CS1026 – Digital Logic Design

Quine-McCluskey Algorithm Example II

Alistair Morris [1]

[1]Distributed Systems Group
Trinity College Dublin

February 14, 2016

Today's Overview

Quine-McCluskey Overview I

So a few more examples where we fins:

- A minimum-cost sum-of-products 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

Simple Example I

$F(x_0.x_1, x_2) = \sum m(0, 1, 2, 4, 5, 7)$

	x_2	x_1	x_0	y
0:	0	0	0	1
1:	0	0	1	1
2:	0	1	0	1
3:	0	1	1	0
4:	1	0	0	1
5:	1	0	1	1
6:	1	1	0	0
7:	1	1	1	1

Simple Example II

	x_2	x_1	x_0	
0:	0	0	0	$\rightarrow$
1:	0	0	1	$\rightarrow$
2:	0	1	0	$\rightarrow$
4:	1	0	0	$\rightarrow$
5:	1	0	1	$\rightarrow$
7:	1	1	1	$\rightarrow$

Iteration 0:

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

Simple Example III

	x_2	x_1	x_0	
0, 1:	0	0	-	→
0, 2:	0	-	0	✓
0, 4:	-	0	0	→
1, 5:	-	0	1	→
4, 5:	1	0	-	→
5, 7:	1	-	1	✓

Iteration 1:

- ✓ – Prime Implicants
- → – Non Prime Implicants

Simple Example IV

	x_2	x_1	x_0	
0, 1, 4, 5:	-	0	-	✓

Iteration 2:

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

Simple Example V

Prime Implicant Table:

	x_2	x_1	x_0	0	1	2	4	5	7
0, 1, 4, 5:	-	0	-	○	●		●	○	
0, 2:	0	-	0	○		●			
5, 7:	1	-	1					○	●

We have..

- x_1'
- $x_2'x_0'$
- x_2x_0

Simple Example VI

Prime Implicant Table:

	x_2	x_1	x_0	0	1	2	4	5	7
0, 1, 4, 5:	-	0	-	○	●		●	○	
0, 2:	0	-	0	○		●			
5, 7:	1	-	1					○	●

So we get the SOP:

- $x_1' + x_2'x_0' + x_2x_0$

Don't Care Example I

	x_2	x_1	x_0	y
0:	0	0	0	0
1:	0	0	1	×
2:	0	1	0	0
3:	0	1	1	×
4:	1	0	0	1
5:	1	0	1	0
6:	1	1	0	0
7:	1	1	1	0

	x_2	x_1	x_0	
1:	0	0	1	→
3:	0	1	1	→
4:	1	0	0	✓

	x_2	x_1	x_0	
1, 3:	0	-	1	(×)

$$F(x_0, x_1, x_2.x_3) = \sum m(4) + \sum d(1,3)$$

- x denotes *Don't Cares*

Don't Care Example II

	x_2	x_1	x_0	4
4:	1	0	0	●

Nice and easy:

- $y = x_2 x_1' x_0'$

Solving the Table I

	x_2	x_1	x_0	y
0:	0	0	0	0
1:	0	0	1	1
2:	0	1	0	×
3:	0	1	1	×
4:	1	0	0	1
5:	1	0	1	1
6:	1	1	0	×
7:	1	1	1	0

	x_2	x_1	x_0	
1:	0	0	1	→
2:	0	1	0	→
3:	0	1	1	→
4:	1	0	0	→
5:	1	0	1	→
6:	1	1	0	→

	x_2	x_1	x_0	
1, 3:	0	-	1	✓
1, 5:	-	0	1	✓
2, 3:	0	1	-	(×)
2, 6:	-	1	0	(×)
4, 5:	1	0	-	✓
4, 6:	1	-	0	✓

$$F(x_0, x_1, x_2.x_3) = \sum m(1, 4, 5) + \sum d(2, 3, 6)$$

- x denotes *Don't Cares*

Solving the Table II

	x_2	x_1	x_0	1	4	5
1, 3:	0	-	1	○		
1, 5:	-	0	1	○		○
4, 5:	1	0	-		○	○
4, 6:	1	-	0		○	

Whoops.. What to do?

Solving the Table III

	x_2	x_1	x_0	1	4	5
1, 3:	0	-	1	○		
1, 5:	-	0	1	○		○
4, 5:	1	0	-		○	○
4, 6:	1	-	0		○	

Use Petrick's method

- Determine all minimum Sum-Of-Products (SOP) solutions

Solving the Table IV

1. Label the rows of the reduced prime implicant chart P_1, P_2, P_3, P_4, etc.
2. Form a logical function P which is true when all the columns are covered.
3. Reduce P to a minimum sum of products by multiplying out and applying $X + XY = X$.
4. Each term in the result represents a solution, that is, a set of rows which covers all of the minterms in the table. To determine the minimum solutions, first find those terms which contain a minimum number of prime implicants.
5. Next, for each of the terms found in step five, count the number of literals

Solving the Table V

6. Choose the term or terms composed of the minimum total number of literals, and write out the corresponding sums of prime implicants.

Remember

- P consists of a product of sums where each sum term has the form $(P_{i0} + P_{i1} + \cdots + P_{iN})$, where each P_{ij} represents a row covering column i.
- You saw this process with SAR

Solving the Table VI

	x_2	x_1	x_0	1	4	5
1, 3:	0	-	1	○		
1, 5:	-	0	1	○		○
4, 5:	1	0	-		○	○
4, 6:	1	-	0		○	

We have the implicants:

- $x_2' x_0 \equiv p_0$
- $x_1' x_0 \equiv p_1$
- $x_2 x_1' \equiv p_2$
- $x_2 x_0' \equiv p_3$

Solving the Table VII

$$(p_0 + p_1)(p_2 + p_3)(p_1 + p_2) \equiv (p_0p_2 + p_0p_3 + p_1p_2)(p_1p_2)$$
$$\equiv (p_0p_2 + p_1p_2 + p_1p_3)(p_1p_2)$$
$$\equiv (p_0p_1p_2 + p_0p_2 + p_0p_2p_3 + p_1p_2 + p_1p_3 + p_1p_2p_3$$
$$\equiv (p_0p_2 + p_1p_2 + p_1p_3)$$

Minimal boolean Expression

$$y = (x_2'x_0) + (x_2x_1')$$

Painful.. a computer does not mind (if you have a lot of time)!

Any now relax

Next time.. Flip Flops and Latches

- No more simiplication! zzz

Any Problems?

- Ask!
- E-Mail: *morrisa5@scss.tcd.ie*
- LinkedIn: `linkedin.com/in/alistair-morris-9712b247`

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.