CSEE E6861y **The Quine-McCluskey Method** Handout 5

Prof. Steven Nowick January 21, 2016

Introduction

The Quine-McCluskey method is an exact algorithm which finds a minimum-cost sum-of-products implementation of a Boolean function. This handout introduces the method and applies it to several examples.

There are 4 main steps in the Quine-McCluskey algorithm:

1. Generate Prime Implicants
2. Construct Prime Implicant Table
3. Reduce Prime Implicant Table
    (a) Remove Essential Prime Implicants
    (b) Row Dominance
    (c) Column Dominance
4. Solve Prime Implicant Table

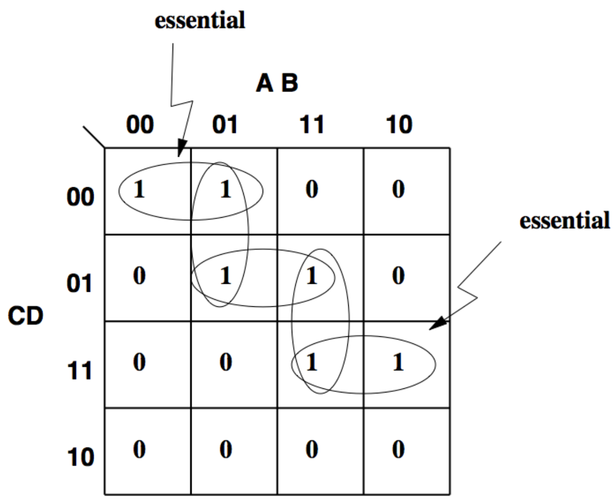
Note: For this course, you are not responsible for Step #1 on this handout: the method for generating all prime implicants of a Boolean function. You can look over this method, but you will be learning, and be responsible for, a more powerful modern prime-generation technique later in this course.

In Step #1, the prime implicants of a function are generated using an iterative procedure. In Step #2, a prime implicant table is constructed. The columns of the table are the prime implicants of the function. The rows are minterms of where the function is 1, called *ON-set minterms*. The goal of the method is to cover all the rows using a minimum-cost *cover* of prime implicants.

In particular, 'minimum-cost' for this handout means to have fewest prime implicants (i.e. AND gates) in the final solution. However, the algorithm has been extended to consider more complex cost functions, such as minimizing the total number of gate inputs, power optimization, and so on.

The reduction step (Step #3) is used to reduce the size of the table. This step has three sub-steps **which are iterated until no further table reduction is possible.** At this point, the reduced table is either (i) empty or (ii) non-empty. If the reduced table is empty, the removed essential prime implicants form a minimum-cost solution. However, if the reduced table is *not* empty, the table must be "solved" (Step #4). The table can be solved using either "Petrick's method" or the "branching method". This handout focuses on Petrick's method. The branching method is discussed in the books by McCluskey, Roth, etc., but you will not be responsible for the branching method.

The remainder of this handout illustrates the details of the Quine-McCluskey method on 3 examples. Example #1 is fairly straightforward, Examples #2 is more involved, and Example #3 applies the method to a function with "don't-cares". But first, we motivate the need for *column dominance* and *row dominance*.



Karnaugh map with set of prime implicants: illustrating "column dominance"

Column Dominance

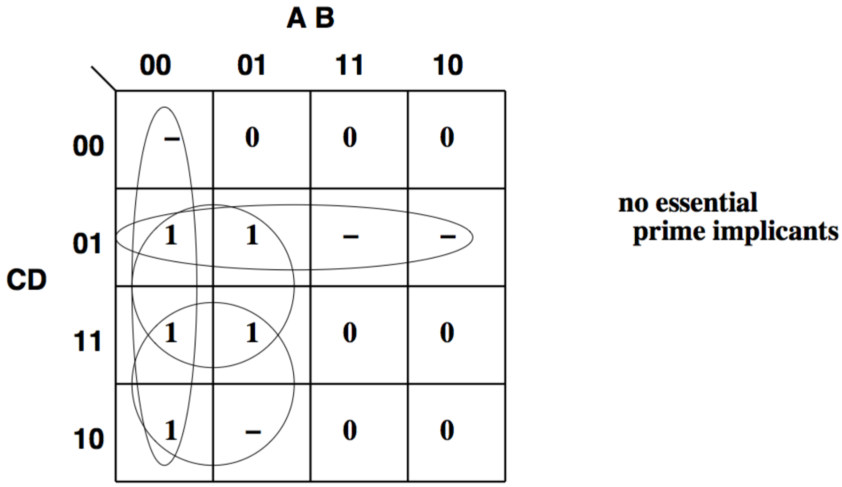
Consider the following Karnaugh map of a 4-input Boolean function:

