مینیمم کردن طرح مدارات منطقی که بیش از یک خروجی دارند
با استفاده از یک الگوریتم کامپیوتری

امین رفیع زاده
بخش مهندسی برق دانشگاه شهید باهنر کرمان
Rafizade@gmail.com

مریم ارجمند
بخش مهندسی برق دانشگاه شهید باهنر کرمان

چکیده: در طراحی مدارات منطقی خیلی از موارد مایلیم که چند مدار با چند خروجی متفاوت را به صورت مینیمم دو سطحی (حداقل تأخیر زمانی با مینیمم گیتهای منطقی) طرح کنیم بعنوان مثال طراحی مدارات منطقی با PAL ها ، اگر متغیرهای ورودی و تعداد توابع (مدارات) زیاد باشد مایلیم این کار را با کامپیوتر انجام دهیم، راه پیشنهادی من با استفاده از الگوریتم **Quine McClusky** یک الگوریتم کامپیوتری برای طرح مینیمم دو سطحی مدارات منطقی پیشنهاد می کند. تمام عکسهای استفاده شده در این مقاله از برنامه کامپیوتری که بر اساس الگوریتم پیشنهادی نوشته ام، تهیه شده است.

کلمات کلیدی: *طراحي مدارات منطقي، مينيمم دو سطحي، طراحي مدارات منطقي با كامپيوتر، Quine McClusky*

مقدمه: Quine McClusky یک راه برای مینیمم کردن توابع بصورت دو سطحی است که بیشتر برای توابع با متغیرهای ورودی بیش از 6 تا و نیز کامپیوتری کردن حل این توابع مورد استفاده قرار می گیرد.
این روش بر اساس جمع آوری ترمهایی است که با هم در یک بیت اختلاف دارند.

شرح مقاله: مراحل کار الگوریتم Quine McClusky به صورت زیر می باشد:
1- ابتدا تابع را بصورت SOP یا POS طرح شده و شماره مینترمها یا ماکسترمها بصورت باینری در آورده به عنوان مثال اگر شماره مینترمهای یک تابع بصورت SOP مثل زیر باشند داریم:

F1
0
1
2
5
7
10
11
14
15

0	1	2	5	7	10	11	14	15
0000	0001	0010	0101	0111	1010	1011	1110	1111

2- سپس مینترمها (یا ماکسترمها)ی تابع را بر اساس تعداد یکهای مو جود در آنها دسته بندی می کنیم

0000		
0001	0010	
0101	1010	
0111	1011	1110
1111		

3- مینترمهای هر دسته را با دسته مجاور بعدی مقایسه می کنیم چنانچه این مینترمها فقط در یک متغیر اختلاف داشته باشد آن مینترم را ترکیب کرده و به جای متغیر مورد اختلاف علامت (-) را قرار می دهیم و بقیه متغیر های ثابت نوشته می شود و کنار ترکیبهایی که دوباره ترکیب شدند علامتی مثل (Y) قرار می دهیم.

0000 Y		
0001 Y	0010 Y	
0101 Y	1010 Y	
0111 Y	1011 Y	1110 Y
1111 Y		

(0,1)	000-
(0,2)	00-0
(1,5)	0-01
(2,10)	-010
(5,7)	01-1
(10,11)	101-
(10,14)	1-10
(7,15)	-111
(11,15)	1-11
(14,15)	111-

4- ترکیبهای بدست آمده را مانند مرحله قبل با یکد یگر مقایسه می کنیم جنانجه دو ترکیب با محل علامت (-)یکسان فقط در یک متغیر اختلاف داشته باشد تشکیل یک ترکیب جهار تایی میدهند که اینبار نیز به جای متغیر مورد اختلاف (-) می نویسیم و کنار ترکیبهایی که دوباره ترکیب شدند علامتی مثل (Y) قرار می دهیم.

(0,1)	000-	(10,11,14,15)	1-1-
(0,2)	00-0	(10,14,11,15)	1-1-
(1,5)	0-01		
(2,10)	-010		
(5,7)	01-1		
(10,11)	101- Y		
(10,14)	1-10 Y		
(7,15)	-111		
(11,15)	1-11 Y		
(14,15)	111- Y		

5- در این مرحله باید ترکیب تکراری نیز بوجود آید که باید حذف شوند .

