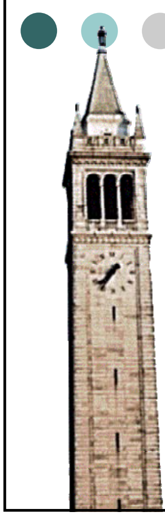


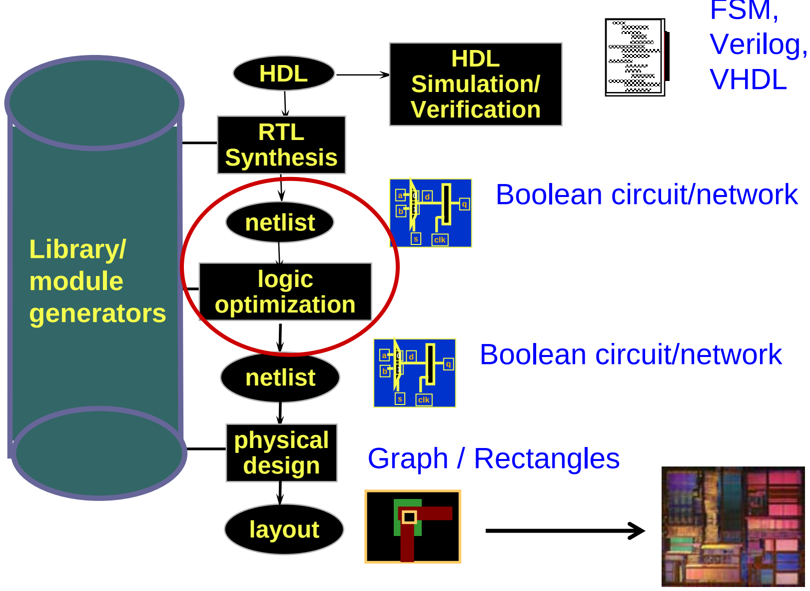
Fundamental Algorithms for System Modeling, Analysis, and Optimization

Edward A. Lee, Jaijeet Roychowdhury, Sanjit A. Seshia
UC Berkeley
EECS 144/244
Fall 2011

Lec 3: Boolean Algebra and Logic Optimization - 1

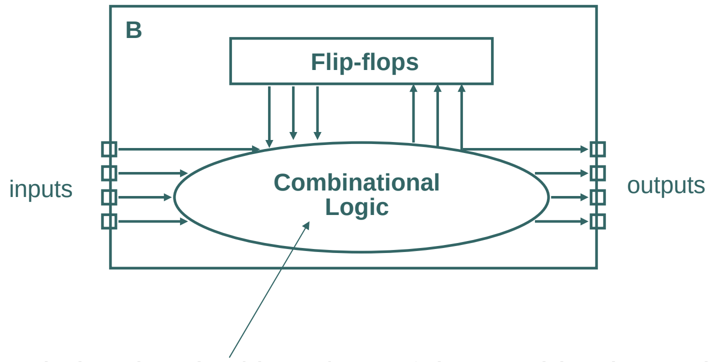
Thanks to S. Devadas, K. Keutzer, S. Malik, R. Rutenbar for several slides

RTL Synthesis Flow



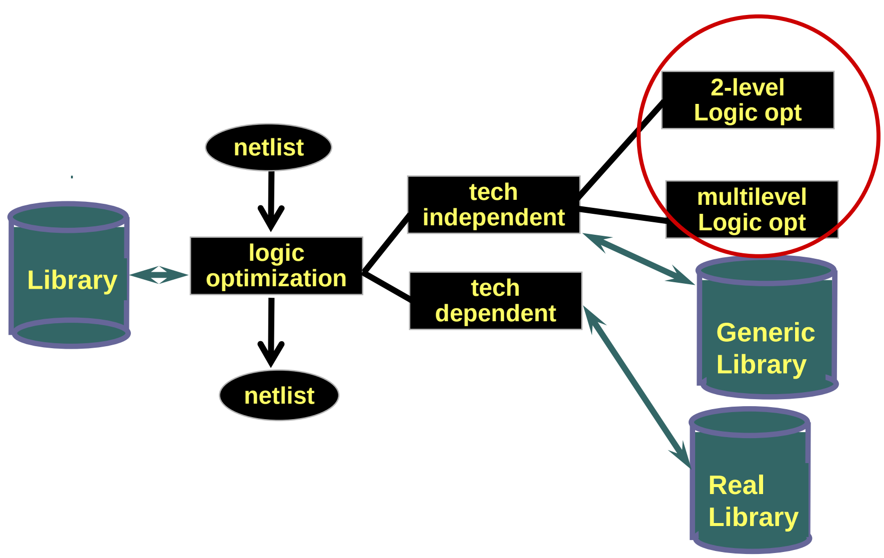
K. Keutzer

Reduce Sequential Ckt Optimization to Combinational Optimization



Optimize the size/delay/etc. of the combinational circuit (viewed as a Boolean network)

Logic Optimization



Outline of Topics

Basics of Boolean algebra

Two-level logic optimization

Multi-level logic optimization

Boolean function representation: BDDs

Definitions – 1: What is a Boolean function?

