Introduction to Logic Synthesis

Adnan Aziz
E.C.E.
University of Texas at Austin

Today's lecture

- Design automation
  - history
  - significance
- Course outline
  - applications
- Administrivia
- Boolean functions

VLSI design automation

- Need for VLSI design assistants fueled by
  - complexity of VLSI - \(10^7\) transistors on a chip
  - profitability strongly linked to time to market
- CAD tools are an integral part of hardware
design efforts
VLSI Design - Big Picture

RTL level → Gate level → Transistors → Layout

reg x[0:4];
x = (y+z) mod 6;

Design verification, High level synthesis
Logic optimization, test generation
Analog simulation
Place and route

CalTech experiment

- CalTech experiment - “silicon compilation” 197x
  - framework - pioneering effort
  - poor quality silicon
- Berkeley approach
  - bottom up, rooted in optimization
  - competitive with designers

Models and Algorithms

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>MODEL</th>
<th>ALGORITHMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>reg x[0:4]; x = (y+z) mod 6;</td>
<td>FSM network, Data Flow Graph</td>
<td>Graph traversal, Integer programming</td>
</tr>
<tr>
<td></td>
<td>Graph of logical operators and delay elements</td>
<td>Boolean satisfiability, graph covering</td>
</tr>
<tr>
<td></td>
<td>Differential equations</td>
<td>Numerical analysis</td>
</tr>
<tr>
<td></td>
<td>Geometric objects (polygons)</td>
<td>min spanning tree, convex hull</td>
</tr>
</tbody>
</table>

CalTech experiment - “silicon compilation” 197x
+ framework - pioneering effort
- poor quality silicon
Berkeley approach
+ bottom up, rooted in optimization
+ competitive with designers

Berkeley approach
+ bottom up, rooted in optimization
+ competitive with designers
Commercial significance

- Semiconductor market 1996
  - 120 Billion US $ - 85% digital
- Microelectronics CAD 1996
  - 1.2 Billion US $
    - System level = 75 million US $
    - RT level = 375 million US $
    - Gate level = 275 million US $
    - Misc = 275 million US $

Logic synthesis - technology

- Many ways of implementing design in VLSI
  - Custom
  - Semicustom: cell based and array based
    - Cell based - standard cell/macro cell
    - Array based - MPGA/FPGA
- Choice based on various factors
  - Density, performance, flexibility, design time, manufacturing time, cost (volume)
- Logic synthesis - largely technology independent

What will you get from this class?

Study automatic generation of gate level implementations from HDL specifications

- Fundamental CAD algorithms + data structures:
  - Graph covering, SAT heuristics, shortest path,...
  - Boolean logic, BDDs, TBFs, FSMs,...
- Tools: end user + programming
  - VIS, SIS, Synopsys
  - Verilog, BLIF, BLIF-mv
Course summary

- 2 level logic minimization
  - Boolean functions, PLAs
- Multi-level logic minimization
  - share common logic, node simplification
- Topics in logic synthesis
  - timing analysis & optimization, testing, FPGAs
- Sequential synthesis
  - sequential functions, encoding, state minimization, retiming, verification

2 level minimization

- Reduce size of PLA implementing logic function

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Designer specification

VLSI implementation

Multi-level minimization

- Logic network

1. factoring
   \[ f = ab + c' + d'e'; \]
   \[ g = ab + c' + de; \]
2. node simplification
   \[ h = ab + c'; \]
   \[ f = h + d'e'; \]
   \[ g = h + de; \]

Designer specification

VLSI implementation
Timing analysis & optimization

- how fast is design?

- make design faster

Sequential synthesis

- representation: FSM

- encoding:

- retiming:

Administrivia - I

- Prerequisites
  - Digital design, C, mathematical maturity

  - DeMicheli, McGraw Hill 1994

- Format/Evaluation:
  - 8 homeworks = 30%
  - Midterm 1 + Midterm 2 = 35 %
  - Project = 35%