

PRIME AND NON-PRIME IMPLICANTS IN THE MINIMIZATION OF MULTIPLE-VALUED LOGIC FUNCTIONS*

Parthasarathy P. Tirumalai
Building 52L, Mailstop 57
Hewlett-Packard Company
5301 Stevens Creek Blvd.
Santa Clara, CA 95052

Jon T. Butler
Department of Electrical and
Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5004

ABSTRACT

We investigate minimal sum-of-products expressions for multiple-valued logic functions for realization by programmable logic arrays. Our focus is on expressions where product terms consist of the MIN of interval literals on input variables and are combined using one of two operations - SUM or MAX. In binary logic, the question of whether or not prime implicants are sufficient to optimally realize all functions has been answered in the affirmative. We consider the same question for higher radix functions. When the combining operation is MAX, prime implicants are sufficient. However, we show that this is not the case with SUM. There is also the question of whether all functions can be optimally realized by successively selecting implicants that are prime with respect to the intermediate functions. We show that this is not true either. In fact, the number of implicants in a solution using prime implicants successively can be significantly larger than the number of implicants in a minimal solution.

I. INTRODUCTION

Advances in integrated circuit technology have made it possible to design circuits that operate using more than two logic levels. This has inspired research on logic design techniques for realizing multiple-valued functions. A recent development has been the introduction of programmable logic arrays (PLA's) for implementing multiple-valued logic (MVL) functions [1, 2, 5].

Synthesis with a multiple-valued PLA involves expressing a target function in a sum-of-products form. Each product term is realized by one column in the

PLA; these terms are combined together using a combining operation to produce the target function. Since an expression with fewer product terms results in a PLA with fewer columns, VLSI chip area is reduced.

It is well known that in binary logic minimal sum-of-products expressions can always be obtained by the use of prime implicants alone [4]. Non-prime implicants need never be considered to minimize the number of product terms required. In this paper, we consider this question for higher radix functions. Our interest is in sum-of-products expressions where product terms consist of the MIN of interval literals on input variables and the combining operation is SUM or MAX. These are the types of expressions realized by the PLA's in [2, 5].

II. BACKGROUND AND NOTATION

Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of n variables, where x_i takes on values from $R = \{0, 1, \dots, r-1\}$. A function $f(X)$ is a mapping $f: R^n \rightarrow R \cup \{r\}$, where r is the *don't care* value. Specifically, $f(X)$ is said to be an n -variable r -valued function. Fig. 1 shows a map representation of a 2-variable 4-valued function. A function value $f(x)$ corresponding to a specific assignment of values x to variables in X is called a *minterm*. For example, in Fig. 1 there are three minterms with value 1 and three with value 2.

Functions realized by the PLA's described in [2, 5] are composed of four functions,

$$1. \text{ literal: } a x_1^b = \begin{cases} 0 & \text{if } x_1 < a \text{ or } x_1 > b \\ r-1 & \text{if } a \leq x_1 \leq b \end{cases}$$

* Research supported in part by NSF Grant MIP-8706553, NATO Grant 423/84, an NPS Foundation Grant, and a Hewlett-Packard Fellowship.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE MAY 1989		2. REPORT TYPE		3. DATES COVERED	
4. TITLE AND SUBTITLE Prime and Non-Prime Implicants in the Minimization of Multiple-Valued Logic Functions				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School, Department of Electrical and Computer Engineering, Monterey, CA, 93943				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We investigate minimal sum-of-products expressions for multiple-valued logic functions for realization by programmable logic arrays. Our focus is on expressions where product terms consist of the MIN of interval literals on input variables and are combined using one of two operations - SUM or MAX. In binary logic, the question of whether or not prime implicants are sufficient to optimally realize all functions has been answered in the affirmative. We consider the same question for higher radix functions. When the combining operation is MAX, prime implicants are sufficient. However, we show that this is not the case with SUM. There is also the question of whether all functions can be optimally realized by successively selecting implicants that are prime with respect to the intermediate functions. We show that this is not true either. In fact the number of implicants in a solution using prime implicants successively can be significantly larger than the number of implicants in a minimal solution.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

x_1				
x_2	0	1	2	3
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
3	2	2	2	0

Figure 1. An example of a 2-variable 4-valued function.

