UCR CS122B, Winter 2007

Advance Embedded and Real Time Systems

Prof. Harry Hsieh

Homework #1

Due Date: Thursday, Jan 25th, at the beginning of the class
No late homework will be accepted

You may work in group, but you must turn in your own write-up.

1. (5 points) Define Embedded Systems.

2. (10 points) Given a particular product selling at $10 per unit, assume: a purely SW implementation with NRE cost at $20,000 and unit cost at $8; an FPGA implementation with NRE cost $40,000 and unit cost of $4, with some delay resulting in reduction of 10,000 unit in sale volume; an ASIC implementation with NRE cost of $100,000 and unit cost of $1, with significant delay resulting in reduction of 100,000 unit. How many units do we expect to sell, originally, to justify a SW implementation, an FPGA implementation, or a ASIC implementation?

3. (5 points) Draw block diagrams of processors
    a. First draw a General Purpose Processor.
    b. Modify the diagram to show application specific instruction processor. Why the modification and what is the trade-off?
    c. Modify the diagram to show single purpose processor. Why the modification and what is the trade-off?

4. (5 points) Define non-blocking communication and rendez-vous communication. Highlight the difference.

5. (10 points)Use CFSM, draw description of a seatbelt alarm controller with the following function: "If Key is turned on and the belt is not fasten, ring the alarm for 5 second, or until the belt is fasten." You may want to think about two concurrent processes: a controller and a timer. Write down any other assumption that you might have made, especially about the communication.

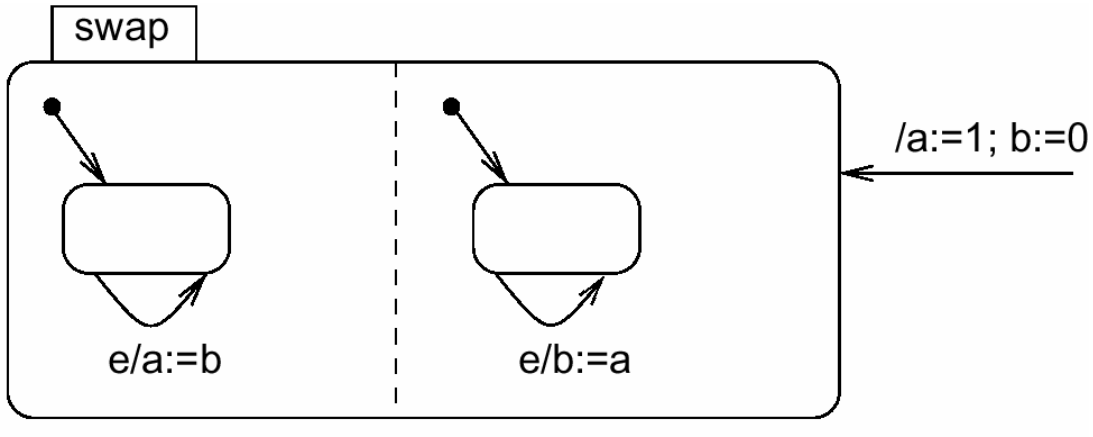
6. (10 points) Given the following netlist of the format, (component, input(s) / output(s))

INV IN1 / s1
NAND1 s1 IN2 / s2
NAND2 IN2 IN3 / s3
OR s1 s2 s3 / OUT

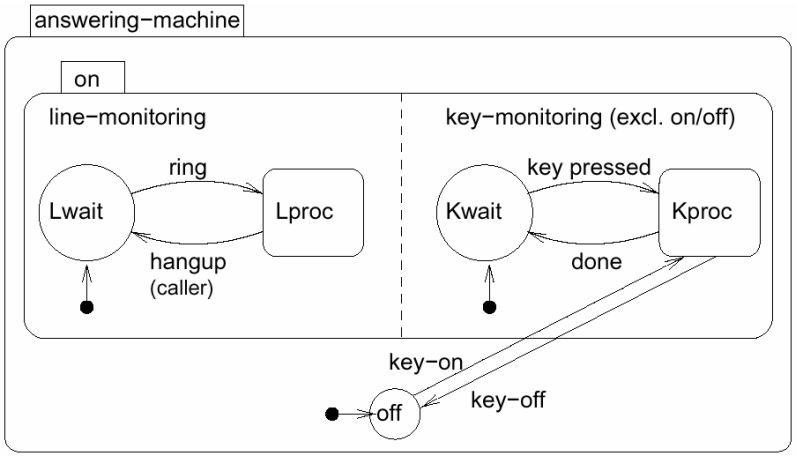
Assuming at time 0 IN1 goes from 1->0, and IN2 stays at 1, and IN3 goes from 0->1. Further assume inverter has 10ns delay while NAND and OR has 20ns delay. Draw the timing diagram using the concept of DE and delta cycle

7. (5 points) Extend Peterson's algorithm to 3 processes. Make sure it remains fair, lock-free, and mutually inclusive.