There are 5 prime implicants, each of which covers 2 ON-set minterms. First, we note that two implicants are *essential prime implicants:* $A'C'D'$ and ACD. These implicants must be added to the final cover. There are 3 remaining prime implicants. We must pick a minimum subset of these to cover the uncovered ON-set minterms.

Here is the prime implicant table for the Karnaugh map. The 5 prime implicants are listed as columns, and the 6 ON-set minterms are listed as rows.

	$A'C'D'$ (0,4)	$A'BC'$ (4,5)	$BC'D$ (5,13)	ABD (13,15)	ACD (11,15)
0	X				
4	X	X			
5		X	X		
11					X
13			X	X	
15				X	X

We cross out columns $A'C'D'$ and ACD and mark them with asterisks, to indicate that these are essential. Each row intersected by one of these columns is also crossed out, because that minterm is now covered. At this point, prime implicant $BC'D$ covers 2 remaining ON-set minterms (5 and 13). However, prime implicant $A'BC'$ covers only one of these (namely, 5), as does ABD (namely, 13). Therefore we can *always* use $BC'D$ instead of either $A'BC'$ or ABD, since it covers the same minterms. That is, $BC'D$ **column-dominates** $A'BC$, and $BC'D$ **column-dominates** ABD. The *dominated* prime implicants can be crossed out, and only column $BC'D$ remains.



Karnaugh map with set of prime implicants:
illustrating "row dominance"

Row Dominance

Consider the following Karnaugh map of a 4-input Boolean function:

There are 4 prime implicants: $A'B'$, $C'D$, $A'D$ and $A'C$. None of these is an essential prime implicant. We must pick a minimum subset of these to cover the 5 ON-set minterms. Here is the prime implicant table for the Karnaugh map. The 4 prime implicants are listed as columns, and the 5 ON-set minterms are listed as rows.

	$A'B'$ (1,2,3)	$C'D$ (1,5)	$A'D$ (1,3,5,7)	$A'C$ (2,3,7)
1	X	X	X	
2	X			X
3	X		X	X
5		X	X	
7			X	X

Note that row 3 is contained in three columns: $A'B'$, $A'D$, and $A'C$. Row 2 is covered by two of these three columns: $A'B'$ and $A'C$, and row 7 is also covered by two of these three columns: $A'D$ and $A'C$. In this case, *any prime implicant* which contains row 2 also contains row 3. Similarly, *any prime implicant* which contains row 7 also contains row 3. Therefore, we can ignore the covering of row 3: it will always be covered as long as we cover row 2 or row 7. To see this, note that row 3 **row dominates** row 2, and row 3 **row dominates** row 7. The situation is now the reverse of column dominance: **we cross out the dominating (larger) row**. In this case, row 3 can be crossed out; it no longer needs to be considered.

Similarly, row 1 row dominates row 5. Therefore row 1 can be crossed out. We are guaranteed that row 1 will still be covered, since any prime implicant which covers row 5 will also cover row 1.

Example #1:

$$F(A, B, C, D) = \Sigma m(0, 2, 5, 6, 7, 8, 10, 12, 13, 14, 15)$$

The Notation. The above notation is a shorthand to describe the Karnaugh map for F. First, it indicates that F is a Boolean function of 4 variables: A, B, C, and D. Second, each *ON-set minterm* of F is listed above, that is, minterms where the function is 1: 0, 2, 5, ... Each of these numbers corresponds to one entry (or square) in the Karnaugh map. For example, the decimal number 2 corresponds to the minterm $ABCD = 0010$, (0010 is the binary representation of 2). That is, $ABCD = 0010$ is an ON-set minterm of F; *i.e.,* it is a 1 entry. All remaining minterms, not listed above, are assumed to be 0.

Step 1: Generate Prime Implicants.

Note: You should learn this basic method for generating prime implicants (Step #1), but I will not ask you to reproduce it. See Roth book on reserve for more details. (Instead, you will soon learn the more advanced fast recursive algorithm for prime generation.)