2. MIN: $x_1 \cdot x_2$ ($= \text{minimum}(x_1, x_2)$), and
3. MAX: $x_1 \circ x_2$ ($= \text{maximum}(x_1, x_2)$), or

$$\text{SUM: } x_1 + x_2 = \begin{cases} r-1 & \text{if } \text{plus}(x_1, x_2) > r-1 \\ \text{plus}(x_1, x_2) & \text{otherwise} \end{cases}$$

where *plus* represents ordinary addition and x_i is viewed as an integer. The SUM operation (+) is thus addition truncated to the highest logic value.

When $r = 2$, the MIN and literal correspond respectively to the binary AND and x^* , where $x^* \in \{x, \bar{x}\}$. There is no difference between SUM and MAX; they represent the OR operation. In the realization of functions by a multiple-valued PLA, constants and literals occur as operands of the MIN functions. A *product term* $p \cdot \Phi(x_1, x_2, \dots, x_n)$ is the MIN of a constant and a set of literals where each variable x_i appears at most once, and p is a constant in the set $\{1, 2, \dots, r\}$. For example, $f(x_1, x_2) = 2 \cdot x_1^4 \cdot x_2^1$ is a product term that is 2 when x_1 is 2, 3, or 4 and x_2 is 1. An *implicant* of a multiple-valued function $f(X)$ is a product term $I(X) = p \cdot \Phi(x_1, x_2, \dots, x_n)$ such that $f(x) \geq I(x)$ for every assignment of values x to variables in X and $p \in \{1, 2, \dots, r-1\}$. Also, $|I(X)|$ denotes the nonzero constant p associated with the implicant $I(X)$. An implicant $I(X)$ of a function $f(X)$ is a *prime implicant* if there is no other implicant $I'(X)$ of $f(X)$ such that $I'(x) \geq I(x)$ for every assignment of values x to variables in X . For example, $1 \cdot x_1^0 \cdot x_2^2$ is an implicant of the function shown in Fig. 1, but it is not a prime implicant. However, $1 \cdot x_1^0 \cdot x_2^3$ is a prime implicant. Any function can be expressed as the SUM or MAX of implicants [5]. For example, the function in Fig. 1 can be expressed as the SUM of 2 implicants,

$$f(x_1, x_2) = 1 \cdot x_1^0 \cdot x_2^2 + 2 \cdot x_1^1 \cdot x_2^3$$

We use the term *sum-of-products* to describe functions realized by multiple-valued PLA's, where sum refers to SUM or MAX. A sum-of-products expression for function $f(X)$ is *minimal* if there is no other expression for $f(X)$ with fewer product terms. For example, (1) is a minimal sum-of-products expression. Given $f(X)$, implicant $I(X)$ covers a minterm at x if $f(x) = I(x)$. Therefore, $g(X) = f(X) - I(X)$ has the property $g(x) = 0$.

III. MINIMAL SUM-OF-PRODUCTS EXPRESSIONS USING MAX

A version of this problem has been considered by Miller and Muzio [3]. In their work, varying costs are assigned to literals of the input variables, and, therefore, also to the product terms. They use the MAX as the combining operator and show that in order to minimize total cost, prime implicants are not sufficient. This observation is made on the basis of an example where the use of prime implicants alone results in a non-minimal solution based on their cost factors. However, if all literals are assigned the same cost, prime implicants are sufficient.

In a PLA, however, each product term is realized with one column. The interest is then to simply minimize the number of product terms. The following theorem shows that when the combining operator is MAX, only prime implicants need be considered. This result has been applied in previous papers but as far as we know has not been proven formally.

Theorem 1: Let $f(X)$ be an n -variable r -valued function, and let S_p be the set of prime implicants of $f(X)$. There exists a minimal sum-of-products expression of $f(X)$ where sum is MAX such that every product term is a member of S_p .