6- مرحله 4 و 5 را برای ترکیبات بزرگتر تکرار می کنیم.

7- به هر کدام از ترکیبهای غیر تکراری بدست آمده یک مستلزم اول (Prime Implicant (PI))

(0,1)	000- PI 1	(10,11,14,15)	1-1- PI 2
(0,2)	00-0 PI 3		
(1,5)	0-01 PI 4		
(2,10)	-010 PI 5		
(5,7)	01-1 PI 6		
(10,11)	101- Y		
(10,14)	1-10 Y		
(7,15)	-111 PI 7		
(11,15)	1-11 Y		
(14,15)	111- Y		

در این مرحله باید توجه کرد که اگر مینترمی با بقیه ترکیب نشد به آن نیز یک مستلزم اول بدهیم.

8- جدول جدیدی تشکیل میدهیم که سطر بالای آن مینترمها یا ماکسترمهای تابع . در ستون چپ آن تمامی مستلزم های اول بدست آمده درج می شود و در داخل این جدول تعیین می کنیم که هر مستلزم چه ترمهایی را می پوشاند.

9- اگر مینترمی فقط توسط یک مستلزم اول پو شانده شده باشد آن مستلزم یک مستلزم اول اساسی می گوییم و با گذاشتن یک (*) آن را مشخص می کنیم.

(Essential Prime Implicant (EPI))

		0	1	2	5	7	10	11	14	15
PI 1	000-	Y	Y							
*PI 2	1-1-						Y	Y	Y	Y
PI 3	00-0	Y		Y						
PI 4	0-01		Y		Y					
PI 5	-010			Y			Y			
PI 6	01-1				Y	Y				
PI 7	-111					Y				Y

10- EPI ها را با مینترمهایی که می پو شانند از جدول حذف می کنیم و بین مستلزمهای باقی مانده مینیمم را انتخاب می کنیم که بقیه مینترمها پوشانده شوند.

11- تابع را با جمع کردن مستلزم ها ی بدست آمده مشخص می کنیم و اگر بصورت ماکسترم باشد با ضرب کردن آنها :

F(X1, X2, X3, X4) = *PI 2 + PI 1 + PI 6 + PI 3

12- هر مستلزم را با نوشتن یک جمله حاصلضرب با شرایط زیر مشخص می کنیم :

برای فرم مینترمی: (-) برای حذف متغیر (0) برای نقیض متغیر و (1) برای خود متغیر

برای فرم ماکسترمی: (-) برای حذف متغیر (0) برای خود متغیر و (1) برای نقیض متغیر

F(X1, X2, X3, X4) = X1 X3 + X'1 X'2 X'3 + X'1 X2 X4 + X'1 X'2 X'4

همانطور که می بینیم این روش برای مینیمم کردن **یک** تابع بصورت دو سطحی می باشد. ولی اگر بخواهیم مداری که از چند تابع تشکیل شده است به صورت مینیمم دو سطحی طرح کنیم این الگوریتم به صورت جداگانه این تابع ها را مینیمم می کند و مدار مجموع دارای تعداد گیت و ورودیهای بسیار زیادی می شود.

راه پیشنهادی :

روش پیشنهادی من بر مبنای پیدا کردن ترمهای مشترک بین توابع مختلف است که با انتخاب یک گیت برای آنها میتوان تعداد گیت های استفاده شده و نیز تعداد ورودیها در یک مدار را خیلی کم کرد.