List Minterms

Column I	
0	0000
2	0010
8	1000
5	0101
6	0110
10	1010
12	1100
7	0111
13	1101
14	1110
15	1111

Combine Pairs of Minterms from Column I

A check ($\surd$) is written next to every minterm which can combined with another minterm.

Column I			*Column II*	
0	0000	√	(0,2)	00-0
2	0010	√	(0,8)	-000
8	1000	√	(2,6)	0-10
5	0101	√	(2,10)	-010
6	0110	√	(8,10)	10-0
10	1010	√	(8,12)	1-00
12	1100	√	(5,7)	01-1
7	0111	√	(5,13)	-101
13	1101	√	(6,7)	011-
14	1110	√	(6,14)	-110
15	1111	√	(10,14)	1-10
			(12,13)	110-
			(12,14)	11-0
			(7,15)	-111
			(13,15)	11-1
			(14,15)	111-

Combine Pairs of Products from Column II

A check (√) is written next to every product which can combined with another product.

Column I			*Column II*			*Column III*	
0	0000	√	(0,2)	00-0	√	(0,2,8,10)	-0-0
2	0010	√	(0,8)	-000	√	(0,8,2,10)	-0-0
8	1000	√	(2,6)	0-10	√	(2,6,10,14)	–10
5	0101	√	(2,10)	-010	√	(2,10,6,14)	–10
6	0110	√	(8,10)	10-0	√	(8,10,12,14)	1–0
10	1010	√	(8,12)	1-00	√	(8,12,10,14)	1–0
12	1100	√	(5,7)	01-1	√	(5,7,13,15)	-1-1
7	0111	√	(5,13)	-101	√	(5,13,7,15)	-1-1
13	1101	√	(6,7)	011-	√	(6,7,14,15)	-11-
14	1110	√	(6,14)	-110	√	(6,14,7,15)	-11-
15	1111	√	(10,14)	1-10	√	(12,13,14,15)	11–
			(12,13)	110-	√	(12,14,13,15)	11–
			(12,14)	11-0	√		
			(7,15)	-111	√		
			(13,15)	11-1	√		
			(14,15)	111-	√		

Column III contains a number of duplicate entries, *e.g.* (0,2,8,10) and (0,8,2,10). Duplicate entries appear because a product in Column III can be formed in several ways. For example, (0,2,8,10) is formed by combining products (0,2) and (8,10) from Column II, and (0,8,2,10) (the same product) is formed by combining products (0,8) and (2,10).

Duplicate entries should be crossed out. The remaining unchecked products cannot be combined with other products. These are the prime implicants: (0,2,8,10), (2,6,10,14), (5,7,13,15), (6,7,14,15), (8,10,12,14) and (12,13,14,15); or, using the usual product notation: $B'D'$, CD', BD, BC, AD' and AB.

Step 2: Construct Prime Implicant Table.

	$B'D'$ (0,2,8,10)	CD' (2,6,10,14)	BD (5,7,13,15)	BC (6,7,14,15)	AD' (8,10,12,14)	AB (12,13,14,15)
0	X					
2	X	X				
5			X			
6		X		X		
7			X	X		
8	X				X	
10	X	X			X	
12					X	X
13			X			X
14		X		X	X	X
15			X	X		X

Step 3: Reduce Prime Implicant Table.

Iteration #1.

(i) Remove Primary Essential Prime Implicants

	$B'D'(*)$ (0,2,8,10)	CD' (2,6,10,14)	$BD(*)$ (5,7,13,15)	BC (6,7,14,15)	AD' (8,10,12,14)	AB (12,13,14,15)
(∘)0	X					
2	X	X				
(∘)5			X			
6		X		X		
7			X	X		
8	X				X	
10	X	X			X	
12					X	X
13			X			X
14		X		X	X	X
15			X	X		X

* indicates an essential prime implicant

∘ indicates a distinguished row, i.e. a row covered by only 1 prime implicant

In step #1, *primary essential prime implicants* are identified. These are implicants which will appear in *any* solution. A row which is covered by only 1 prime implicant is called a *distinguished row*. The prime implicant which covers it is an *essential prime implicant*. In this step, essential prime implicants are identified and removed. The corresponding column is crossed out. Also, each row where the column contains an X is completely crossed out, since these minterms are now covered. These essential implicants will be added to the final solution. In this example, $B'D'$ and BD are both primary essentials.