Proof: The proof is constructive starting with any minimal expression for $f(X)$. Let,

$$f(X) = I_1(X) \circ I_2(X) \circ \dots \circ I_i(X) \circ \dots \circ I_k(X) \quad (1)$$

be a minimal sum-of-products expression for the function $f(X)$. This implies that for every assignment of values x to variables in X ,

$$f(X) = \text{maximum} \left\{ I_1(X), I_2(X), \dots, \right.$$

$$I_i(X), \dots, I_k(X) \}. \quad (2)$$

Let some implicant $I_i(X)$ in the expression in (1) be non-prime, that is, $I_i(X) \notin S_p$. Since $I_i(X)$ is not a prime implicant, it follows that there exists a prime implicant $J_i(X)$ such that,

$$J_i(X) \geq I_i(X) \quad (3)$$

for every assignment of values x to variables in X . Also, from the definition of an implicant, for every possible assignment x ,

$$f(X) \geq J_i(X). \quad (4)$$

Equations (2), (3) and (4) together give,

$$f(X) = \text{maximum} \left\{ I_1(X), I_2(X), \dots, J_i(X), \dots, I_k(X) \right\}. \quad (5)$$

Therefore, $J_i(X)$ can replace $I_i(X)$ in (1) giving,

$$f(X) = I_1(X) \circ I_2(X) \circ \dots \circ J_i(X) \dots \circ I_k(X). \quad (6)$$

In a similar fashion every non-prime implicant in equation (1) can be replaced with a corresponding prime implicant. Thus a minimal expression for $f(X)$ can be obtained that uses only prime implicants.

Q.E.D.

As an example, consider the function shown in Fig. 1. A minimal expression for this function is,

$$f(x_1, x_2) = 1. {}^0x_1^0 \cdot {}^0x_2^2 \circ 2. {}^0x_1^2 \cdot {}^3x_2^3 \quad (7)$$

in which $1. {}^0x_1^0 \cdot {}^0x_2^2$ is non-prime. However, it may be replaced by $1. {}^0x_1^0 \cdot {}^0x_2^3$ yielding a minimal expression consisting of prime implicants alone,

$$f(x_1, x_2) = 1. {}^0x_1^0 \cdot {}^0x_2^3 \circ 2. {}^0x_1^2 \cdot {}^3x_2^3. \quad (8)$$

IV. MINIMAL SUM-OF-PRODUCTS EXPRESSIONS USING SUM

In this section, we ask the question - Does there exist a minimal sum-of-products expression for every

n -variable r -valued function $f(X)$ using SUM as the combining operator where all the product terms are prime implicants of $f(X)$? The answer, unfortunately, is NO. In fact, there exist functions where no expression (minimal or non-minimal) contains prime implicants exclusively. Consider, for example, the 2-variable 4-valued function $f(x_1, x_2)$ shown in Fig. 1. This function has just two prime implicants,

- (1) $1. {}^0x_1^0 \cdot {}^0x_2^3$, and
- (2) $2. {}^0x_1^2 \cdot {}^3x_2^3$.

No combination of these two prime implicants yields the function $f(x_1, x_2)$. In particular, the SUM of these two prime implicants has the value 3 when $(x_1, x_2) = (0, 3)$ but $f(0, 3) = 2$. A non-prime implicant must be used in every expression for $f(x_1, x_2)$ - minimal or not.

This negative result implies that a search for a minimal sum-of-products expression must include implicants that are non-prime with respect to the target function. In a typical function, there are many more implicants than prime implicants. For example, the trivial 4-valued function $f(x_1, x_2) = 2$ has one prime implicant and 300 implicants.

		x_1			
		x_2	0	1	2
	0	1	0	0	0
	1	1	0	0	0
	2	1	0	0	0
	3	0	0	0	0

Figure 2. The function of Fig. 1 with one prime implicant subtracted.

However, there is still the possibility that a minimal sum-of-products expression exists consisting of implicants that are prime with respect to the intermediate functions. That is, at each step in the minimization of a given function, the chosen implicant is prime with respect to the intermediate function. This property holds for the function of Fig. 1. Specifically, subtract the prime implicant $I_1(x_1, x_2) = 2. {}^0x_1^2 \cdot {}^3x_2^3$.

The remaining function, $g(x_1, x_2) = f(x_1, x_2) - I_1(x_1, x_2)$ can be covered with the implicant $1 \cdot 0x_1^0 \cdot 0x_2^2$ which is prime with respect to $g(x_1, x_2)$ (Fig. 2). A formal description of an algorithm that chooses implicants that are prime with respect to the intermediate functions is given below.