1- ابتدا از تابع های موجود یک مجموعه می سازیم:

{ F1, F2, F3 }

2- تمام زیر مجموعه های این مجموعه را می نویسیم:

{F1} {F2} {F3} {F1, F2} {F1, F3} {F2, F3} {F1, F2, F3} {Ø}

3- حال این زیرمجموعه ها را بر اساس تعداد توابع موجود در آنها مرتب می کنیم:

{F1, F2, F3}→{F2, F3}→{F1, F2}→{F1,F3}→{F1}→{F2}→{F3}→{Ø}

4- به ترتیب تمام ترمهای مشترک توابع در هر مجموعه را به الگوریتم Quine McClusky می دهیم تا تحلیل بکند. به عنوان مثال :

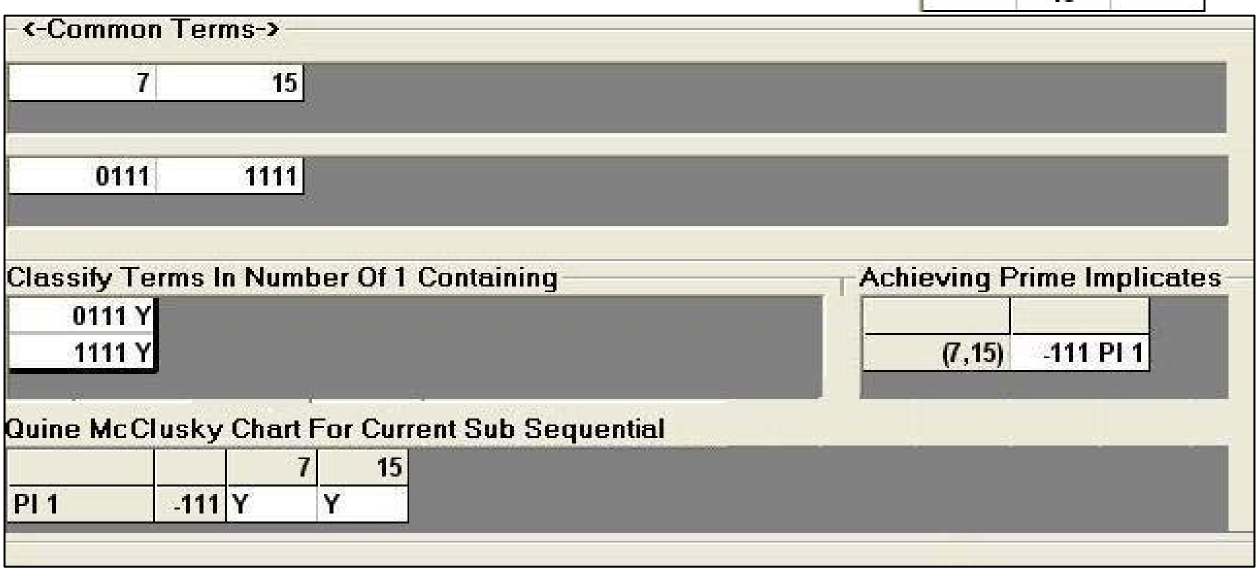
$F1(X1, X2, X3, X4) = \sum m(7, 10, 14, 15)$

$F2(X1, X2, X3, X4) = \sum m(0, 1, 4, 5, 7, 12, 13, 15)$

$F3(X1, X2, X3, X4) = \sum m(7, 10, 12, 13, 14, 15)$

F1	F2	F3
7	0	7
10	1	10
14	4	12
15	5	13
	7	14
	12	15
	13	
	15	

Common terms of { F1, F2, F3 } are {7, 15}



5- جدولی جدیدی تشکیل می دهیم که در سطر بالای آن تمام توابع با مینترمهای آنها باشند و در طرف چپ آن PI های بدست آمده از الگوریتم Quine McClusky را می نویسیم و مینترمهایی که می پوشاند را برای توابعی که در این زیر مجموعه قرار دارند علامت می زنیم:

Functions		F1				F2								F3					
Terms		7	10	14	15	0	1	4	5	7	12	13	15	7	10	12	13	14	15
PI 1	-111	Y			Y					Y			Y	Y					Y

6- برای تمام زیر مجموعه ها به ترتیب مراحل 4 و 5 را تکرار می کنیم اگر در مرحله 4 یک PI جدیدی بوجود آید و ترمی یا ترمهایی را بپو شاند که بوسیله PI های قبلی پو شانده نشده باشد یک PI به آن اختصاص می دهیم و در جدول نهایی (مرحله 5) قرار می دهیم.