(ii) Row Dominance

The table is simplified by removing rows and columns which were crossed out in step (i). (*Note:* you do not need to do this, but it makes the table easier to read. Instead, you can continue to mark up the original table.)

	CD' (2,6,10,14)	BC (6,7,14,15)	AD' (8,10,12,14)	AB (12,13,14,15)
6	X	X		
12			X	X
14	X	X	X	X

Row 14 *dominates* both row 6 and row 12. That is, row 14 has an "X" in every column where row 6 has an "X" (and, in fact, row 14 has "X"'s in other columns as well). Similarly, row 14 has in "X" in every column where row 12 has an "X". Rows 6 and 12 are said to be *dominated by* row 14.

A *dominating* row can always be eliminated. To see this, note that every product which covers row 6 also covers row 14. That is, if some product covers row 6, row 14 is *guaranteed* to be covered. Similarly, any product which covers row 12 will also cover row 14. Therefore, row 14 can be crossed out.

(iii) Column Dominance

	CD' (2,6,10,14)	BC (6,7,14,15)	AD' (8,10,12,14)	AB (12,13,14,15)
6	X	X		
12			X	X

Column CD' *dominates* column BC. That is, column CD' has an "X" in every row where column BC has an "X". In fact, in this example, column BC also dominates column CD', so each is *dominated by* the other. (Such columns are said to *co-dominate* each other.) Similarly, columns AD' and AB dominate each other, and each is dominated by the other.

A *dominated* column can always be eliminated. To see this, note that every row covered by the dominated column is also covered by the dominating column. For example, $C'D$ covers every row which BC covers. Therefore, the dominating column can always replace the dominated column, so the dominated column is crossed out. In this example, CD' and BC dominate each other, so either column can be crossed out (but not both). Similarly, AD' and AB dominate each other, so either column can be crossed out.

Iteration #2.

(i) Remove Secondary Essential Prime Implicants

	$CD'(**)$ (2,6,10,14)	$AD'(**)$ (8,10,12,14)
(∘)6	X	
(∘)12		X

** indicates a secondary essential prime implicant

∘ indicates a distinguished row

In iteration #2 and beyond, *secondary essential prime implicants* are identified. These are implicants which will appear in *any* solution, *given* the choice of column-dominance used in the previous steps (if 2 columns co-dominated each other in a previous step, the choice of which was deleted can affect what is an "essential" at this step). As before, a row which is covered by only 1 prime implicant is called a *distinguished row*. The prime implicant which covers it is a *(secondary) essential prime implicant.*

Secondary essential prime implicants are identified and removed. The corresponding columns are crossed out. Also, each row where the column contains an X is completely crossed out, since these minterms are now covered. These essential implicants will be added to the final solution. In this example, both CD' and AD' are secondary essentials.

Step 4: Solve Prime Implicant Table.

No other rows remain to be covered, so no further steps are required. Therefore, the minimum-cost solution consists of the primary and secondary essential prime implicants $B'D'$, BD, CD' and AD':

$$F = B'D' + BD + CD' + AD'$$

Example #2:

$$F(A,B,C,D) = \Sigma m(0,2,3,4,5,6,7,8,9,10,11,12,13)$$

Step 1: Generate Prime Implicants.

Use the method described in Example #1.

Step 2: Construct Prime Implicant Table.

	$A'D'$	$B'D'$	$C'D'$	$A'C$	$B'C$	$A'B$	BC'	AB'	AC'
0	X	X	X						
2	X	X		X	X				
3				X	X				
4	X		X			X	X		
5						X	X		
6	X			X		X			
7				X		X			
8		X	X					X	X
9								X	X
10		X			X			X	
11					X			X	
12			X				X		X
13							X		X

Step 3: Reduce Prime Implicant Table.

Iteration #1.

(i) Remove Primary Essential Prime Implicants

There are no primary essential prime implicants: each row is covered by at least two products.

(ii) Row Dominance

	$A'D'$	$B'D'$	$C'D'$	$A'C$	$B'C$	$A'B$	BC'	AB'	AC'
0	X	X	X						
2	X	X		X	X				
3				X	X				
4	X		X			X	X		
5						X	X		
6	X			X		X			
7				X		X			
8		X	X					X	X
9								X	X
10		X			X			X	
11					X			X	
12			X				X		X
13							X		X

There are many instances of row dominance. Row 2 dominates 3, 4 dominates 5, 6 dominates 7, 8 dominates 9, 10 dominates 11, 12 dominates 13. Dominating rows are removed.

