Compiling Esterel into Better Circuits

Jia Zeng
Department of Computer Science
Columbia University, New York

Esterel

- Characters effecting circuitry translation
  - High level control constructs
    - More complex, more challenges
    - Better understanding of program behavior
      - Enable aggressive optimization
  - Explicit and implicit pause statement
    - Implicit state machine

3 Stages of Circuit Translation

- 1. State assignment
  - Base on Program Dependence Graph (PDG) [9,2]
- 2. Hardware synthesis
  - Straight forward when coding has been chosen
- 3. Circuit optimization
  - 1) sequential optimization
    - State encoding
  - 2) combinational optimization
    - SIS

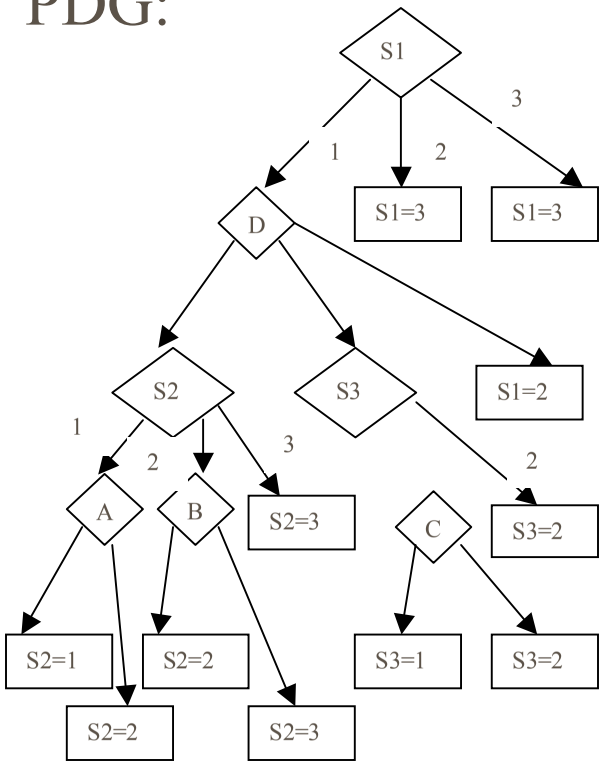
An Example of PDG

- An example program:

- abort
- [
- await A; await B
  ||
  await C;
  ]
  when D;

pause;

PDG:



State Encoding

- Heuristic Search
  - Find a solution with fewest latches under the requirement
  - Or, exhaust the search space
- Search Space
  - 1. Berry's one-hot encoding [1]
  - 2. Edwards's group-hot-by-level encoding [2]
  - 3. Variant

Classic State Assignment

- Based on Finite State Machine
- Reduce reachable state by minimizing the incomplete specified machines

NOVA: Two-level logic optimization

- Basic concept: State code adjacency
- Constraint satisfaction to graph-embedding problem
- Best strategy: iohybrid_code
  - ihybrid_code
    - 1 Partition input constrains to SIC, RIC
    - 2 Encode with minimum length under SIC
    - 3 Increase embedding cube to satisfy RIC within encoding space
  - ihybrid_code + output constrains

MUSTANG: Multilevel logic optimization

- Basic concept: State code adjacency
- Aim: maximize the size and number of common cubes
- Tool: Attraction Graph with weighted edges

Berry's Method of Translation

- First outlined in 1992 [1]
- Hardware synthesis
  - Sub-circuit for each statement
  - Registers only for pause statement
- Encoding leaf states by one-hot coding
  - Encoding/decoding circuits are trivial
  - Reachable state space redundant

Sentovich, Toma and Berry: the technique latches reducing

- Steps:
  - 1. Compute reachable state set using BDDs
  - 2. Remove sequential redundancies and re-synthesizing the circuit

Edwards's Advanced Method

- 1. CFG: New structural translation
  - More efficient at removing redundant circuits than removing by analyzing the circuit
- 2. Better state encoding
  - High level encoding – greater flexibility and larger encoding space
- 3. Don't care information
  - Helpful to logic synthesis

Bibliographies

- [1] Gerard Berry. Esterel on hardware. Philosophical Transactions of the Royal Society of London. Series A, 339:87-103, April 1992. Issue 1652, Mechanized Reasoning and Hardware Design.
- [2] Stephen A. Edwards. High-level synthesis from the synchronous language Esterel. In Proceedings of the International Workshop of Logic and Synthesis (IWLS). New Orleans, Louisiana, June 2002.

Bibliographies

- [3] Gerard Berry. Efficient latch optimization using exclusive sets. In Proceedings of the 34th Design Automation Conference, pages 8-11, Anaheim, California, June 1997.
- [4] Stephen A. Edwards. An Esterel compiler for large control-dominated systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 21(2), February 2002

Bibliographies (cont)

- [5] Ellen M. Sentovich, Horia Toma, Gerard Berry. Latch optimization in circuits generated from high-level descriptions. ICCAD'96, November 1996.
- [6] Gary D. Hachtel, Fabio Somenzi. Logic Synthesis and Verification Algorithms. Keluwer Academic Publishers. 1996.

Bibliographies (cont)

- [7] Tiziano Villa, Alberto Sangiovanni-Vincentelli. NOVA: State Assignment of Finite State Machines for Optimal Two-Level Logic Implementations. In The Proceedings of the 26th ACM/IEEE Design Automation Conference, pages 327-332, June 1989.

Bibliographies (cont)

- [8] S. Devadas, B. Ma, R. Newton, and A. Sangiovanni-Vincentelli. MUSTANG: State Assignment of Finite State Machines Targeting Multi-level Logic Implementations. IEEE Transactions on -Computer-Aided Design, vol. 7, no. 12, December 1988.
- [9] W. Baxter and H. R. Bauer, III. The program dependence graph and vectorization. In Proceedings of the Sixteenth Annual ACM SIGACT/SIGPLAN Symposium on Principles of Programming Languages, Austin, TX, 1989