synchSMs in RI Tools

(1 Day)

This lab reviews use of concurrent synchSMs using the RIBS/RIMS tools.

PRELAB

No required prelab. However, students may wish to install the latest RI tools on their own machines, and review Programming Embedded Systems (PES).

Getting started:

- All course materials can be found on iLearn.ucr.edu
- View the course syllabus and subscribe to the course online textbook (it's free with the provided coupon)
- Log in to your CS account, open a terminal and start the virtual machine with the command 'win7' (without the ' ')
- Start RI tools, the software key can be found by logging in to the book web site and viewing the welcome page.

PART I -- Single synchSM

Bouncing LED: Using RIBS, capture a synchSM that lights the LEDs one-at-a-time in sequence: B0, then B1, then B2, ..., then B7, then B0 again. Each LED stays on for 1 second. Next, have A0 select the sequence direction: 0 as above, 1 in reverse.

Pattern Sequencer: Next, create any sequence of 10 output patterns, such as 11111111, 01010101, 10101010, ... (ten such patterns). Store them in an array. Each pattern displays for 1 second. A0 being 0 goes through the pattern forwards, 1 backwards.

Music Accompaniment: Use a smart phone/tablet to play at least 20 seconds of a (clean) song of your choice, create a light show that accompanies the song, following the song in some way.

PART II -- Concurrent synchSM's

Using at least three "concurrent" synchSMs, devise a system where A7A6 being 00 executes the above Bouncing LED synchSM, 01 the Pattern Sequencer, and 10 the Music Accompaniment. 11 simply outputs 11111111. While the synchSMs are concurrent, note that all but one will be waiting in a wait state at any time.

Demo to your TA.

PART III -- Creativity

Develop an idea for system with at least two truly concurrent synchSMs. The more interesting the behavior, the better. Note that the concurrency need not only relate to output displays, but may also detect certain events/sequences on the inputs. Capture the system in RIBS and simulate.

PART IV -- Partner Challenge

Write a specification of your above system for a partner to implement. Write your names along with each spec, e.g., Creator: Jonny Smith, Implementer: Marry Jones.

Demo the implementation of that other system to the system creator and get their approval that the system is correct.

When both are done, demo to your TA.

Submit your C code for Part IV on iLearn:

yourname_creatorname_lab1.c

PART V (Challenge)

Create a google doc that has a specification and diagrams for future people learning to program state machines . Make it interesting and fun. Perhaps a game or a real application. Show and demo to your TA.