CS1026 – Digital Logic Design

Quine-McCluskey Algorithm Example I

Shane Sheehan [1]

[1]ADAPT
Trinity College Dublin

January 29, 2017

Today's Overview

Quine-McCluskey Overview I

An exact algorithm which finds:

- A minimum-cost Sum-of-Products (SoP) implementation [Majumder et al., 2015]

for a Boolean function.

Quine-McCluskey Overview II

Main steps in the Quine-McCluskey algorithm:

1. Generate Prime Implicants
2. Construct Prime Implicant Table
3. Reduce Prime Implicant Table
    1. Remove Essential Prime Implicants
    2. Row Dominance
    3. Column Dominance
4. Solve Prime Implicant Table

Quine-McCluskey Overview III

In Step 1, we generate the prime implicants of a function using an iterative procedure:

1. List mid terms in ascending order
2. Merge row to remove non-prime implicants

Quine-McCluskey Overview IV

In Step 2, we make a prime implicant table

1. Columns denote the prime implicants
2. Rows denotes the ON-set (1) minterms

Remember

We need cover all the rows using a minimum-cost cover of prime implicants.

Quine-McCluskey Overview V

Minimum cost?

- Have fewest prime implicants (i.e. AND gates)

However, we could consider more complex cost functions

- Power optimization, etc.

But we don't in this course.. ;-)

Quine-McCluskey Overview VI

Iterative Reduction step (Step 3) reduces the size of the table.

- Crossing out rows and columns in the table
- until no further table reduction can occur.

At this point, we hopefully have an *empty* reduced table.

- Now remove essential prime implicants to find *minimum-cost solution*

Quine-McCluskey Overview VII

However, if the reduced table is not empty, it become necessary to solve the table (Step 4).

- We normally use Petrick's method
- A Branch and bound method

See next lecture

Quine-McCluskey Example I

So.. our easy example:
$F(x_0, x_1, x_2, x_3) = \sum m(1, 2, 5, 12, 14)$

- 4 variables: A, B, C and D
- This just describes the K-Map

Quine-McCluskey Example II

- The truth table
- $F(x_0, x_1, x_2, x_3) = \sum m(1, 2, 5, 12, 14)$

	x_3	x_2	x_1	x_0	y
0:	0	0	0	0	0
1:	0	0	0	1	1
2:	0	0	1	0	1
3:	0	0	1	1	0
4:	0	1	0	0	0
5:	0	1	0	1	1
6:	0	1	1	0	0
7:	0	1	1	1	0
8:	1	0	0	0	0
9:	1	0	0	1	0
10:	1	0	1	0	0
11:	1	0	1	1	0
12:	1	1	0	0	1
13:	1	1	0	1	0
14:	1	1	1	0	1
15:	1	1	1	1	0

Quine-McCluskey Example III

Iteration 0 (Row Dominance)

	x_3	x_2	x_1	x_0	
1:	0	0	0	1	$\rightarrow$
2:	0	0	1	0	✓
5:	0	1	0	1	$\rightarrow$
12:	1	1	0	0	$\rightarrow$
14:	1	1	1	0	$\rightarrow$

- Implicants:
  - Prime – ✓
  - Non-prime – $\rightarrow$

Quine-McCluskey Example IV

Iteration 1 (Row Dominance)

	x_3	x_2	x_1	x_0	
1, 5:	0	-	0	1	✓
12, 14:	1	1	-	0	✓

- Implicants:
  - Prime – ✓
  - Non-prime – $\rightarrow$

Quine-McCluskey Example V

We cannot do any Column Dominance here.. So to the prime implicant chart:

	x_3	x_2	x_1	x_0	1	2	5	12	14
1, 5:	0	-	0	1	●		●		
12, 14:	1	1	-	0				●	●
2:	0	0	1	0		●			

Green dots indicate the *Essential prime implicants*

Quine-McCluskey Example VI

	x_3	x_2	x_1	x_0	1	2	5	12	14
1, 5:	0	-	0	1	●		●		
12, 14:	1	1	-	0				●	●
2:	0	0	1	0		●			

Extracted essential prime implicants:

- $x_3' x_1' x_0$
- $x_3 x_2 x_0'$
- $x_3' x_2' x_1 x_0'$

Quine-McCluskey Example VII

	x_3	x_2	x_1	x_0	1	2	5	12	14
1, 5:	0	-	0	1	●		●		
12, 14:	1	1	-	0				●	●
2:	0	0	1	0		●			

The Sum Of Products (SOP)

$(x_3' x_1' x_0) + (x_3 x_2 x_0') + (x_3' x_2' x_1 x_0')$

Done! :-)

Any now relax

Next time.. What happens if have don't cares?

- Makes the algorithm even easier

Any Problems?

- Ask!
- E-Mail: *Sheehas1@scss.tcd.ie*
- LinkedIn: www.linkedin.com/in/shane-sheehan-1ab534b9

References (Homework) I

Majumder, A., Chowdhury, B., Mondai, A. J., and Jain, K. (2015).
Investigation on quine mccluskey method: A decimal manipulation based novel approach for the minimization of boolean function.
In *Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015 International Conference on*, pages 18–22. IEEE.