Algorithm 1:

1. $S \leftarrow \emptyset$.
2. If $f(X)$ has no minterms in the range $[1, 2, \dots, r-1]$, STOP and return S .
3. $I(X) \leftarrow$ some prime implicant of $f(X)$ such that $I(X)$ covers at least one minterm in $f(X)$ that is in the range $[1, 2, \dots, r-1]$.
4. $S \leftarrow S \cup I(X)$.
5. $f(X) \leftarrow f(X) - I(X)$.
6. Go to step 2.

It should be clear that at each iteration at least one additional minterm in the function $f(X)$ which is nonzero and not a don't care is covered. Further, the subtraction process does not generate *new* minterms where the original function was a zero or a don't care; when a minterm is covered it either becomes a zero or a don't care. Since there are only a finite number of minterms to start with, the procedure must terminate after a finite number of steps. It is important to note that all the implicants in the solution set S are not necessarily prime with respect to the input function, although at least one must be so for any non-trivial input function. Note also that each iteration through the loop requires the examination only of prime implicants of the current $f(X)$.

Does this now mean that we can *always* select prime implicants in some order to obtain a minimal expression for a given function? Consider the 2-variable 5-valued function shown in Fig. 3 [6]. This function has a cycle of 2's in it. There is only one minimal expression for this function,

$$f(x_1, x_2) = 2 \cdot 0x_1^2 \cdot 1x_2^1 \odot 2 \cdot 1x_1^1 \cdot 2x_2^4 \odot 2 \cdot 2x_1^4 \cdot 3x_2^3 \odot 2 \cdot 3x_1^3 \cdot 0x_2^2. \quad (9)$$

None of the four implicants in the above expression are prime. It is easily seen that if any prime implicant is chosen, it *breaks up* a string of 2's in a perpendicular direction, thereby producing a non-minimal solution. Algorithm 1 never produces a minimal solution in this case.

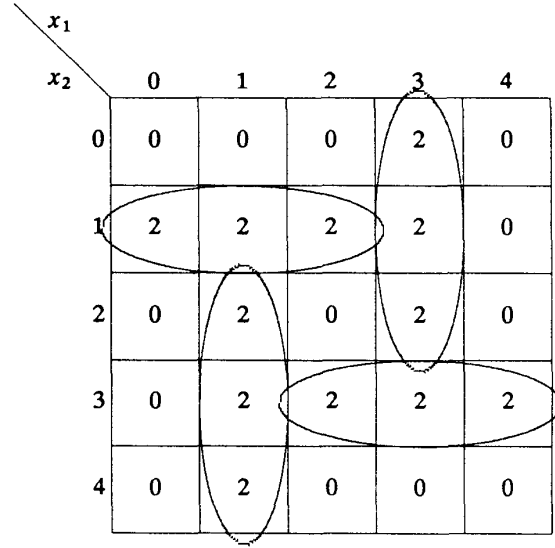


Figure 3. A 2-variable 5-valued function that requires the use of non-prime implicants to produce a minimal expression.

This negative result implies that a search for a minimal sum-of-products expression must include implicants that are non-prime with respect to the target function *and* intermediate functions. As this example applies to the case of 2-variable 5-valued functions, it is natural to ask if a similar observation is true of other radices and variable cardinalities. Since the function of Fig. 3 can be a subfunction of an n -variable 5-valued function where $n \geq 3$, it is true of these functions, as well. Further, similar examples can be constructed for any radix $r > 5$ and for $n \geq 2$.

However, the example does not extend to $r < 5$. Indeed, for $r = 2$, prime implicants are sufficient to produce minimal sum-of-products expression for all functions. Thus, there is the question of whether the observation holds for $3 \leq r \leq 4$. For the case of $n = 2$ and $r = 4$, the observation is shown to be true by the function of Fig. 4.

A minimal expression for this function requires five implicants.

$$f(x_1, x_2) = 1 \cdot 0x_1^2 \cdot 0x_2^0 + 1 \cdot 3x_1^3 \cdot 1x_2^3 + 1 \cdot 2x_1^3 \cdot 0x_2^1 + 1 \cdot 1x_1^1 \cdot 1x_2^2 + 1 \cdot 1x_1^1 \cdot 2x_2^2. \quad (10)$$

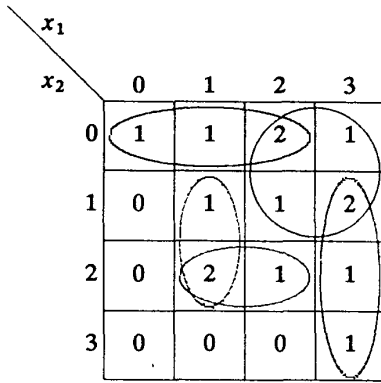


Figure 4. A 2-variable 4-valued function which requires the use of non-prime implicants to produce a minimal expression.

