ESE535: Electronic Design Automation

Day 17: March 23, 2011
Two-Level Logic-Synthesis

Problem

• **Given**: Expression in combinational logic
• **Find**: Minimum (cost) sum-of-products expression
• Ex.
  - \( Y = a^*b^c + a^*b^c + a^*b^c \)
  - \( Y = a^*b + a^*c \)

EDA Use

• Minimum size PLA, PAL, …
  – Programmable Logic Array
  – Programmable Array Logic
• Minimum number of gates for two-level implementation
• Starting point for multi-level optimization

PLA

• Directly implement flat (two-level) logic
  – \( O = a^*b^c*d + a^*b^d + b^c*d \)
• Exploit substrate properties allow wired-OR
**Wired-or**

- Connect series of inputs to wire
- Any of the inputs can drive the wire high

**Programmable Wired-or**

- Use some memory function to programmable connect (disconnect) wires to OR
- Fuse:

**Diagram Wired-or**

**Wired-or array**

- Build into array
  - Compute many different or functions from set of inputs
Combined or-arrays to PLA

- Combine two or (nor) arrays to produce PLA (or-and / and-or array)

PLA

- Can implement each and on single line in first array
- Can implement each or on single line in second array

Strictly speaking: or in first term and in second, but with both polarities of inputs, can invert so is and-or.

Nanowire PLA

PLA and PAL

PAL = Programmable Array Logic
PAL has fixed OR plane.

EDA Use for 2-level Logic Min.

- Minimum size PAL, PLA, ...
  - Programmable Logic Array
  - Programmable Array Logic
- Minimum number of gates for two-level implementation
- Starting point for multi-level optimization
Complexity

• Set covering problem
  – NP-hard

Terminology (1)

• Literals -- a, /a, b, /b, ....
  – Qualified, single inputs
• Minterms --
  – full set of literals covering one input case
  – in y=a*b+a*c
    • a*b*c
    • a*/b*c

Cost

• PLA/PAL – to first order costs is:
  – number of product terms
• Abstract (mis, sis)
  – {multilevel,sequential} interactive synthesis
  – number of literals
    • cost(y=a*b+a*c) = 4
• General (simple, multi-level)
  – \( \sum \) cost(product-term)
    • e.g. nand2=4, nand3=5,nand4=6...

Terminology (2)

• Cube:
  – product covering one or more minterms
  – Y=a*b+a*c
  – cubes:
    • a*b*c    abc
    • a*b       ab
    • a*c       ac

Terminology (3)

• Cover:
  – set of cubes
  – sum products
  – \{abc, a/bc, ab/c\}
  – \{ab, ac\}

Truth Table

• Also represent function
  Specify on-set only

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Cube/Logic Specification

- Canonical order for variables
- Use \{0,1,-\} to indicate input appearance in cube
  - 0 = inverted  
  - 1 = not inverted  
  - - = not present

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

In General

- Three sets:
  - on-set (must be set to one by cover)
  - off-set (must be set to zero by cover)
  - don't care set (can be zero or one)
- Don’t Cares:
  - allow freedom in covering (reduce cost)
  - arise from cases where value doesn’t matter
    - e.g. outputs in non-existent FSM state
    - data bus value when not driving bus

Multiple Outputs

- Can reduce to single output case
  - write equations on inputs and each output
    - with onset for relation being true
    - after cover
      - remove literals associated with outputs

Multiple Outputs

- Could Optimize separately
- By optimizing together
  - Maximize sharing of cubes/product-terms

Multiple Outputs

- Consider:
  - \[ X=\overline{a}/b+ab+ac \]
  - \[ Y=\overline{bc} \]
- Trivial solution has 4 product terms
Multiple Outputs

• Consider:
  – $X = /a/b + ab + ac$
  – $Y = /bc$

• Now read off cover:
  – $Y = /bc$
  – $A = /a/b/c + /bc + ab$

  Only need 3 product terms (versus 4 w/ no sharing)

Prime Implicants

• Implicant -- cube in on-set
  – (not entirely in don’t-care set)

• Prime Implicant -- implicant, not contained in any other cube
  – for $y = a^*b + a^*c$
    • $a^*b$ is a prime implicant
    • $a^*b^*c$ is not a prime implicant (contained in $ab$, $ac$)
  – i.e. largest cube still in on-set (on+dc-sets)

Prime Implicants

• Minimum cover will be made up of primes
  – fewer products if cover more
  – fewer literals in prime than contained cubes

• Necessary but not sufficient that minimum cover contain only primes
  – $y = ab + ac + b/c$
  – $y = ac + b/c$

• Number of PI’s can be exponential in input size
  – more than minterms, even!
  – Not all PI’s will be in optimum cover

Essential Prime Implicants

• Prime Implicant which contains a minterm not covered by any other PI
  – Essential PI must occur in any cover
  – $y = ab + ac + b/c$
  – $ab\ 11\ - 110\ 111$
  – $ac\ 1-1\ 101\ 111$ * essential (only 101)
  – $b/c\ -10\ 110\ 010$ * essential (only 010)

Restate Goal

• Goal in terms of PIs
  – Find minimum size set of PIs that cover the on-set.