8. (5 points) For the following statechart, draw the value of a and b after
   a. 1 occurrence of e
   b. 2 occurrences of e
   c. 3 occurrences of e



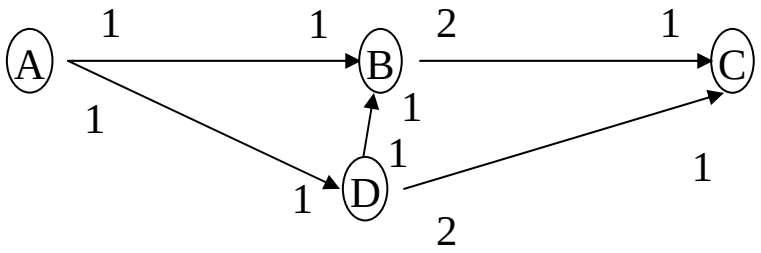
9. (5 points) Redraw the following state chart with regular FSM. Do NOT use AND-state, OR-state, or the exception mechanism. Do not use concurrency either (not CFSM). Treat Lproc and Kproc as simple states.



10.(10 points) Use Statemate semantic of Statechart as presented in class, draw state chart description of a seatbelt alarm controller with the following function: “If Key is turned on and the belt is not fasten, ring the alarm for 5 second, or until the belt is fasten.” You are required to use AND-state and (X)OR-state intelligently in your design. You may want to think about two concurrent processes: a controller and a timer. Write down any other assumption that you might have made.

11.(5 points) Given the following Synchronous Dataflow Graph

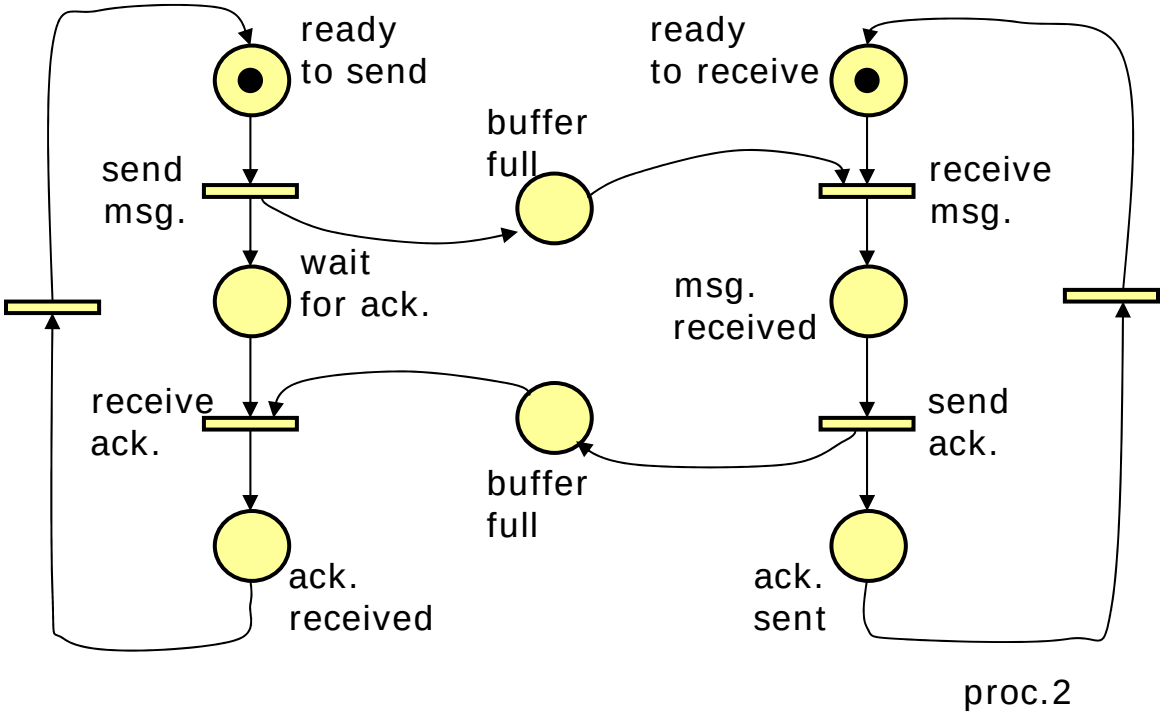
a. Is this SDF consistent? Prove it.
b. Give a valid single appearance schedule.
c. What is the buffer usage for your schedule?
d. Write the “code segment” associated with the invocation of this SDF graph.



12.(8 points) Redraw the dining philosopher example so that it can model the possibility of deadlock.

13.(10 points) Use Petri Net, draw description of a seatbelt alarm controller with the following function: “If Key is turned on and the belt is not fasten, ring the alarm for 5 second, or until the belt is fasten.” Write down any other assumption that you might have made. Is it live, safe, or fair? Explain.

14.(7 points) For the following petri net, draw the coverability tree/graph. explain why is it live, safe, or fair.


ready
to send
send
msg.
wait
for ack.
receive
ack.
ack.
received
buffer
full
buffer
full
ready
to receive
receive
msg.
msg.
received
send
ack.
ack.
sent
proc.2