ECE 153a/253
Homework 1

Due: Wed Oct. 13, 2010

Reading: Read the MicroBlaze ref and Embedded System ref, and the paper by Lavagno and Sentovitch--

Problems:

1. Define the following terms:

Refinement, Mapping, Binding, Composition, Validation, Verification, Synthesis

2. Uncertainty and Model Abstraction: A common mis-conception about computer behavior modeling is that the finite nature of the model means that system designs are complete. For example, a multi-issue reordering pipeline processor has a completely deterministic state (up to the effects of noise on synchronization elements!) so, in theory, the state for any possible set of instructions is predictable. In practice, such pipelines have multiple monitors which operate somewhat independently and which are conservative (i.e. they will tend to error by being restrictive of the state). Monitors make sure that the pipeline resources are not overwhelmed and that there are sufficient buffers for possible near-future activities (like a fixed-point divide or mod operation which takes multiple cycles). The upshot is that real designs have both constructive refined specifications, and implicit constraints like “no more than 3 pending operands on this FIFO if the next instruction is a branch”. Thus the state of affairs is that despite having all the specification, it is probably infeasible verify most future state claims on the design.

Given this state of affairs, why should we bother to strive to make modular, compassible models for design abstraction? What are the benefits beyond provable verification?

3. The Hand Mixer Control Loop is an practical example of a polled interface. The decision about such interfaces is usually driven by design constraints and, in particular, timing constraints. For simplicity, lets assume a new polled design with the following tasks:

task A: 5mS latency, must run every 45mS, requires data from task B and C to run.

task B: 2mS latency, must run every 75mS only requires state from old task B (previous iteration)

task C: 10mS latency, must run every 90mS, requires data from previous task B and task D

task D: 3mS latency, must run every 30mS

All data is timely, i.e. is only valid for the time till the next iteration (i.e. B’s output is good for 75-2mS = 73mS after it runs), it doesn’t matter if the process is run more often as long as you have state that is timely.

What are the fastest and slowest possible loop rates consistent with these constraints? Show the scheduling of tasks.