None of the implicants in (10) is prime. It is not easy to see that there is no minimal expression that uses prime implicants only. This function has six prime implicants which are listed below.

- (1) $1. {}^0x_1^3 \cdot {}^0x_2^0$
- (2) $1. {}^3x_1^3 \cdot {}^0x_2^3$
- (3) $1. {}^1x_1^3 \cdot {}^0x_2^2$
- (4) $2. {}^2x_1^2 \cdot {}^0x_2^0$
- (5) $2. {}^1x_1^1 \cdot {}^2x_2^2$
- (6) $2. {}^3x_1^3 \cdot {}^1x_2^1$

It can be verified that if any of these prime implicants is subtracted from the function, the new function obtained still needs five implicants in a minimal cover. For example, if implicant (3) is subtracted, the function becomes as shown in Fig. 5. There are five 1's left and each requires one implicant to cover it. This 2-variable 4-valued function is, therefore, an example where prime implicants cannot be successively picked to produce a minimal expression.

The case of ternary logic functions remains. Not allowing don't cares, there are $3^{(3^2)} = 19683$ 3-valued functions of 2-variables. A computer program was developed to analyze each of these functions, and record those for which a minimal sum-of-products expression cannot be generated by successively choosing prime implicants. Twenty one such functions were found. These can be placed in six classes, A through F

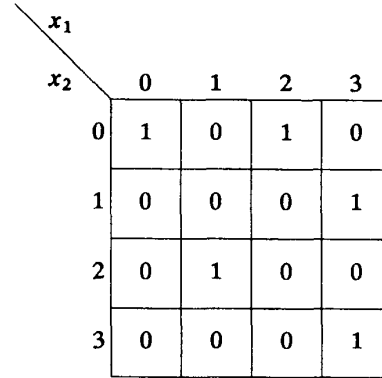


Figure 5. The function of Fig. 4 with one prime implicant subtracted.

where each member of a class is obtained from another by a rotation of 90, 180 or 270 degrees. Classes labelled A through E consist of four functions each while class F consists of a single symmetric function. Fig. 6 shows a representative from each of the six classes.

The function in class F is interesting. It has a minimal cover consisting of five implicants,

$$f(x_1, x_2) = 1. {}^0x_1^2 \cdot {}^0x_2^0 + 1. {}^2x_1^2 \cdot {}^0x_2^2 + 1. {}^0x_1^2 \cdot {}^2x_2^2 + 1. {}^0x_1^0 \cdot {}^0x_2^2 + 2. {}^1x_1^1 \cdot {}^1x_2^1. \quad (11)$$

No matter what implicant is subtracted from this function, a 2 can only become a 1 or a 3 (don't care), or it can remain a 2. Therefore, the only way to cover the 1's in the function using prime implicants is with the implicant $1. {}^0x_1^2 \cdot {}^0x_2^2$. However, if this implicant is subtracted, five 1's remain. These require five more implicants as no two 1's are adjacent. Thus, if only prime implicants are used, this function has a cover whose size is at least six.

This pattern extends to functions with a larger number of variables as well. Consider an n -variable 3-valued function $f(X)$ with $n \geq 2$ which is as follows,

$$f(X) = \begin{cases} 2 & \text{when } x_i, x_j \in \{0, 2\}, 0 < i, j \leq n, i \neq j \\ 2 & \text{when } x_1 = x_2 = \dots = x_n = 1. \\ 1 & \text{otherwise} \end{cases} \quad (12)$$