(iii) Column Dominance

	$A'D'$	$B'D'$	$C'D'$	$A'C$	$B'C$	$A'B$	BC'	AB'	AC'
0	X	X	X						
3				X	X				
5						X	X		
7				X		X			
9								X	X
11					X			X	
13							X		X

Columns $A'D'$, $B'D'$ and $C'D'$ each dominate one another. We can remove any two of them.

Iteration #2.

(i) Remove Secondary Essential Prime Implicants

	$A'D'(**)$	$A'C$	$B'C$	$A'B$	BC'	AB'	AC'
(∘)0	X						
3		X	X				
5				X	X		
7		X		X			
9						X	X
11			X			X	
13					X		X

** indicates a secondary essential prime implicant

∘ indicates a distinguished row

Product $A'D'$ is a secondary essential prime implicant; it is removed from the table.

(ii) Row Dominance

No further row dominance is possible.

(iii) Column Dominance

No further column dominance is possible.

Step 4: Solve Prime Implicant Table.

	$A'C$	$B'C$	$A'B$	BC'	AB'	AC'
3	X	X				
5			X	X		
7	X		X			
9					X	X
11		X			X	
13				X		X

There are no additional secondary essential prime implicants, and no further row- or column-dominance is possible. The remaining covering problem is called a *cyclic covering problem.* A solution can be obtained using one of two methods: (i) *Petrick's method* or (ii) the *branching method.* We use Petrick's method below; see Devadas et al. and Hachtel/Somenzi books for a discussion of the branching method.

Petrick's Method

In Petrick's method, a *Boolean expression* P is formed which describes all possible solutions of the table. The prime implicants in the table are numbered in order, from 1 to 6: $p_1 = A'C$, $p_2 = B'C$, $p_3 = A'B$, $p_4 = BC'$, $p_5 = AB'$, $p_6 = AC'$. For each prime implicant p_i, a Boolean variable P_i is used which is true whenever prime implicant p_i is included in the solution. Note the difference!: p_i is an *implicant*, while P_i is a corresponding *Boolean proposition* (i.e., true/false statement) which has a true (1) or false (0) value. $P_i = 1$ means "I select prime implicant p_i for inclusion in the cover", while $P_i = 0$ means "I do not select prime implicant p_i for inclusion in the cover.

Using these P_i variables, a larger Boolean expression P can be formed, which captures the precise conditions for every row in the table to be covered. Each clause in P is a *disjunction* (OR) of several possible column selections to cover a particular row. The *conjunction* (AND) of all of these clauses is the Boolean expression P, which describes precisely the conditions to be satisfied for *all rows* are covered.

For the above prime implicant table, the covering requirements can be captured by the Boolean equation:

$$P = (P_1 + P_2)(P_3 + P_4)(P_1 + P_3)(P_5 + P_6)(P_2 + P_5)(P_4 + P_6)$$

If Boolean variable $P = 1$, each of the disjunctive clauses is satisfied (1), and all rows are covered. In this case, the set of $P_i's$ which are 1 indicate a *valid cover* using the corresponding selection of primes

$p_i's$ (columns). If $P = 0$, then at least one disjunctive clause is not satisfied (0), meaning that at least one row is not covered. In this case, the set of $P_i's$ which are 1 correspond to a set of selected primes p_i which *do not* form a valid cover. Note that the above equation is simply a rewriting of the prime implicant table as a Boolean formula: the clauses correspond to the rows.

In the right expression, the sum $(P_1 + P_2)$ describes the covering requirement for row 3: product p_1 or p_2 *must* be included in the solution, in order to cover row 3. Similarly, the sum $(P_3 + P_4)$ describes the covering requirement for row 5: product p_3 or p_4 *must* be included to cover row 5. Each sum corresponds to a different row of the table. These sums are ANDed together, since *all* such requirements must be satisfied.