Definitions – 1: What is a Boolean function?

Let $B = \{0, 1\}$ and $Y = \{0, 1\}$

Input variables: $X_1, X_2 \dots X_n$

Output variables: $Y_1, Y_2 \dots Y_m$

A logic function ff (or 'Boolean' function, switching function) in n inputs and m outputs is a map

ff: $B^n \longrightarrow Y^m$

Definition used in Logic Optimization

don't care – aka "X"

Let $B = \{0, 1\}$ and $Y = \{0, 1, 2\}$

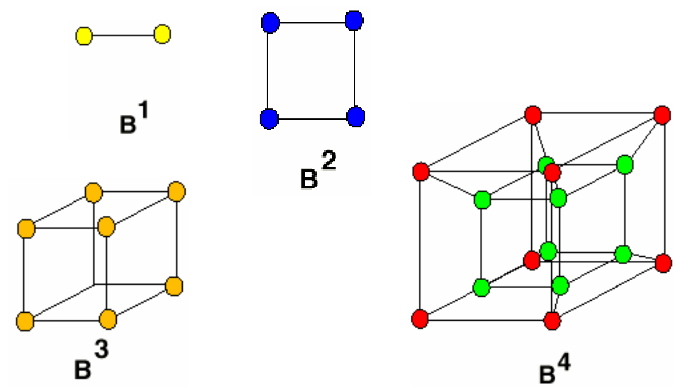
Input variables: $X_1, X_2 \dots X_n$

Output variables: $Y_1, Y_2 \dots Y_m$

A logic function ff (or 'Boolean' function, switching function) in n inputs and m outputs is a map

ff: $B^n \longrightarrow Y^m$

The Boolean n-Cube, B^n



- $\mathcal{B} = \{0, 1\}$
- $\mathcal{B}^2 = \{0, 1\} \times \{0, 1\} = \{00, 01, 10, 11\}$

Definitions – 2: ON/OFF/DC sets

If a logic function ff maps some input $b \in B^n$ to a 2 on some output i then function is incompletely specified, else completely specified

ON-SET$_i$ $\subseteq$ B^n, the set of all input values for which ff$_i$(x) = 1

OFF-SET$_i$ $\subseteq$ B^n, the set of all input values for which ff$_i$(x) = 0

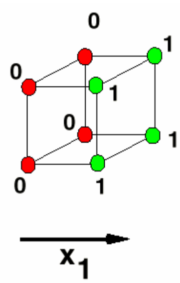
DC-SET$_i$ $\subseteq$ B^n, the set of all input values for which ff$_i$(x) = 2

Literals: What is a literal?

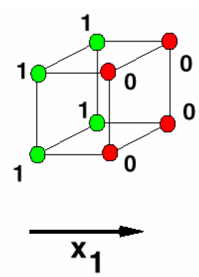
Literals

A literal is a variable or its negation y, $\overline{y}$

It represents a **logic function**



$f = x_1$



$f = \overline{x}_1$

Green – ON-set
Red – OFF-set

Boolean Formulas -- Syntax

Boolean functions can be **represented** by **formulas** defined as catenations of

- parentheses - (,)
- literals - $x, y, z, \overline{x}, \overline{y}, \overline{z}$
- Boolean operators - $+$ (OR), $\times$ (AND)
- complementation - e.g. $\overline{x + y}$

Examples:

$$\begin{aligned} f &= x_1 \times \overline{x}_2 + \overline{x}_1 \times x_2 \\ &= (x_1 + x_2) \times (\overline{x}_1 + \overline{x}_2) \\ h &= a + b \times c \\ &= \overline{\overline{a} \times (\overline{b} + \overline{c})} \end{aligned}$$

We will usually replace $\times$ by catenation, e.g. $a \times b \rightarrow ab$.

"Semantic" Description of Boolean Function

EXAMPLE: Truth table form of an incompletely specified function

ff: $B^3 \longrightarrow Y^2$

X_1	X_2	X_3	Y_1	Y_2
0	0	0	1	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	2
1	1	0	1	1
1	1	1	2	1

Y_1: ON-SET$_1$ = {000, 001, 100, 101, 110}
OFF-SET$_1$ = {010, 011}
DC-SET$_1$ = {111}

Operations on Logic Functions

(1) Complement: $f \longrightarrow \bar{f}$
interchange ON and OFF-SETS

(2) Product (or intersection or logical AND)
$h = f \cdot g$ (what happens to ON/OFF sets?)