Common terms of { F2, F3 } are {7, 12, 13, 15}

<-Common Terms->

7	12	13	15
0111	1100	1101	1111

Classify Terms In Number Of 1 Containing

1100 Y	
0111 Y	1101 Y
1111 Y	

Achieving Prime Implicates

(12,13)	110- PI 1
(7,15)	-111 PI 2
(13,15)	11-1 PI 3

Quine Mc Clouskey Chart For Current Sub Sequential And Total Chart

		7	12	13	15
PI 1 = PI 2	110-		Y	Y	
PI 2	-111	Y			Y
PI 3	11-1			Y	Y

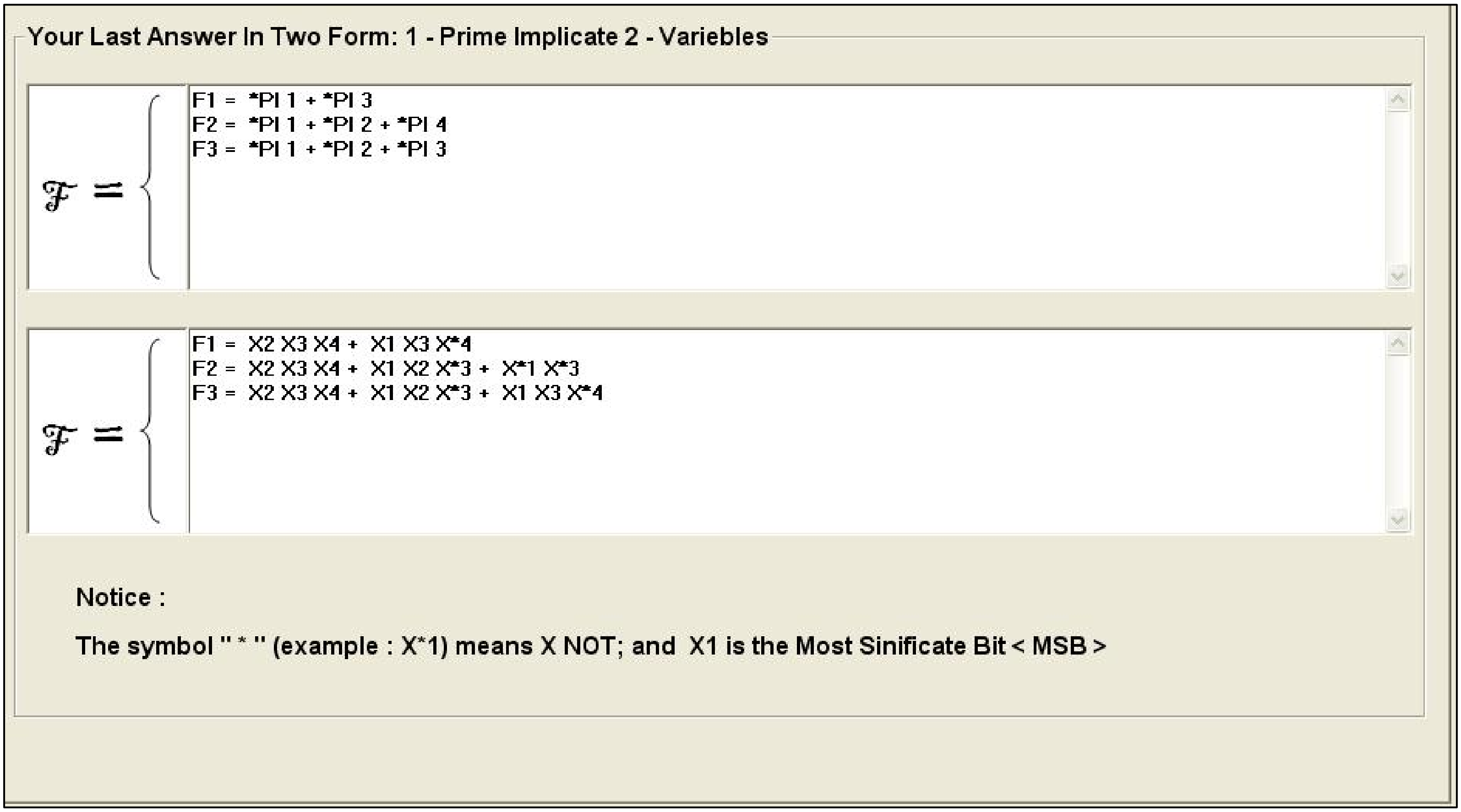
Functions		F1				F2								F3					
Terms		7	10	14	15	0	1	4	5	7	12	13	15	7	10	12	13	14	15
PI 1	-111	Y			Y					Y			Y	Y					Y
PI 2	110-										Y	Y				Y	Y		