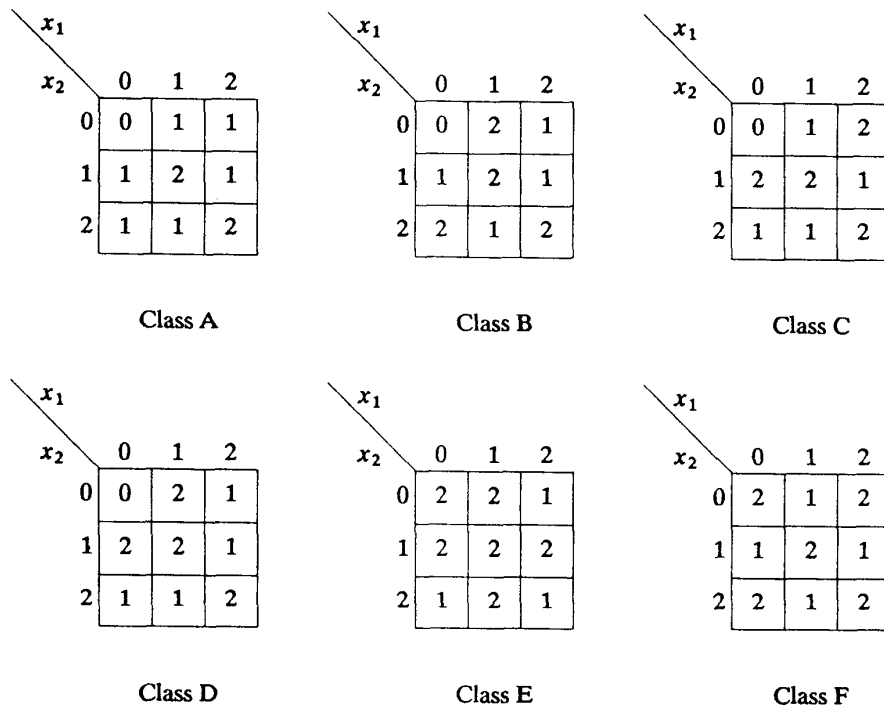


Figure 6. Representative 2-variable 3-valued functions which require the use of non-prime implicants to produce minimal expressions.

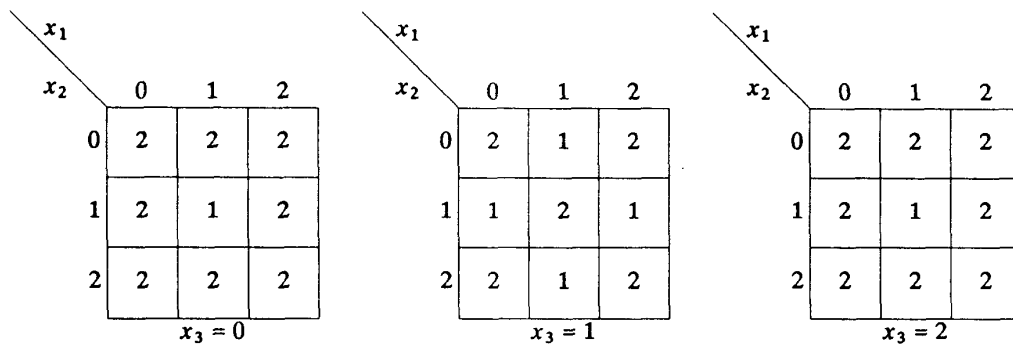


Figure 7. A 3-variable 3-valued function which requires the use of non-prime implicants to produce a minimal expression.

Fig. 7 shows the function of (12) when $n = 3$. It is instructive to imagine this function as a set of 1's and 2's distributed on a n -dimensional hyper-cube. For such a cube, a *face* is obtained by fixing some variable at 0 or 2. Similarly, an *edge* - which is the *meeting* of two faces, is obtained by fixing a pair of variables at 0

or 2, independently. The function of (12) has the value 2 along all edges and at all off-face locations, and the value 1 everywhere else. It can be expressed in a sum-of-products expression using $(2n + 1)$ implicants as follows,

TABLE I. Implicants needed to produce minimal sum-of-products expressions using interval literals of input variables to form product terms.