(3) Sum (or union or logical OR):
$h = f + g$ (ON/OFF sets?)

CNF and DNF

CNF: Conjunctive Normal Form (product of sums: POS)

DNF: Disjunctive Normal Form (sum of products: SOP)

CNF → DNF: what is the worst-case blow up?

How about DNF → CNF?

Cube

A cube is a conjunction (AND) of literals

Examples: (set of variables = {a,b,c,d})

- ab
- $ab\bar{d}$
- $a\bar{b}c\bar{d}$

A cube is a logic function (also view as set)

2-level Minimization: Minimizing SOP (DNF)

$$F1 = \bar{A}\,\bar{B} + \bar{A}\,B\,D + \bar{A}\,B\,\bar{C}\,\bar{D} + A\,B\,C\,\bar{D} + A\,\bar{B} + A\,B\,D$$

Inputs	**Outputs**
0 0 - -	**1**
0 1 - 1	**1**
0 1 0 0	**1**
1 1 1 0	**1**
1 0 - -	**1**
1 1 - 1	**1**

$$F1 = \bar{B} + D + \bar{A}\,\bar{C} + A\,C$$

- 0 - -	**1**
- - - 1	**1**
0 - 0 -	**1**
1 - 1 -	**1**

minimum representation
(number of cubes, literals)

Implicants

An *implicant* of f is a *cube* p that does not intersect the OFF-SET of f

$p \subseteq f_{ON} \cup f_{DC}$

Prime Implicants

An *implicant* of f is a *cube* p that does not intersect the OFF-SET of f

$p \subseteq f_{ON} \cup f_{DC}$

A *prime implicant* of f is an implicant p such that

(1) No other implicant q contains it (i.e. $p \not\subseteq q$)

(2) $p \not\subseteq f_{DC}$

A *minterm* is a fully specified implicant

e.g., 011, 111 (not 01-)

Examples of Implicants/Primes

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

000, 00- are implicants, but not primes (-0-)

1-1

0-0

Prime and Irredundant Covers

A *cover* is a set of cubes C such that

$C \supseteq f_{ON}$ and $C \subseteq f_{ON} \cup f_{DC}$

All of the ON-set is covered by C

C is contained in the ON-set and Don't Care Set

A *prime cover* is a cover whose cubes are all prime implicants

An *irredundant cover* is a cover C such that removing any cube from C results in a set of cubes that no longer covers the function (ON-set)

Example Covers

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

0 0 -
1 0 - is a cover.
1 1 -

Example Covers

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

0 0 -
1 0 - is a cover. Is it prime?
1 1 - Is it irredundant?

Minimum covers

Defn: A *minimum cover* is a cover of minimum cardinality

Theorem: There exists a minimum cover that is a prime and irredundant cover.

Why?

Minimum covers

Defn: A *minimum cover* is a cover of minimum cardinality

Theorem: There exists a minimum cover that is a prime and irredundant cover.

Given any cover C

(a) if redundant, not minimum

(b) if any cube q is not prime, replace q with prime $p \supseteq q$ and continue until all cubes prime; it is a minimum prime cover

Example Covers

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

0 0 -
1 0 - **is a cover. Is it prime?**
1 1 - **Is it irredundant?**

What is a minimum prime and irredundant cover for the function?

Example Covers

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

0 0 -
1 0 - **is a cover. Is it prime?**
1 1 - **Is it irredundant?**

- 0 -
1 1 - **is a cover. Is it prime?**
Is it irredundant?
Is it minimum?

Example Covers

X_1	X_2	X_3	Y_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	2

0 0 -
1 0 - is a cover. Is it prime?
1 1 - Is it irredundant?

- 0 -
1 1 - is a cover. Is it prime?
Is it irredundant?
Is it minimum?

What about
- 0 -
1 - -

The Quine-McCluskey Method: Exact Minimization

Step 1: List all minterms in ON-SET and DC-SET

Step 2: Use a prescribed sequence of steps to find all the prime implicants of the function

Step 3: Construct the prime implicant table

Step 4: Find a minimum set of prime implicants that cover all the minterms

Espresso Algorithm: Heuristic Minimization

ESPRESSO (F, DC) {

F is ON-SET, DC is Don't Care Set

1. R = U - (F ∪ DC) U is universe cube
2. n = |F|
3. F = *Reduce* (F, DC); // reduce implicants in F to non-prime cubes
4. F = *Expand* (F, R); // expand cubes to prime implicants
5. F = *Irredundant* (F, DC); // extract minimal cover of prime implicants
6. If |F| < n goto 2, else, post-process & exit

}

Multi-level Logic Optimization

2-level optimization is a 'solved' problem:

Espresso is considered the last word on the topic

But most circuits are not two-level!

Need techniques to optimize size of multi-level circuits

- Size measured in terms of number of literals, depth of the circuit, etc.