Mapping Tutorial

Abhijit Davare

6/29/05

Overview

- Events are associated together with ltl synch constraints
  - Either one-way (implication) or two-way synch (simultaneity) statements are used
- Three main challenges
  - Figuring out which events need to be synchronized together
  - Obtaining references to those events
  - Emitting the synch constraints with optional variable equality

Which events?

- Usually the beginning or end of functions
  - E.g. beg(p, o.func) where “p” is a process, “o” is same as “p” or a medium, and “func” is the name of a function in “o”.
- May also use beginning and end of labels

Getting Event References

- An event reference has four parts:
  - beg or end
  - Process reference
  - Object reference
  - Label or function name
    - Don’t use functions or labels that are inherited from a superclass

Synch Constraints and Variables

- One-way or two-way implications
- For variable equality portion, one of the variables should be of type Nondet
  - E1 <=> E2: var1@E1 == var2@E2
  - Either var1 or var2 should be Nondet
- Currently, Nondet variables have integer values

simpleMapping example

- Simple mapping example located in metro/examples/simpleMapping from the Metropolis release
- Read the README.txt file for an overview
  - Mapping of architectural tasks to a Producer-Consumer-like functional model
  - Read and write services mapped along with relevant arguments

Detailed overview of mapper.mmm code

- Instantiate functional and architectural models and obtain references to functional and architectural processes
  - Lines 39-47
- One-to-one mapping between processes and tasks
  - “For” loop from line 50 to the end of the file

Detailed Overview cont'd

- Obtain references to all storage elements in output FIFOs by traversing netlist, ports and connections
  - Lines 62 – 90
- Similarly for all input FIFOs
  - Lines 92-119
- Obtain references to beginning and end of architectural read and write events
  - Lines 130-133

Detailed Overview cont'd

- Emit one-way synch constraints of the form: functional event ➔ architectural event
  - Lines 142 – 176
- Obtain the functional events
  - Lines 146 – 156
- Constrain the events, beginning of function includes variable equality
  - Lines 158-175

Detailed Overview cont'd

- Pad the arrays until all of equal size
  - Lines 179-185
- Emit synchronization constraints of the form: Architectural event ➔ Func. Event 1 || … || Functional Event n
  - Lines 186-283

More Information

- Syntax details in Metropolis Metamodel documents
- Intel mapping example in metroworkspace/examples/intel
- Email me at davare@eecs.berkeley.edu