

זיכרונות ורכיבים ניתנים לצריבה

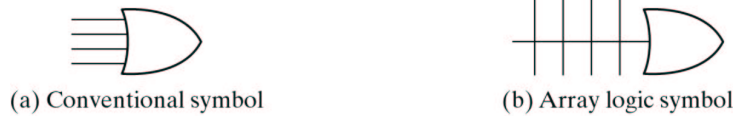
(פרק 5-7,8,9 ב Mano).

תוכן:

- 1. ROM
- 2. PLD
- 3. PLA
- 4. PAL
- 5. תרגיל לדוגמא

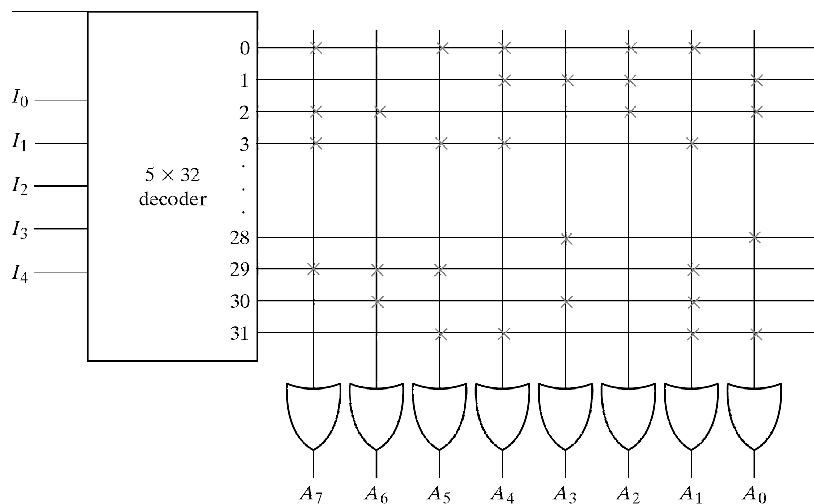
ROM.1

רכיב ה ROM הוא זכרון לקריאה בלבד (Read Only Memory). הרכיב מקבל כתובת (או m קלטים) ומוציא כפלט k ביטים של data (או k ערכי פונקציות). ניתן לפרש את פעולת ה ROM כזכרון או כאוסף פונקציות בוליאניות. במקרה הראשון כל כתובת מכילה מילת זכרון, ובמקרה השני, ה ROM מכיל אוסף של טבלאות אמת המגדירות את הפונקציות הבולאניות. עקב ריבוי החוטים ברכיבים אלו, נהוג סימון פשטני, בו כל הצטלבות היא פיוז הניתן לניתוק. הסימון החדש הינו:



ניתוק הפיוזים נקרא "צריבה", ופעולת הצריבה הינה תכנות הרכיב באופן חד פעמי.

דוגמת רכיב ROM 32x8 ניתן לראות בשרטוט הבא:



הרכיב מורכב ממפענח 5×2^5 ומטריצת פיוזים. המפענח מוציא את כל ה minterms האפשריים ומטריצת הפיוזים קובעת אילו מה minterms יגיעו לאיזה שער OR. פיוז שרוף מנתק את הקו ומסומן ב '+', פיוז שאינו שרוף מסומן בשרטוט ב 'x'. רכיב ROM יקרא ROM MxK אם יש לו מרחב כתובות בגודל M (ולכן הוא מכיל מפענח $M \rightarrow \log(M)$) ו K ביטי פלט.

רכיבים נוספים:

ROM :PROM (Programmable ROM) הניתן לצריבה ע"י המשתמש באמצעות מכשיר

מיוחד.

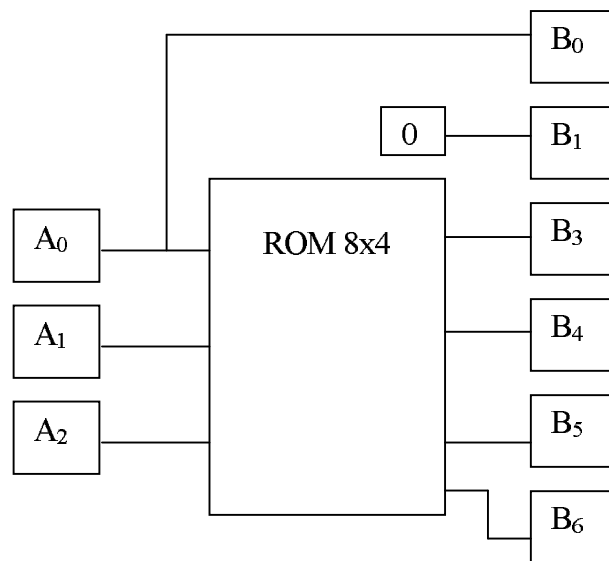
PROM :EPROM (Erasable PROM) הניתן לצריבה רב פעמית.

דוגמא 1: השתמשו ב ROM על מנת לתכנן מעגל המחשב את x^2 .

