Quine-McCluskey Method

CS341 Digital Logic & Computer Organization

Introduction

- A tabular procedure.
- Algorithm can be programmed.
- Minimizing principles are the same as for the K-maps, i.e., based on logical adjacency property to reduce the Boolean function (x + x’ = 1).
- Compares minterms with all others and combines them if possible.

An example that illustrates the steps of the algorithm

- Simplify $F(w, x, y, z) = wx'y'z'+wxy'z'+w'xyz'+wxy'z+w'x'yz+wx'y'z+w'x'y'z+w'x'y'z'$

- Note: don't cares should be included for possible further simplification.

Step 1: convert minterms to binary form

wx’y’z’+wxy’z’+w’xyz’+wxy’z+w’x’yz+wx’y’z+w’x’y’z+w’x’y’z’

1000 + 1100 + 0110 + 1101 + 0011 + 1001 +0001 + 0000

Step 2: Arrange the binary representations in a table with different number of 1s in each group

	minterm	sub#
Group 1: zero 1s	0	0000

Group 2: one 1	1	0001
	8	1000

Group 3: two 1s	12	1100
	6	0110
	3	0011
	9	1001

Group 4: three 1s	13	1101

Step 3: Perform matches between table entries

- Two term match if they differ in exactly one position.
- A new term is formed with a dash ‘-’ substituted in the position where the two binary value differ. The new terms are them arranged according to step 2.
- Of these new terms, a match occurs if two terms differ in exactly one position and have a ‘-’ in the same position. Again, a ‘-’ is substituted in the position where the terms differ, and this procedure is repeated until no more matches are found.
- Terms involving a match is “checked”. The unchecked terms are called ***prime implicants*** or PIs. All possible matches must be considered. Repeated terms need not be copied.