Function type (n-variable, r-valued)	Combining Operator	Implicants sufficient to produce a minimal sum-of-products expression
Binary case ($r = 2$) n-variable, 2-valued	MAX/SUM	Prime imp. only [4] (MAX and SUM are both equivalent to OR)
Multiple-valued case ($r \geq 3$) n-variable, r-valued 1-variable, r-valued n-variable, r-valued, ($n \geq 2$)	MAX SUM SUM	Prime imp. only (Theorem 1) Prime imp. only (Theorem 3) All implicants (Theorem 2)

$$f(X) = 1. {}^0x_1^0 + 1. {}^2x_1^2 + 1. {}^0x_2^0 + 1. {}^2x_2^2 + \dots + 1. {}^0x_n^0 + 1. {}^2x_n^2 + 2. {}^1x_1^1. {}^1x_2^1. \dots. {}^1x_n^1. \quad (13)$$

As before, the only prime implicant that can be used to cover the 1's in this function corresponds to the constant 1 function. If this is subtracted out, however, 1's are left along every *edge* and in the *center*. Note that no two edges can be covered completely with one implicant. Also, the 1 left in the center is completely isolated. Since an n -dimensional hyper-cube has $2n(n-1)$ edges, an expression using prime implicants requires $2n(n-1) + 2$ product terms. Thus, the difference between using all implicants and using successive prime implicants can be large. For instance, when $n = 6$, the number of product terms required are 13 and 62 respectively! To generalize this function for any radix $r > 2$, the values 1 and 2 in equation (12) may be replaced with $r-2$ and $r-1$ respectively. This leads us to the following theorem.

Theorem 2: In each set of n -variable r -valued functions with $n \geq 2$ and $r \geq 3$, there exists a function $f(X)$ with the property that all minimal sum-of-products expressions for $f(X)$ where sum is SUM are composed entirely of non-prime implicants.

Finally, in the case of unary r -valued functions a minimal expression can be found using only prime implicants. The procedure is as follows. Let $f(x)$ be an r -valued function of a single variable x .

1. $S \leftarrow \emptyset$.
2. If $f(x)$ has no minterms in the range $[1, 2, \dots, r-1]$, STOP and return S .
3. Find a nonzero minterm with the smallest logic value. Let $I(x)$ be the prime implicant that covers this minterm (for unary functions there can be at most one prime implicant associated with each minterm).
4. $S \leftarrow S \cup I(x)$.
5. $f(x) \leftarrow f(x) - I(x)$.
6. Go to step 2.

On exit, S contains the prime implicants required to cover the input function $f(x)$. This proves the following.

Theorem 3: Let $f(x)$ be a 1-variable r -valued function and let S_p be the set of prime implicants of $f(x)$. There exists a minimal sum-of-products expression where sum is SUM such that every product term is a member of S_p .

V. CONCLUSION

Finding minimal sum-of-products expressions for multiple-valued functions is important to implement these functions using compact programmable logic arrays. This paper has focussed on expressions where

interval literals of input variables are used to form product terms and the combining operation is either SUM or MAX. The question of whether it is sufficient to consider prime implicants alone, or whether all implicants need to be considered to obtain minimal expressions has been investigated. When the MAX operation is used, a minimal expression can be found that uses only prime implicants of the input function. With the SUM there may not exist *any* expression that uses only prime implicants of the input function. It is possible, however, to select prime implicants *successively* to cover the input function. It has been seen that this does not guarantee minimality for all but unary functions. There exist functions for which the difference between using successive prime implicants only and using all implicants is significantly large. Table I shows a summary of the results.

References

- [1] G. W. Dueck and D. M. Miller, "A 4-valued PLA using the Modsum," *Proceedings of the 16th International Symposium on Multiple-Valued Logic*, May 1986, pp. 232-240.
- [2] H. G. Kerkhoff and J. T. Butler, "Design of a high-radix programmable logic array using profiled peristaltic charge-coupled devices," *Proceedings of the 16th International Symposium on Multiple-Valued Logic*, May 1986, pp. 100-103.
- [3] D. M. Miller and J. C. Muzio, "On the minimization of many-valued functions," *Proceedings of the 9th International Symposium on Multiple-Valued Logic*, May 1979, pp. 294-299.
- [4] W. V. Quine, "The problem of simplifying truth functions," *Am. Math. Monthly*, Oct. 1952, pp. 521-531.
- [5] P. Tirumalai and J. T. Butler, "On the realization of multiple-valued logic functions using CCD PLA's," *Proceedings of the 14th International Symposium on Multiple-Valued Logic*, May 1984, pp. 33-42.
- [6] P. Tirumalai and J. T. Butler, "Analysis of minimization algorithms for multiple-valued programmable logic arrays," *Proceedings of the 18th International Symposium on Multiple-Valued Logic*, May 1988, pp. 226-236.