Computing Primes

• Start with minterms
  – for on-set and dc-set
• merge pairs (distance one apart)
• for each pair merged,
  – mark source cubes as covered
• repeat merging for resulting cube set
  – until no more merging possible
• retain all unmarked cubes which aren’t entirely in dc-set
### Compute Prime Example

0  0000  0, 8 -000  
5  0101  5, 7 01-1  
7  0111  7,15 -111  
8  1000  8, 9 100-  
9  1001  9,10,11 10-0  
10 1010  10,11,14,15 1-1-  
11 1011  11,15 1-11  
14 1110  14,15 111-  
15 1111  14,15 111-  

(in-class assignments, back of preclass sheet; record solutions on board.)

Note this is preclass 3.

### Covering Matrix

<table>
<thead>
<tr>
<th>Minterms</th>
<th>Prime Implicants</th>
<th>Goal: minimum cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Essential Reduction

- **Must pick essential PI**
  - Pick and eliminate row and column

Which essential?

<table>
<thead>
<tr>
<th>/b/c/d</th>
<th>/abd</th>
<th>bcd</th>
<th>a/b</th>
<th>ac</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Essential Reduction

- **This case:**
  - Cover determined by essentials
  - Preclass 3: ac+a/b+/abd+/b/c/d

- **General case:**
  - Reduces size of problem
Dominators: Column

• If a column (PI) covers the same or strictly more than another column – can remove dominated column

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

C dominates B

Any others?

G dominates H

New Essentials

• Dominance reduction may yield new Essential PIs

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

C, G now essential

What’s essential?

Dominators: Row

• If a row has the same (or strictly more) PIs than another row, the larger row dominates – we can remove the dominating row

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

C, G now essential

What’s essential?

Any others?

E dominates D and F

Cover = {C, E, G}

Cyclic Core

• After applying reductions – essential – column dominators – row dominators

• May still have a non-trivial covering matrix

• How do we move forward from here?

Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cyclic Core

- Cannot select (e.g. essential) or exclude (e.g. dominated) a PI definitively.
- Make a guess
  - A in cover
  - A not in cover
- Proceed from there

Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A in Cover:

- C dominates B
- G dominates H

Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A not in Cover:

What now?

Basic Two-Level Minimization (espresso-exact)

- Generate Prime Implicants
- Reduce (essential, dominators)
- If not done,
  - pick a cube
  - branch (back to reduce) on selected/not
    - i.e. search tree … branch and bound
- Save smallest

Branching Search

A in cover

A not in cover

\{A,B\}, \{A,B\}, \{A\}

A and B not in cover

\{A,B,C\}, \{A,B,C\}
Branching Search w/ Implications

- A in cover
- A not in cover
- \{A/B, C\}
- \{/A,B,/C\}

Implications Prune Tree
(like BCP in SAT)

Only exponential in decision where must branch

Optimization

- Summarize Minterms (signature cubes)
  - rows represent collection of minterms with same primes
- Avoid generating full set of PIs
  - pre-combining dominators during generation
- Branch-and-bound pruning
  - get lower bound on remaining cost of a cover by computing independent set of primes
    - (not necessarily maximal, that would be NP-hard)

Heuristic

- Don’t backtrack when select prime for inclusion/exclusion
  - pick cover large set of minterms/signatures
  - weight to select “hard” to cover signatures
- Generate reduced set of PIs
- Iterative improvement

Canonical Form

- Can start with any form of logical expression
- Get unique truth-table/minterms
- Problem not sensitive to input statement
  - compare covering (decomposition)
  - compare sequential programming languages
- Cost: potentially exponential explosion in minterms/PIs

Summary

- Formulate as covering problem
- Solution space restricted to PIs
- Essentials must be in solution
- Use dominators to further reduce space
- Then branching/pruning to explore rest of PIs
- Ways to reduce work
  - group minterms/PIs together early
  - mostly fall into this general scheme

Admin

- Reading for Monday online (web)
- Assign5a graded
- Assign5b due Monday
Big Ideas

- Canonical Form
  - eliminate bias of input specification
- Technique:
  - branch-and-bound
  - pruning search – exploit structure
  - Dominators