MULTIPLE-LEVEL LOGIC OPTIMIZATION

© Giovanni De Micheli
Stanford University

Outline

- Representations.
- Taxonomy of optimization methods:
  - Goals: area/delay.
  - Algorithms: algebraic/Boolean.
  - Rule-based methods.
- Examples of transformations.
- Boolean and algebraic models.

Motivation

- Multiple-level networks:
  - Semi-custom libraries.
  - Gates versus macros (PLAs):
    * More flexibility.
    * Better performance.
- Applicable to a variety of designs.

Circuit modeling

- Logic network:
  - Interconnection of logic functions.
  - Hybrid structural/behavioral model.
- Bound (mapped) networks:
  - Interconnection of logic gates.
  - Structural model.
Example of bound network

\[ p = ce + de \]
\[ q = a + b \]
\[ r = p + a' \]
\[ s = r + b' \]
\[ t = ac + ad + bc + bd + e \]
\[ u = q'c + qc' + qc \]
\[ v = a'd + bd + c'd + ae' \]
\[ w = v \]
\[ x = s \]
\[ y = t \]
\[ z = u \]

Example of network

Example circuit terminal behavior

\[ f = \begin{bmatrix} d'd + bd + c'd + ae' \\ d' + b' + c + d \\ ac + ad + bc + bd + e \\ a + b + c \end{bmatrix} \]
Network optimization

- Minimize area (power) estimate:
  - subject to delay constraints.

- Minimize maximum delay:
  - subject to area (power) constraints.

- Maximize testability.

- Minimize power.

Estimation

- Area:
  - Number of literals.
  - Number of functions/gates.

- Delay:
  - Number of stages.
  - Refined gate delay models.
  - Sensitizable paths.

Problem analysis

- Multiple-level optimization is hard.

- Exact methods:
  - Exponential complexity.
  - Impractical.

- Approximate methods:
  - Heuristic algorithms.
  - Rule-based methods.

Strategies for optimization

- Improve circuit step by step.
  - Circuit transformations.

- Preserve network behavior.

- Methods differ in:
  - Types of transformations.
  - Selection and order of transformations.
Example elimination

- Eliminate one function from the network.
- Perform variable substitution.
- Example:
  - \( s = r + b' \); \( r = p + a' \)
  - \( s = p + a' + b' \).

Example decomposition

- Break one function into smaller ones.
- Introduce new vertices in the network.
- Example:
  - \( v = a'd + bd + c'd + ae' \).
  - \( j = d' + b + c'; v = jd + a e' \).
• Find a common sub-expression of two (or more) expressions.

• Extract sub-expression as new function.

• Introduce new vertex in the network.

• Example:
  \(- p = ce + de\); \(t = ac + ad + bc + bd + e\);
  \(- p = (c + d)e\); \(t = (c + d)(a + b) + e\);
  \(- \Rightarrow k = c + d\); \(p = ke\); \(t = ka + kb + e\);

• Simplify a local function.

• Example:
  \(- u = q'c + q'd + qc\);
  \(- \Rightarrow u = q + c\)
• Simplify a local function by using an additional input that was not previously in its support set.

• Example:
  
  \[- t = ka + kb + e. \]
  \[\therefore t = kg + e \]
  \[- \text{Because } q = a + b. \]

\[ j = a' + b + c' \]
\[ k = c + d \]
\[ q = a + b \]
\[ s = ke + a' + b' \]
\[ t = kg + e \]
\[ u = q + c \]
\[ v = jd + ae' \]
**Algorithmic approach**

- Each operator has well-defined properties:
  - Heuristic methods still used.
  - Weak optimality properties.

- Sequence of operators:
  - Defined by *scripts*.
  - Based on experience.

---

**Example elimination algorithm**

- Set a threshold $k$ (usually 0).
- Examine all expressions.
- Eliminate expressions if the increase in literals does not exceed the threshold.

---

```plaintext
ELIMINATE( Gn(V,E), k) {
    repeat {
        vx = selected vertex with value < k;
        if (vx = 0) return;
        replace x by fx in the network;
    }
}
```

---

**Example MIS/SIS rugged script**

- sweep; eliminate -1
- simplify -m nocomp
- eliminate -1
- sweep; eliminate 5
- simplify -m nocomp
- resub -a
- fx
- resub -a; sweep
- eliminate -1; sweep
- full-simplify -m nocomp
Boolean and algebraic methods

Boolean methods:
- Exploit properties of logic functions.
- Use don’t care conditions.
- Complex at times.

Algebraic methods:
- View functions as polynomials.
- Exploit properties of polynomial algebra.
- Simpler, faster but weaker.

Example

Boolean substitution:
- $h = a + bcd + e$; $q = a + cd$
- $\Rightarrow h = a + bq + e$
- Because $a + bq + e = a + b(a + cd) + e = a + bcd + e$.

Algebraic substitution:
- $t = ka + kb + e$.
- $\Rightarrow t = kq + e$
- Because $q = a + b$.

Summary

Multilevel logic synthesis is performed by step-wise transformations.

Algorithms are based on both the Boolean and the algebraic models.

Rule-based systems.