Since P is a Boolean expression, it can be multiplied out into sum-of-products form:

$$P = P_1P_4P_5 + P_1P_3P_5P_6 + P_2P_3P_4P_5 + P_2P_3P_5P_6$$
$$+P_1P_2P_4P_6 + P_1P_2P_3P_6 + P_2P_3P_4P_6 + P_2P_3P_6$$

Each product describes a solution for the table. Only two products have 3 Boolean variables; the remainder have 4 variables. These two products, $P_1P_4P_5$ and $P_2P_3P_6$, describe two minimal solutions. The first product describes a solution which includes prime implicants p_1, p_4 and p_5; that is, $A'C$, BC' and AB'. The second product describes a solution using prime implicants p_2, p_3 and p_6; that is, $B'C$, $A'B$ and AC'.

Both solutions have a minimal number of prime implicants, so either can be used. With either choice, we must include the secondary essential prime implicant, $A'D'$, identified earlier. Therefore, the two minimum-cost solutions are:

$$F = A'D' + A'C + BC' + AB'$$
$$F = A'D' + B'C + A'B + AC'$$

Example #3: Don't-Cares (Roth example, pp. 163-4).

$$F(A, B, C, D) = \Sigma m(2, 3, 7, 9, 11, 13) + \Sigma d(1, 10, 15)$$

Step 1: Generate Prime Implicants.

The don't-cares are *included* when generating prime implicants.

Note: As indicated earlier, you should learn this basic method for generating prime implicants (Step #1), but I will not ask you to reproduce it. See Roth book on reserve for more details. (Instead, you will soon learn the more advanced fast recursive algorithm for prime generation.)

List Minterms

Column I	
1	0001
2	0010
3	0011
9	1001
10	1010
7	0111
11	1011
13	1101
15	1111

Combine Pairs of Minterms from Column I

A check ($\surd$) is written next to every minterm which can combined with another minterm.

Column I			*Column II*	
1	0001	$\surd$	(1,3)	00-1
2	0010	$\surd$	(1,9)	-001
3	0011	$\surd$	(2,3)	001-
9	1001	$\surd$	(2,10)	-010
10	1010	$\surd$	(3,7)	0-11
7	0111	$\surd$	(3,11)	-011
11	1011	$\surd$	(9,11)	10-1
13	1101	$\surd$	(9,13)	1-01
15	1111	$\surd$	(10,11)	101-
			(7,15)	-111
			(11,15)	1-11
			(13,15)	11-1

Combine Pairs of Products from Column II

A check (√) is written next to every product which can combined with another product.

Column I			*Column II*			*Column III*	
1	0001	√	(1,3)	00-1	√	(1,3,9,11)	-0-1
2	0010	√	(1,9)	-001	√	(2,3,10,11)	-01-
3	0011	√	(2,3)	001-	√	(3,7,11,15)	–11
9	1001	√	(2,10)	-010	√	(9,11,13,15)	1–1
10	1010	√	(3,7)	0-11	√		
7	0111	√	(3,11)	-011	√		
11	1011	√	(9,11)	10-1	√		
13	1101	√	(9,13)	1-01	√		
15	1111	√	(10,11)	101-	√		
			(7,15)	-111	√		
			(11,15)	1-11	√		
			(13,15)	11-1	√		

The unchecked products cannot be combined with other products. These are the prime implicants: (1,3,9,11), (2,3,10,11), (3,7,11,15) and (9,11,13,15); or, using the usual product notation: $B'D$, $B'C$, CD and AD.

Step 2: Construct Prime Implicant Table.

The don't-cares are *omitted* when constructing the prime implicant table, since they do not need to be covered.

	$B'D$ (1,3,9,11)	$B'C$ (2,3,10,11)	CD (3,7,11,15)	AD (9,11,13,15)
2		X		
3	X	X	X	
7			X	
9	X			X
11	X	X	X	X
13				X

Step 3: Reduce Prime Implicant Table.

(i) Remove Essential Prime Implicants

	$B'D$ (1,3,9,11)	$B'C(*)$ (2,3,10,11)	$CD(*)$ (3,7,11,15)	$AD(*)$ (9,11,13,15)
2		X		
3	X	X	X	
(○)7			X	
9	X			X
11	X	X	X	X
(○)13				X

* indicates an essential prime implicant

○ indicates a distinguished row

Step 4: Solve Prime Implicant Table.

The essential prime implicants cover all the rows, so no further steps are required. Therefore, the minimum-cost solution consists of the essential prime implicants $B'C$, CD and AD:

$$F = B'C + CD + AD$$