תחילה נרשום את טבלת האמת:

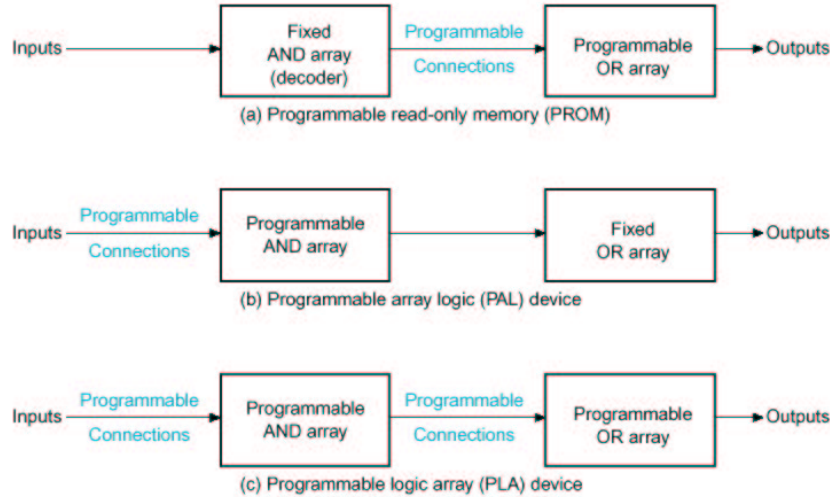
Inputs			Outputs						Decimal
A ₂	A ₁	A ₀	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

נשים-לב כי $B_1=0$ וכן $B_0=A_0$, כך שאפשר להסתפק בחישוב ארבעה פלטים. נצרום ROM כך שיכיל את טבלת האמת בטורים B₂ עד B₅, ונחווט מעגל באופן הבא:



PLD.2

PLD = Programmable Logic Device
 ישנם שלושה סוגים של PLD: PROM, PAL, PLA
 הסוגים השונים מתאימים למיקומים האפשריים של הפיזזים:

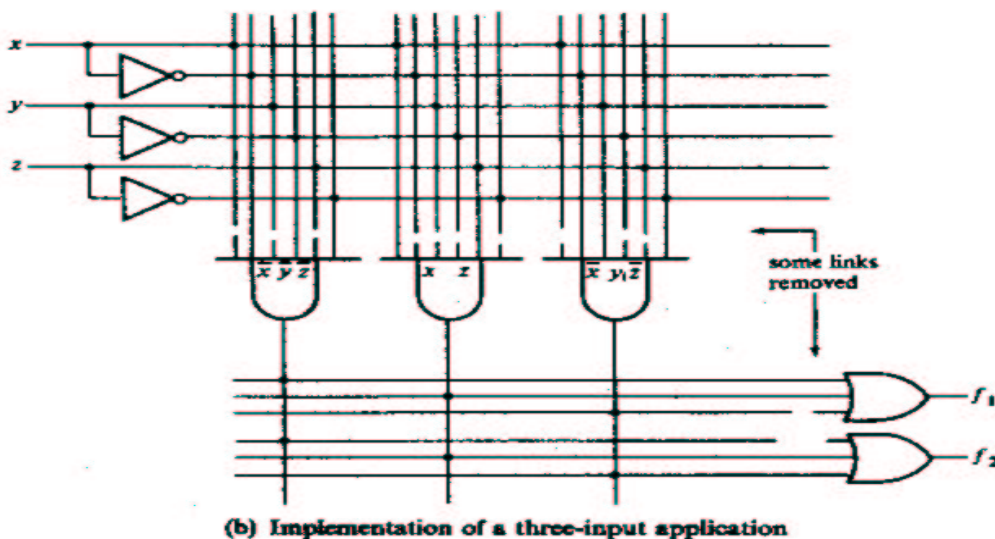


הסוג הראשון הוא PROM אותו הכרנו זה עתה. במקרה של PROM, מממש המפענח שת כל ה minterms האפשריים ע"י שערי AND מתאימים, והתכנות מתבצע ע"י בחירת ה minterms שידליקו את ביט הפלט באמצעות שערי OR. אפשרות אחרת היא לאפשר בחירת ה minterms אך לא לאפשר בחירה בשערי OR. אפשרות זו נקראת PAL. אפשרות נוספת היא, כמובן, לאפשר בחירה גם בשערי AND (בחינת ה minterms) וגם בשערי OR. אפשרות זו נקראת PLA.

PLA.3

PLA = Programmable Logic Array

דוגמא ל PLA:
 מרווח בקו הכניסה לשערי ה AND/OR משמעו פיזז שרוף.



מעגל זה מממש

$$f_1 = \bar{x} \bar{y} \bar{z} + xz \quad f_2 = xz + \bar{x}y\bar{z}$$

מקובל להוסיף ביט בקרה c המחליף, באמצעות שערי XOR, את יציאות ה PLA מ f ל not(f) עבור c=0 או c=1 בהתאמה.

