UNIVERSITY OF TEXAS AT AUSTIN
Dept. of Electrical and Computer Engineering

Mid-Term #1

Date: March 6, 2002 Course: EE 382C-9

Name: ____
Last, First

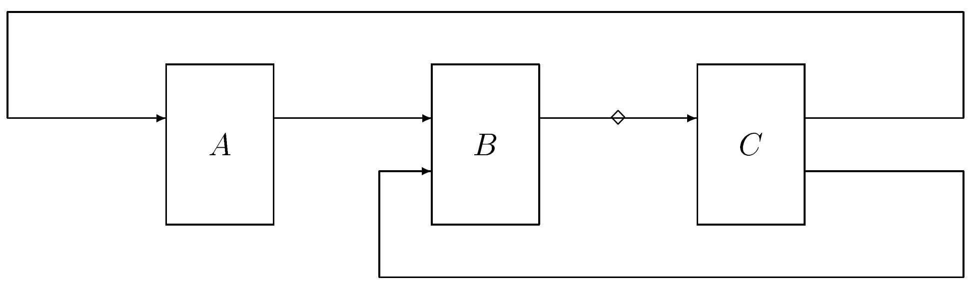
- The exam will last 90 minutes.
- Open book, open notes.
- Calculators are allowed.
- You may use any standalone computer system, i.e., one that is not connected to a network.
- All work should be performed on the midterm itself. If more space is needed, then use the backs of the pages.
- Fully justify your answers.

Problem	Point Value	Your Score	Topic
1	20		Synchronous Dataflow Scheduling
2	20		Clustering
3	20		Process Networks
4	20		Data Converter
5	20		Potpourri
Total	100		

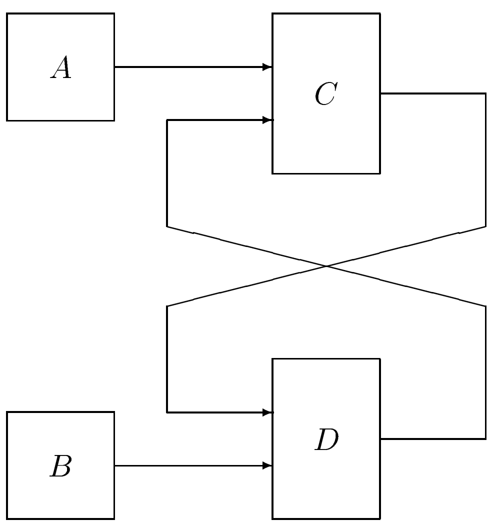
Problem 1.1 Synchronous Dataflow Scheduling. 20 points.

For each of the two Synchronous Dataflow (SDF) graphs given below, indicate whether or not the graph is sample rate consistent (i.e., has a periodic schedule) and has a valid periodic schedule. Give a periodic schedule if one exists.

(a) During execution, A consumes 1 token, A produces 1 token, B consumes 2 tokens on the upper input, B consumes 2 tokens on the lower input, B produces 2 tokens, C consumes 1 token, C produces 1 token on the upper output, and C produces 2 tokens on the lower output. The delay between B and C is 2 tokens.

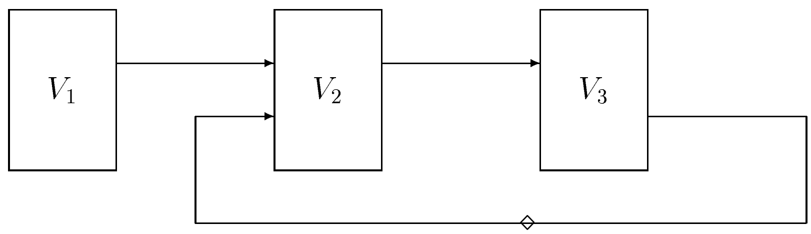


(b) A produces 1 token, and B produces 2 tokens. C consumes 2 tokens on its upper input. D consumes 4 tokens on its lower input. All other input consumption and output production is 1 token.



Problem 1.2 Clustering Synchronous Dataflow Graphs. 20 points.

Show below is a Synchronous Dataflow graph. The $\diamond$ represents one sample of delay. During execution, actor V_1 produces 2 tokens, and actors V_2 and V_3 produce 1 token. Actors V_2 and V_3 consume one token on their inputs.



(a) Determine all the possible admissible periodic schedules for this graph. 7 points.

