High-level Synthesis from the Synchronous Language Esterel

Stephen A. Edwards

Department of Computer Science
Columbia University
www.cs.columbia.edu/~sedwards
Three Ideas for Esterel

Controller synthesis from Program Dependence Graph

- Control flow represented concurrently  [Ferrante et al. 1987]
- Construction usually $O(n)$  [Cytron et al. 1991]
- Trivial, efficient translation into circuits

High-level State Assignment

- Optimizers need reachable states  [Sentovich et al. 1997]
- High-level structure partitions, simplifies state assignment

Don’t-Care Extraction

- Control-flow relationships easy to analyze
- Controllability don’t-cares
An Example

sustain R ← Make R present forever
An Example

weak abort ← Make R present until A is sustain R when immediate A;
An Example

wait I;
weak abort
sustain R
when immediate A;
emit O

Wait for next cycle where I is present
Make signal O present this cycle
An Example

Infinite Loop

await l;
weak abort
sustain R
when immediate A;
emit O

end
An Example

loop
  await I;
  weak abort
    sustain R
  when immediate A;
  emit O
end
end

|| Run Concurrently

loop
  pause; pause;
  present R then
    emit A
end
end
An Example

every S do
  loop
    await I;
    weak abort
      sustain R
    when immediate A;
    emit O
  end
|| loop
  loop
    pause; pause;
    present R then
      emit A
    end
  end
end
An Example

every S do
  loop
    await l;
    weak abort
    sustain R
    when immediate A;
    pause
  end
end

||
loop
  pause, pause;
  present R then
  emit A
end
end

Same-cycle bidirectional communication
An Example

every S do
  loop
    await I;
    weak abort
    sustain R
    when immediate A;
    pause
  end
||
  loop
    pause; pause;
    present R then
    emit A
  end
end

Esterel: [Berry 1992]

Good for hierarchical FSMs
Cycle-based semantics like SystemC
High-level control constructs (exceptions, preemption)
Weak at data manipulation (e.g., no types, pointers)
Hardware Esterel variant proposed to address this
Translation to CCFG

every S do
  loop
    await I;
    weak abort
    sustain R
    when immediate A;
    emit O
  end
end

loop
  pause; pause;
  present R then
    emit A
  end
end
Translation to PDG
Translation to Circuitry
Want more?

See the paper

http://www.cs.columbia.edu/~sedwards