Step 3: Illustration: performing matches between table entries

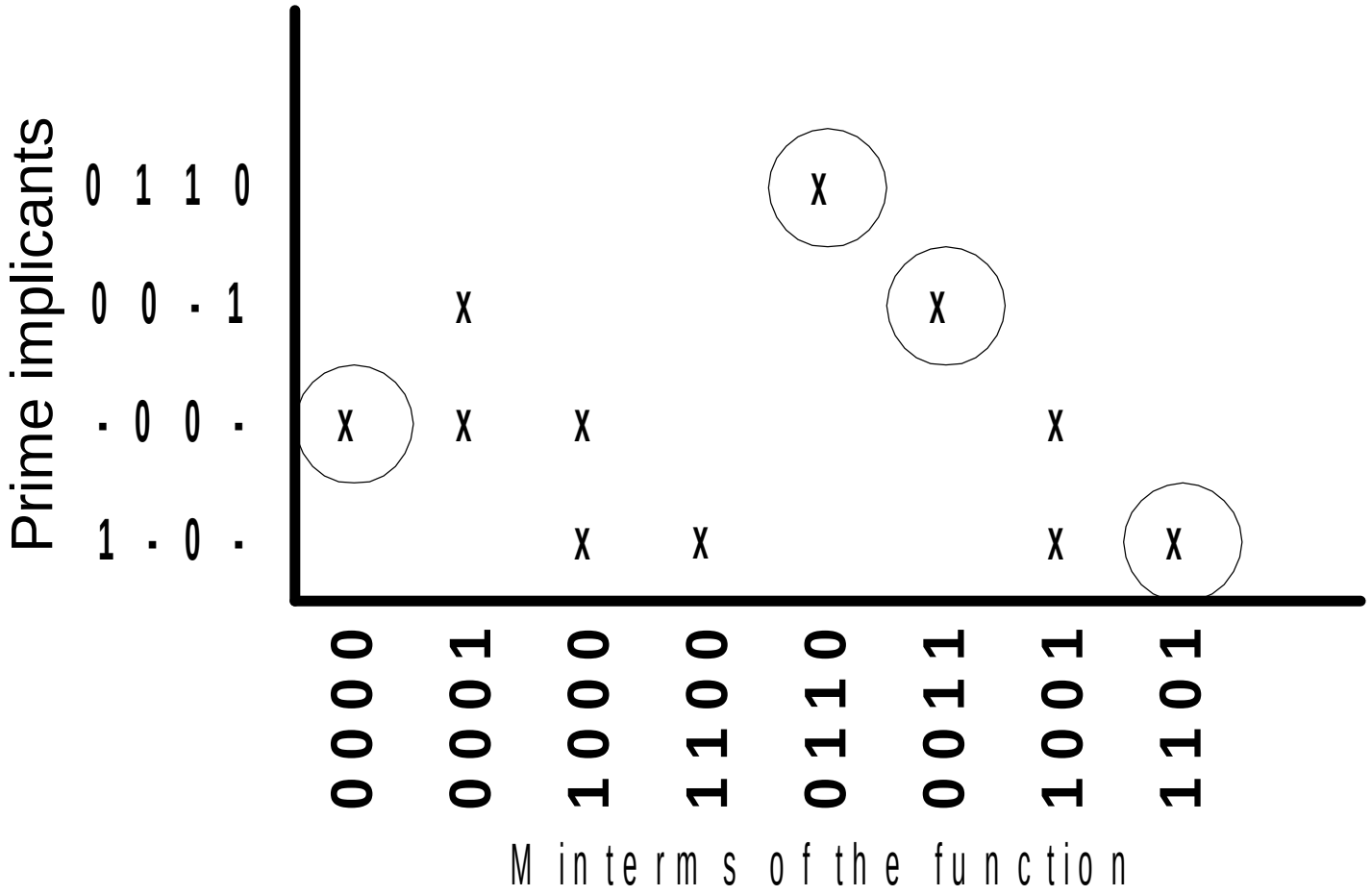
minterm sub#							
0	x	0000					

1	x	0001	x	000- (0,1)			
8	x	1000	x	-000 (0,8)			
		---		---			
12	x	1100	**PI**	**00-1** (1,3)	**PI**	**-00-**	(0,1;8,9)
6	**PI**	**0110**	x	-001 (1,9)			
3	x	0011	x	1-00 (8,12)			
9	x	1001	x	100- (8,9)			
		---		---		---	
13	x	1101	x	110- (12,13)	**PI**	**1-0-**	(8,9;12,13)
			x	1-01 (9,13)			

Step 4: Construct a prime implicant table or graph

- List all minterms of the function along the x-axis.
- List all prime implicants or PIs along the y-axis.
- Place an x's at intersections where a prime implicant matches the minterm in each binary value (treat dashes as don't-cares). If a prime implicant agrees with a minter in each binary position, the **PI *covers*** the minterm.

Step 4: Illustration-constructing a prime implicant graph



Step 5: Choose a minimum set of PIs so that each minterm of the function is covered

- In this example, all PIs are ***essential***, because each ***covers*** at least a minterm (the x's that are circled in the graph) that is not covered by other PIs.
- So the simplified function corresponds to

  0110 + 00-1 + -00- + 1-0- which corresponds to

  w'xyz' + w'x'z + x'y' + wy'

Covering problem: additional steps may be necessary

- In the previous example, all PIs are essential and were all included in the final simplified function. In general, the process of selecting a minimum number of PIs to cover a function is not as simple.
- The following slides describe rules that can be used to derive the minimum number of PIs when the PI graph is complex.

Dominating row or column

- Definition: row (column) i of a PI graph dominates row (column) j if row (column) i contains an x in each column (row) in which row (column) j contains x.

Rule 1 may help reduce the complexity of the PI graph

- Because a row of the PI chart corresponds to a PI, a dominating row covers all the minterns that are covered by a row dominated by it. Hence the follow rule.
- Rule 1: A row dominated by another row can be eliminated from the chart, provided the PI corresponding to the dominated row does not have fewer literals compared with the PI corresponding to the dominating row. When several identical rows are present in a chart, all but the one whose PI has the fewest literals can be eliminated.

Rule 2: may help reduce the complexity of the PI graph

- When a column dominates another column, the PIs that cover the minterm corresponding to the dominated column also cover the minterm corresponding to the dominating column. Hence:
- Rule 2: A column dominating another column can be eliminated. All but one of the identical columns can be eliminated.

A note on *cyclic PI graph*

- A cyclic graph is a graph that contain no essential PIs and cannot be reduced by the rules just discussed.
- In these cases, a row corresponding to a PI with a minimum number of literals is first selected (arbitrarily). The row corresponding to that PI and columns corresponding to the minterms covered by the PI are then removed from the chart. If possible, the two-step procedure shown above is then applied. If the chart remains cyclic after the selection of the first PI, another PI is selected, again arbitrarily. A minimum cover is obtained by repeating the process of selection a row and applying the two-step procedure.

Terminology revisited

- Implicant: a product term of a function if the function has a value of 1 for all minterms of the product term.
- Prime implicant or PI: a product term that corresponds to a valid largest possible subgroup of logically adjacent 1’s. In other words, a PI cannot be contained by another PI.
- Essential PI: a PI that includes a minterm not present in other PIs. It must be included as a term in the final simplified function.

Summary: general procedure for finding a minimum cover for a Boolean function

1. Select all essential PIs. If these PIs covers all the minterms, stop; otherwise, go to the next step.
2. Apply rules 1 and 2 to eliminate redundant rows and columns from the PI graph of non-essential PIs. When the chart is thus reduced, some PIs will become essential. Go back to step 1.

Additional HW to chapter 2

- Use Quine-McCluskey algorithm to simplify

  f(w, x, y, z) =

  w’x’y’z’+w’x’y’z+w’x’yz’+wx’yz+wxy’z’

  +w’xy’z+w’x’yz+wxy’z+wxyz’

Show detailed steps.

Optional extra point programming project

- Implement the Quine-McCluskey algorithm with C++ or other programming languages such as Visual Basic.