Gate Level Minimization Lecture Notes

Group 13

February 17, 2025

1 Introduction

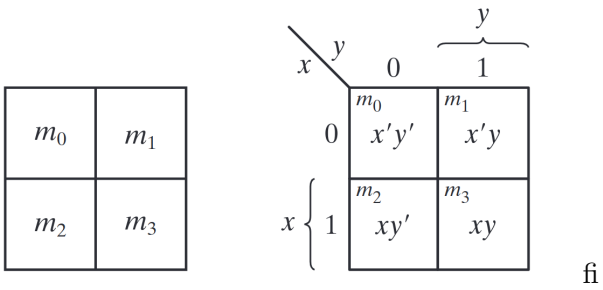
Gate-level minimization is the process of finding optimal implementations of Boolean functions using logic gates. Key aspects include:

- Reducing number of gates and inputs
- Minimizing propagation delays
- Simplifying circuit layout

2 Karnaugh Map Method

2.1 Basic Principles

- Visual representation of truth tables
- Adjacent cells differ by one variable (Gray code)
- Effective for functions with 2-5 variables
- Two standard forms:
  - Sum-of-Products (SOP)
  - Product-of-Sums (POS)



fi

Figure 1: 2-variable K-map structure and minimization example

3 2-Variable K-Maps

3.1 Structure and Minimization

- 4 cells representing minterms m0-m3
- Horizontal axis: variable x
- Vertical axis: variable y

Example Simplification:

$$F(x, y) = \sum(0, 1, 3) \Rightarrow x' + y \tag{1}$$

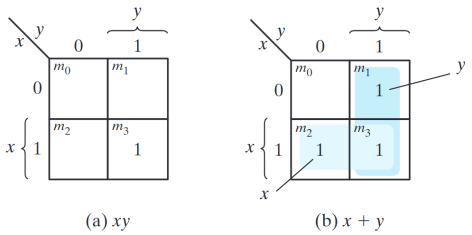


Figure 2: 2-variable K-map example minimization

4 3-Variable K-Maps

4.1 Layout and Adjacency Rules

- 8 cells arranged in 2x4 grid
- Variables typically ordered as x,y,z
- Wrapping adjacency between left-right columns

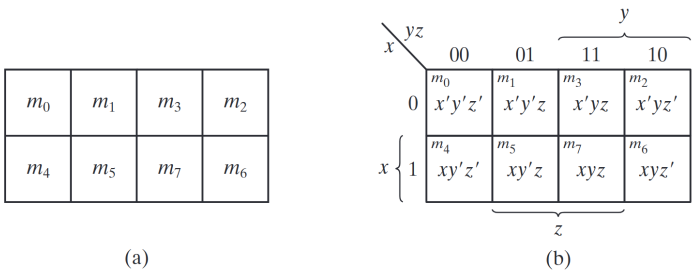


Figure 3: 3-variable K-map structure

4.2 Minimization Examples

1. $F = \sum(2, 3, 4, 5) \Rightarrow x'y + xy'$
2. $F = \sum(3, 4, 6, 7) \Rightarrow yz + xz'$
3. $F = \sum(0, 2, 4, 5, 6) \Rightarrow z' + xy'$

5 4-Variable K-Maps

5.1 Structure and Simplification

- 16 cells arranged in 4x4 grid
- Both horizontal and vertical wrapping
- Typical variable order: w,x,y,z

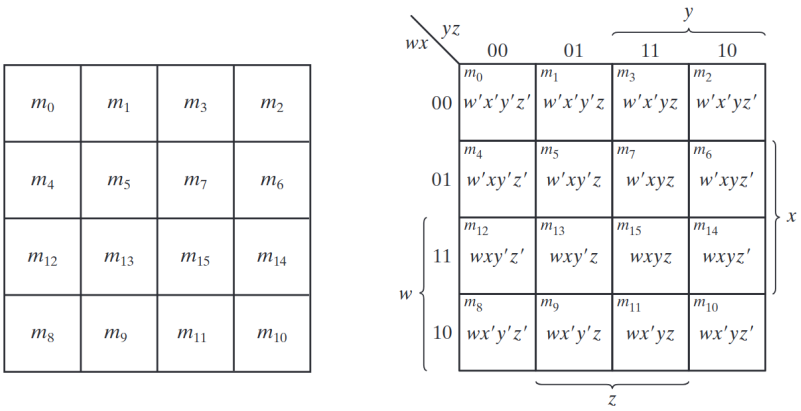


Figure 4: 4-variable K-map layout

Example Minimization:

$$F = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14) \Rightarrow y' + w'z' + xz' \tag{2}$$

