UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences

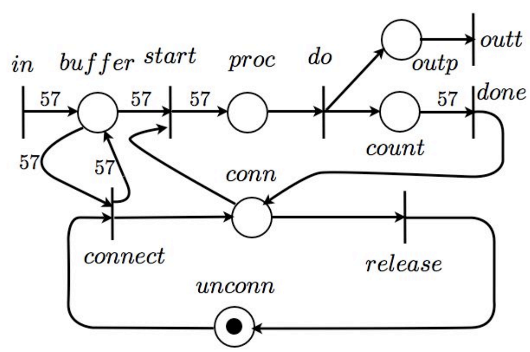
EECS 249 **A. Sangiovanni-Vincentelli**
Fall 2012 **P. Nuzzo**

Homework 3
Due Tuesday, October 30, 2012

General remark: since we will mostly be dealing with design problems throughout this class, an answer to a homework problem may not necessarily be right or wrong. Grade is established mostly based on the reasoning that you follow while developing your solution. Therefore, make sure that you justify all your claims. You can use any kind of sources as long as you include references.

HW3 focuses on: (i) examples of system description, analysis, implementation using Petri nets, Esterel (synchronous reactive model) and the Tagged Signal Model framework; (ii) example of application of the timing analysis techniques seen in class. **We acknowledge Marco Di Natale for helping out with the preparation of this assignment.**

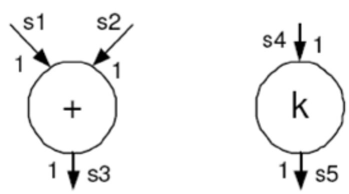
1. **A packet processing sub-system.** There are many ways of transferring data through a network. The network that you are most familiar with, the Internet, is a packet switched connectionless network. It is connectionless in the sense that given a source host and a target host, there is no need to establish a connection through the network to send data from one to the other. In other networks, like the one based on ATM (Asynchronous Transfer Mode), a connection has to be established first. During the connection setup, all the network resources needed to transfer packets from source to destination are reserved for that connection. When the transmission ends, all the resources are released and made available for other connections. The Petri Net below shows the model of a connection-oriented packet processing.



   a) Provide an intuition of how the system works.
   b) Now that you have an idea of what the system is supposed to do, you should be able to apply some of the formal analysis techniques seen in class. Let the transitions be ordered as in the list (*in,start,do,done,outt,release,connect*), and the places be ordered as in the list (*buffer,proc,conn,unconn,count,outp*). Prove that it is not possible to process a packet if the connection has not been established. (Hint: formally, this is equivalent to prove that marking (0,57,0,1,0,0) cannot be reached from the initial marking (0,0,0,1,0,0)).
   c) Determine whether the proposed model is a Free Choice Petri Net.
   d) Now it is time to actually generate code out of the proposed model. Find first a periodic schedule for the network using the method that you like and then write a program in C/C++ (or an equivalent language) that simulates the Petri net model. You can assume that each transition is a task that only consumes and produces tokens.

2. **A traffic-light controller.** Construct an Esterel program that controls a traffic light at the intersection of a busy highway and a farm road according to the following specifications:
   a) Normally, the highway light is green.
   b) If a sensor detects a car on the farm road:
      - The highway light turns yellow then red.
      - The farm road light then turns green until there are no cars or after a long timeout.
      - Then, the farm road light turns yellow then red, and the highway light returns to green.
   c) The program accepts as inputs the car sensor `C`, a short timeout signal `S`, and a long timeout signal `L`.
   d) The program outputs are a timer start signal `R`, and the colors of the highway and farm road lights `HG`, `HY`, `HR`, `FG`, `FY`, and `FR`.

3. **An infinite impulse response filter.** We want use a dataflow model to express an infinite impulse response (IIR) system and represent the resulting model within the Tagged Signal Model framework. In particular, we use the two processes (actors) shown below, where + is an adder and *k* multiplies the input signal by a constant.



Given the following denotational description of the filter

$$y_i = x_i + k \cdot y_{i-1},$$

where *x* is the input sequence of samples and *y* is the output sequence of samples,

   a) Describe the set of values *V* that you are going to use. Write a possible input signal *x*. Describe all possible behaviors of the processes + and *k*.
   b) Using interconnection processes, the two processes + and *k* defined above and the operation of projection, build an IIR first-order filter with input *x* and output *y*.
   c) Describe now the IIR first-order filter without using the projection operator. How many signals does the resulting process have? Describe the set of tags and all the signals of the process.

4. **Automobile engine control.** The following task model represents an example of an engine control application. WCET is the worst-case execution time of each task, periods and WCETs are in ms, and a lower priority index indicates a higher priority. The system consists of Periodic (P) and Sporadic (S) tasks. Sporadic tasks are executed at every revolution of the engine crankshaft (period T) or at the half revolution (T/2).
Modern automobile engines are typically operated at around 2000–3000 revolutions-per-minute (rpm) (33–50 Hz) when cruising, with a minimum (idle) speed around 750–900 rpm (12.5–15 Hz), and an upper limit anywhere from 4500 to 10,000 rpm (75–166 Hz) for a road car or nearly 20,000 rpm for racing engines such as those in Formula 1 cars (currently limited to 18,000 rpm)[1].

[1] Wikipedia

Task	Period	Priority	WCET	Per/Spor
τ1	T/2	1	0.2	S
τ2	T	2	0.3	S
τ3	4	3	0.04	P
τ4	T	4	0.15	S
τ5	T	5	0.1	S
τ6	T	6	0.01	S
τ7	1000	7	1.5	P
τ8	1000	8	5	P
τ9	50	9	1	P
τ10	12	10	0.82	P
τ11	100	11	9.85	P
τ12	1000	12	150	P
τ13	1000	13	100	P
τ14	1000	14	110	P
τ15	1000	15	120	P
τ16	1000	16	131	P

a) Write a program (or Excel script) for computing the worst-case response time of all tasks given the engine speed in rpm. Provide the results for 3000 rpm.

b) Assume the WCETs of all sporadic tasks are doubled. What is the maximum rpm for which all tasks terminate before their next instance (i.e. they are schedulable, assuming that their deadline is D=T)? At 6000 rpm, what is the maximum WCET of τ1 that still guarantees schedulability if all WCETs of the sporadic tasks (except τ1) are doubled?