist. This is where the model checking concepts apply.

To find the path, we annotate the states as safe or not, and start a breath-first search from 0000. The safe path to 1111 then can be found as 0000, 1010, 0010, 1110, 0100, 1101, 0101, 1111.