6 Prime Implicants

6.1 Key Concepts

- **Prime Implicant**: Maximal group of adjacent 1s
- **Essential PI**: Contains at least one unique minterm
- Non-essential PIs: Can be replaced by alternative combinations

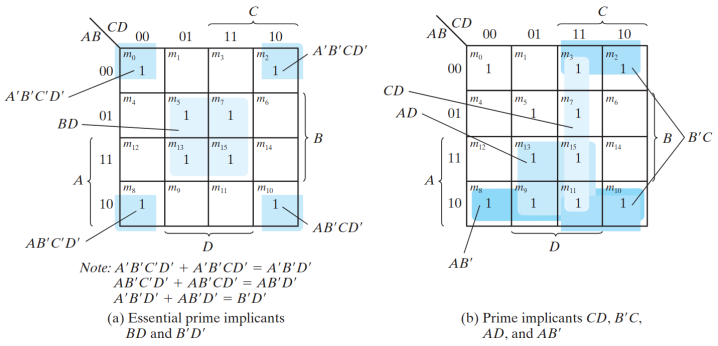


Figure 5: Prime implicant selection example with multiple solutions

7 Product-of-Sums Simplification

7.1 SOP vs POS Implementation

Sum-of-Products (SOP)	Product-of-Sums (POS)
AND-OR structure	OR-AND structure
Minimize 1s in K-map	Minimize 0s in K-map
Direct implementation	Complemented implementation

Table 1: Comparison of SOP and POS implementations

8 Don't Care Conditions

8.1 Handling Incomplete Functions

- X-marked cells represent don't care conditions
- Can be treated as 0 or 1 for better minimization
- Particularly useful in BCD and error-checking circuits

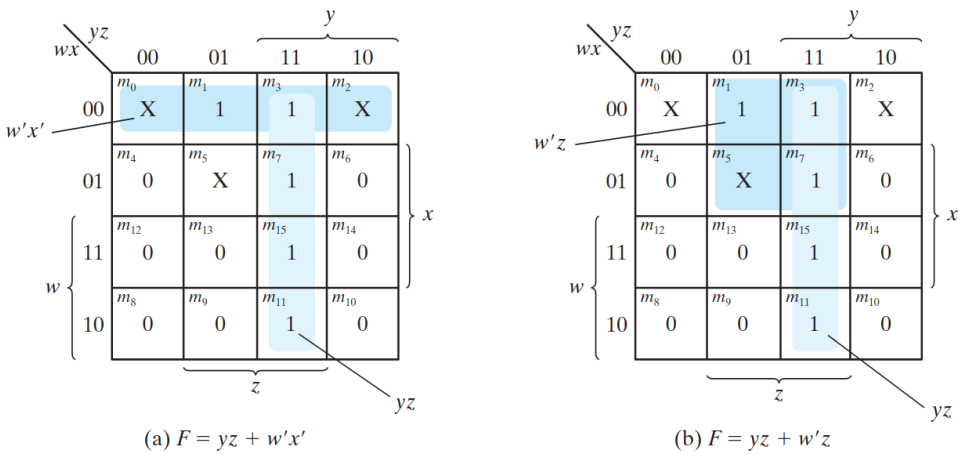


Figure 6: Don't care condition utilization example

9 Conclusion

Key takeaways from gate-level minimization:

- K-maps provide systematic visual simplification
- Proper identification of prime implicants is crucial
- Don't care conditions enable more efficient designs
- Fundamental for both manual design and automated tools