در این مورد چون PI ترمهای 12 و 13 را برای توابع F2,F3 پو شانده که قبلا پوشانده نشده بود پس به آن یک PI اختصاص می دهیم و در جدول نهایی به عنوان PI 2 می نوبسیم ولی چون PI2 و PI3 در جدول Quine McClusky ترم جدیدی را برای توابع F2,F3 نمی پوشانند هیچگونه PI یی را در این جدول به آنها اختصاص نمی دهیم.

7- این کار را برای تمامی زیر مجموعه ها به نرتیب تکرار می کنیم تا جدولی مانند شکل زیر بدست آید:

Functions		F1				F2								F3					
Terms		7	10	14	15	0	1	4	5	7	12	13	15	7	10	12	13	14	15
PI 1	-111	Y			Y					Y			Y	Y					Y
PI 2	110-										Y	Y				Y	Y		
PI 3	1-10		Y	Y											Y			Y	
PI 4	0-0-					Y	Y	Y	Y										

8- حال در این جدول بدست آمده با استفاده از مرحله 9 در الگوریتم Quine McClusky مستلزمهای اول **(EPI)** را برای هر تابع جداگانه بدست آورده و مراحل 10 و 11 و 12 را از روی این جدول نهایی و برای هر تابع جداگانه انجام می دهیم تا به جوابی بصورت زیر برسیم:



در این مثال خاص اگر جداگانه تابع ها را مینیمم می کردیم و مداری از مجموع آنها می بستیم تعداد 28 ورودی و 11 گیت (بدون احتساب گیتهای **NOT**)می داشتیم ولی بوسیله این روش تعداد 19 ورودی و 7 گیت مورد نیاز می باشد.

نتیجه گیری:

این روش همانطور که دیدیم تعداد ورودیها و گیتهای منطقی را بسیار کاهش داده که دارای مزایای زیر می باشد:

1-ورودی کم باعث می شود توان کمتری از منبع تغذیه کشیده شود

2-گیتهای کم باعث صرفه جویی در فضای مدار و هزینه و پیچیدگی کمتر می شود

3-طرح مدار به صورت مینیمم دو سطحی باعث افزایش سرعت کار مدار می شود

4-طراحی مدارات منطقی با استفاده از PAL با این روش باعث افزایش سرعت مدار می شود

البته این روش وقتی تعداد توابع بیشتر می شود بیشتر ارزش خود را نشان می دهد به عنوان مثال برای یک مدار با 4 تابع بدون این روش 66 ورودی و 22 گیت و با این روش 50 ورودی و 16 گیت بیشتر مورد نیاز نمی باشد.

برنامه کامپیوتری نوشته شده بر اساس این الگوریتم عملی بودن و صد درصد درست کار کردن این الگوریتم را به اثبات می رساند.

منابع:

[1] MC Cluskey, E.J.” Minimization of Boolean Functions” , Bell System Tech, J.,35:5, 1417-1444 (Nov, 1956)
[2] Quine, W.V. “The Problem Of Simplifying Truth Functions: Am, Math, Monthly, 59:8, 521-531 (Oct. 1953)
[3] Digital Design, Third Edition, M. Morris Mano, CALIFORNIA STATE UNIVERSITY, LOS ANGELES. ISBN 0 – 13 – 062121 – 8
[4] Intruduction to Ditital Systems