Lec 3: Boolean Algebra and Logic Optimization - 1

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

RTL Synthesis Flow

Library/module generators

RTL Synthesis

HDL Simulation/Verification

HDL

Boolean circuit/network

FSM, Verilog, VHDL

Boolean circuit/network

Graph / Rectangles

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

Library → logic optimization → netlist

- tech independent
- tech dependent

Generic Library → multilevel Logic opt

Real Library → 2-level Logic opt
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: \( X_1, X_2 \ldots X_n \)
Output variables: \( Y_1, Y_2 \ldots Y_m \)
A logic function \( f \) (or ‘Boolean’ function, switching function) in \( n \) inputs and \( m \) outputs is a map
\[ f: B^n \rightarrow Y^m \]

Definition used in Logic Optimization

Let \( B = \{0, 1\} \) and \( Y = \{0, 1, 2\} \)
Input variables: \( X_1, X_2 \ldots X_n \)
Output variables: \( Y_1, Y_2 \ldots Y_m \)
A logic function \( f \) (or ‘Boolean’ function, switching function) in \( n \) inputs and \( m \) outputs is a map
\[ f: B^n \rightarrow Y^m \]

\( \text{don't care – aka "X"} \)
The Boolean n-Cube, $B^n$

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

Definitions – 2: ON/OFF/DC sets

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

- $\text{ON-SET}_i \subseteq B^n$, the set of all input values for which $f_f(x) = 1$
- $\text{OFF-SET}_i \subseteq B^n$, the set of all input values for which $f_f(x) = 0$
- $\text{DC-SET}_i \subseteq B^n$, the set of all input values for which $f_f(x) = 2$
Literals: What is a literal?

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

It represents a logic function

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:
\[
\begin{align*}
f &= x_1 \times \overline{x_2} + \overline{x_1} \times x_2 \\
   &= (x_1 + x_2) \times (\overline{x_1} + \overline{x_2}) \\
h &= a + b \times c \\
   &= \overline{a} \times (\overline{b} + \overline{c})
\end{align*}
\]

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

\[
\begin{array}{cccccc}
X_1 & X_2 & X_3 & Y_1 & Y_2 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 2 \\
1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 2 & 1 \\
\end{array}
\]

\( Y_1: \ \text{ON-SET}_1 = \{000, 001, 100, 101, 110\} \)
\( \text{OFF-SET}_1 = \{010, 011\} \)
\( \text{DC-SET}_1 = \{111\} \)
Operations on Logic Functions

(1) Complement: \( f \rightarrow \overline{f} \)
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 \( \rightarrow \) DNF: what is the worst-case blow up?

How about DNF \( \rightarrow \) CNF?
Cube

A cube is a conjunction (AND) of literals

Examples: (set of variables = \{a,b,c,d\})
- \(ab\)
- \(abd\)
- \(abcd\)

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

2-level Minimization: Minimizing SOP (DNF)

\[ F_1 = \overline{A} \overline{B} + \overline{A} B + D + \overline{A} B C \overline{D} + A B C D + A B + A B D \]

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 - -</td>
<td>1</td>
</tr>
<tr>
<td>0 1 - 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0 - -</td>
<td>1</td>
</tr>
<tr>
<td>1 1 - 1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ F_1 = \overline{B} + D + \overline{A} \overline{C} + A C \]

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 0 - -</td>
<td>1</td>
</tr>
<tr>
<td>- - - 1</td>
<td>1</td>
</tr>
<tr>
<td>0 - 0 -</td>
<td>1</td>
</tr>
<tr>
<td>1 - 1 -</td>
<td>1</td>
</tr>
</tbody>
</table>

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. $p \nsubseteq q$)
2. $p \nsubseteq f_{DC}$

A minterm is a fully specified implicant

e.g., 011, 111 (not 01-)
### Examples of Implicants/Primes

<table>
<thead>
<tr>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>Y₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<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>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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
\[
C \supseteq f_{ON} \quad \text{and} \quad C \subseteq f_{ON} \cup f_{DC}
\]

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

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<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>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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

Example Covers

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<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>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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

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

### Example Covers

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y_1$</th>
<th>$Y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</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>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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

Is it minimum?
Example Covers

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<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>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

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

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.