CprE 588 – Embedded Computer Systems

Homework #2

Assigned: February 24

Due: March 5

Directions:

- Please submit this assignment by the due date via WebCT.
- Submissions should be in the form of 1) a PDF file with the writeup and 2) a ZIP file containing any additional necessary material.
- You must submit individual work, although you may collaborate with classmates on the problems. Please acknowledge any such collaboration when submitting.
- Engineering Distance Education students are provided with an automatic one week extension to the due date.
- Some of these questions do not have a strictly correct answer. You will be graded based on how well-formed your arguments are.

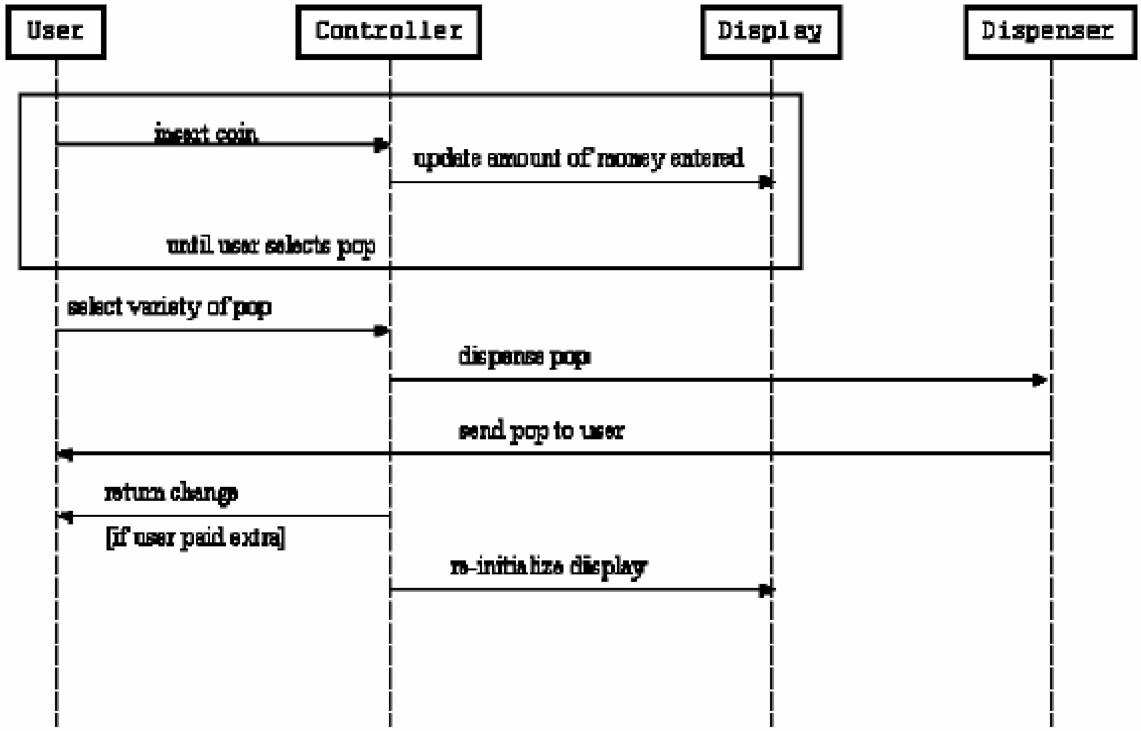
1) Specification Model

Suppose you are given the following problem specification as a UML Sequence Diagram. A brief background on UML sequence diagrams:

- A sequence diagram shows the interactions between several objects
- Each object represented by a box
- Vertical line represents time: the "lifeline" of object
- Time moves from top to bottom in a sequence diagram
- Communication between objects is modeled as messages
- Messages may have a condition in brackets

The following is a UML sequence diagram for purchasing a pop from a vending machine, given the following assumptions:

- The objects in the system are the user, the dispenser (releases cans of pop), the display (shows the amount of money entered and other messages), and the controller (receives coins, issues change, sends orders to the dispenser and updates the display).
- The user will enter enough money before selecting the variety of pop.
- The machine is not out of the variety of pop selected.
- The machine has enough change.



(a) Describe the functional behavior of the vending machine system in a short paragraph.

(b) Draw a top-level Program State Machine diagram for a specification model of the vending machine system.