(b) Which pairs (if any) of connected blocks can be clustered into one superblock and give a valid Synchronous Dataflow graph consisting of the superblock and the remaining block? 7 points.

(c) In the original graph above, let block V_1 read the value from variable R in a global workspace and write its value twice to its output. In addition, let block V_2 read the value of its lower input and write it to the variable R in the global workspace. Is the execution of the graph determinate? 6 points.

Problem 1.3 Process Networks. 20 points.

A Process Network program is a set of processes that communicate through a network of infinite first-in first-out (FIFO) queues also know as channels. In general, it takes infinite time to decide if a Process Network program can be scheduled in bounded memory. It also, in general, takes infinite time to decide if a Process Network program will terminate (halt).

Tom Parks, in his thesis, suggested the following algorithm to schedule Process Network programs:

I. Set a capacity on each channel

II. Block a write if the channel is full

III. Repeat the following:

  A. Run until deadlock occurs.

  B. If there are no blocking writes, then stop the execution of the program.

  C. Among the channels that block writes, select the one with the smallest capacity (queue size) and increase its capacity until the producer associated with it can fire.

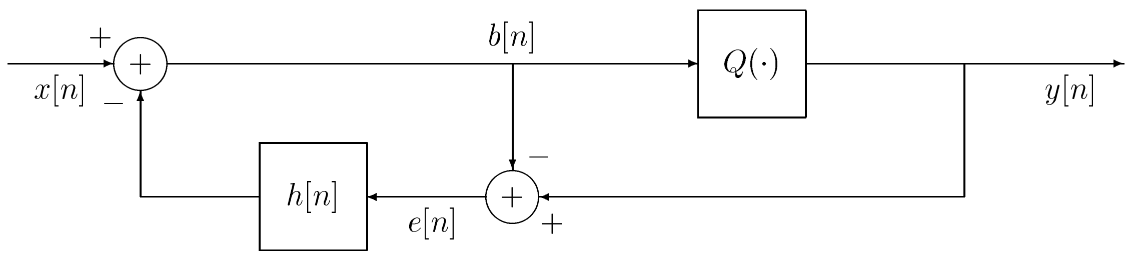
(a) Can this algorithm introduce artificial deadlock? Explain why or why not.

(b) Does this algorithm always execute a Process Network program in bounded memory if a bounded memory implementation exists? Explain why or why not.

(c) Does this algorithm always determine if a Process Network program will halt or not? Explain why or why not.

Problem 1.4 Modeling. 20 points.

The following figure shows a type of sigma-delta modulator. The source (input) $x[n]$ is a stream of eight-bit unsigned integers, and the sink (output) $y[n]$ is a stream of eight-bit unsigned integers of value 0 or 255. $Q(\cdot)$ represents a thresholding quantizer that outputs true (255) or false (0), and $h[n]$ represents a finite impulse response (FIR) filter. Each arc has the same data type (eight-bit unsigned integer). None of the data values on the arcs have time stamps.



(a) Model this system as a valid Synchronous Dataflow graph. Show the production and consumption of tokens on each arc on the above graph. 7 points.

(b) Can the above system be modeled as is using a Process Network model of computation? 7 points.

(c) Compare the efficiency of the runtime performance using a Synchronous Dataflow model vs. using a Process Network model of computation for this system. 6 points.

Problem 1.5 Potpourri. 20 points.

(a) During the execution of a *mutable* system, it is possible change the system's topology. As a mutable system is running, for example, we could delete a node (block) in the graph, or add a node and connect it to another node. Based on the *semantics* of the following models of computation, and *not* their implementations in electronic design automaton tools, indicate whether or not the model of computation supports any form of mutable graphs. If you believe that a model can support mutability, then indicate any special conditions on a system that must exist for mutability (e.g., can only delete nodes, but cannot add any). 10 points.

Model of Computation	Mutable?	If Mutable, List Any Special Conditions
Synchronous Dataflow	______	______________________________
Computation Graphs	______	______________________________
Boolean Dataflow	______	______________________________
Dynamic Dataflow	______	______________________________
Process Network	______	______________________________

(b) Of the above models of computation, which ones always guarantee that a valid program expressed in the model of computation will have a sequential schedule? 3 points.

(c) Define priority inversion in a multithreaded system. State one way to guarantee that priority inversion does not occur. 7 points.