

# Embedded System Design and Modeling

EE382N.23, Fall 2017

## Homework #2 Models of Computation

Assigned: September 25, 2017

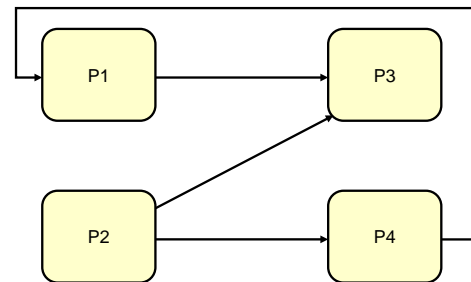
Due: October 9, 2017

### Instructions:

- Please submit your solutions via Canvas. Submissions should include a single PDF with the writeup and a single Zip or Tar archive for any supplementary files (e.g. source files, which has to be compilable by simply running 'make' and should include a README with instructions for running each model).
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.
- Some questions might not have a clearly correct or wrong answer. In general, grading is based on your arguments and reasoning for arriving at a solution.

### Problem 2.1: KPN

Consider the following KPN example, where P3 never consumes any tokens on its P2 queue. For each of the following scheduling strategies, show step by step how the algorithm schedules the sequence of process executions. Is the schedule complete? Bounded? Non-terminating?

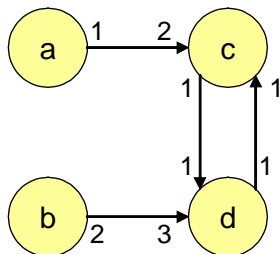


- Data-driven scheduling
- Demand-driven scheduling
- Park's algorithm

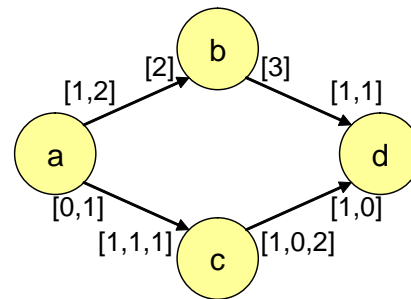
### Problem 2.2: Synchronous Dataflow (SDF)

For each of the following (C)SDF graphs, indicate whether or not the graph is consistent and has a valid periodic schedule. Show the balance equations, the repetition vector, and give a minimal static schedule if one exists.

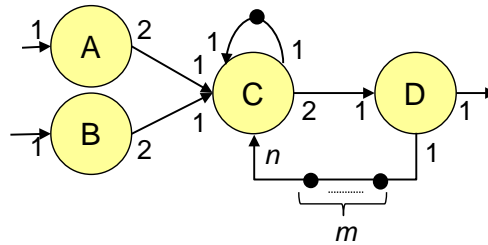
(a)



(b)



- (c) For what integer values of  $m$  and  $n$  is this graph consistent and schedulable? What is the repetition vector and a minimal static schedule when  $n$  is fixed to its minimal value and  $m=2n$ ?



### Problem 2.3: Synchronous/Reactive vs. State Machines

In class we discussed that synchronous/reactive and HCFSM models are really based on the same (synchronous, i.e. lock-step and broadcast) semantics. As such, they can be translated into each other. Given the following synchronous/reactive Esterel program, draw its equivalent HCFSMD model.

```

module M:
input P;
output C: integer;  /// valued signal
signal I in /// local signal
[
  loop await P; emit I; await not P; emit I; end
  ||
  var v:=0: integer in /// local variable, init to zero
    loop await I; v:=v+1; emit C(v); end
  end var
];
end signal
end module

```

### Problem 2.4: State Machines

Convert the following HCFSMD model of an elevator controller consisting of two concurrent, communicating state machines into an equivalent single, flat Mealy FSM *without* data (i.e. an FSM, *not* an FSMD). The HCFSMD has two external inputs (requested and current floor), three external outputs (up/down motor control and door open signals), and two internal signals (start timer and timer timeout). Unless specified otherwise, default value for all signals is absent/0.