(c) Write SpecC code for the specification model of the vending machine, as drawn in the last part. Use the SpecC compiler (scrc) to compile and simulate the model, as in Homework #1. Submit your code and simulation results.

(d) Explain if and how your specification model captures any of the following characteristics of embedded systems:

- Concurrency
- Hierarchy
- State transitions
- Communication
- Synchronization

2) Using SCE for SpecC Modeling

- Review an introduction to the SCE tool (link)
- X11 setup information, which is required to use SCE (link)
- Miscellaneous files used in the introduction (link)
- SCE tutorial (link)
- SCE specification model reference manual (link)

Figure 1 shows the modeling style you will use for all of your SpecC programs in SCE. For example, the vending machine controller corresponds to the Design behavior. Typically this is called a testbench.

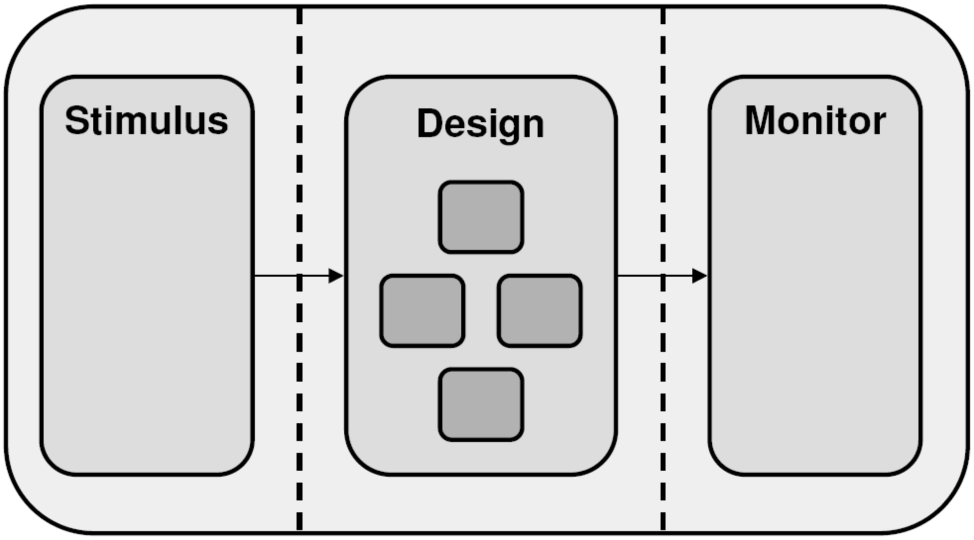


Figure 1: Testbench (tb.sc)

(a) Import the vending machine specification model you wrote for Question 1 into SCE. Use SCE to create the chart of your model. Include a copy with your homework submission.

(b) Refer to the SCE tutorial for information on how to profile your design. Which behavior is the most computation-intensive? Explain. Which behavior is the most connection-intensive? Explain.

3. Architecture Exploration and Refinement

See attached file *hw2.zip* which contains a modified version of the parity checker example. After downloading and extracting, open the file named *tb.sc* inside SCE. This is the testbench; within the code, the top-level Design behavior for the parity checker is named “sut” (system under test). You will need this information when performing the architecture refinement with SCE.

(a) Read the SCE tutorial (link) for instruction on how to do Processing Element (PE) allocation inside SCE. Allocate the same two PEs as in the tutorial: the Motorola_DSP5600 and the HW_Standard.

(i) Create a partition and list its quality metrics.
(ii) Create another, different partition and list its quality metrics.

Submit the charts for both partitions and their quality metrics.

(b) Add the DRAM_64 component to the architecture, and allocate the ‘data’ and ‘ocount’ variables to this memory element. Use same allocation scheme – you must partition the behaviors between the same two PEs as in part (a), while keeping the two variables in the shared memory. Submit the charts for both partitions and their quality metrics.

(c) SCE will insert channels as needed; however in some designs you will want to introduce a communication scheme from the beginning as part of the specification.

(i) Edit the original code for *parity.sc* by adding a channel for communication between components ‘U00’ and ‘U01’. Submit the code and chart for this new specification model with the channel included.
(ii) As in part (a), allocate two PEs and proceed with the architecture refinement. Submit the charts for both partitions and their quality metrics.

(d) Briefly compare and contrast the results of parts (a) – (c).