רכיב נוסף: FPLA = Field PLA הינו PLA הניתן לצריבה ע"י המשתמש (ולא במפעל).

דוגמא 2:

נתונות הפונקציות

$$F_1(A, B, C) = \Sigma(0, 1, 2, 4), \quad F_2(A, B, C) = \Sigma(0, 5, 6, 7)$$

ממשו אותן באמצעות PLA.

נתחיל בצמצום הפונק' בעזרת מפות קרנו:

		BC			
		00	01	B	
A	0	1	1	0	1
	1	1	0	0	0
		C			

$$F_1 = \bar{A}\bar{B} + \bar{A}\bar{C} + \bar{B}\bar{C}$$

$$\bar{F}_1 = AB + AC + BC$$

		BC			
		00	01	B	
A	0	1	0	0	0
	1	0	1	1	1
		C			

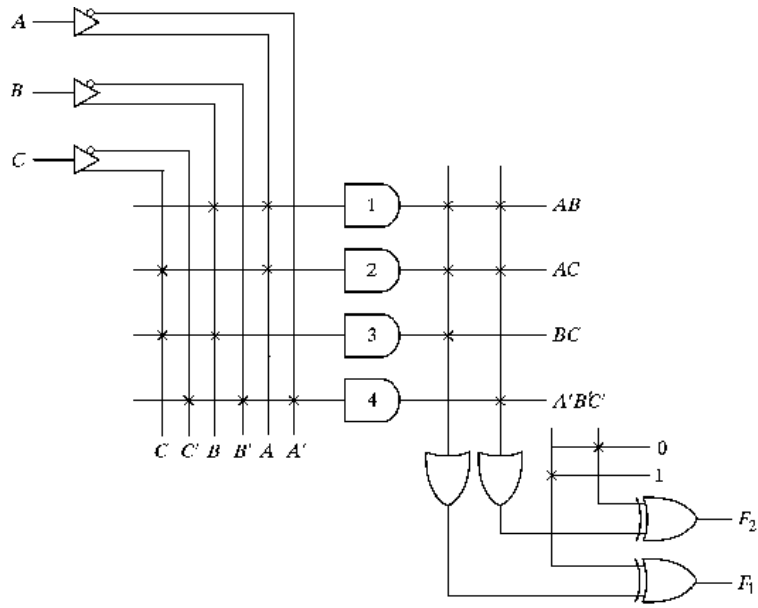
$$F_2 = AB + AC + \bar{A}\bar{B}\bar{C}$$

$$\bar{F}_2 = \bar{A}\bar{C} + \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C}$$

הממוש החסכוני ביותר יהיה כאשר נממש את שלילת F₁ ואת F₂ נממש כרגיל. נקבל את טבלת ה PLA הבאה:

PLA programming table					
	Product term	Inputs A B C	Outputs		
			(C) F ₁	(T) F ₂	
AB	1	1 1 -	1	1	
AC	2	1 - 1	1	1	
BC	3	- 1 1	1	-	
$\bar{A}\bar{B}\bar{C}$	4	0 0 0	-	1	

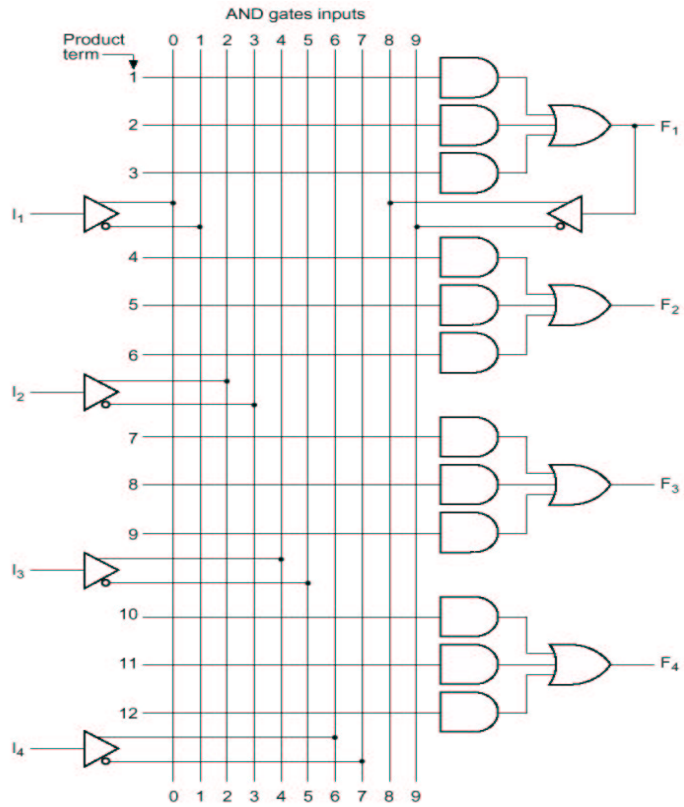
לאחר הצריבה, המעגל יראה כך:



PAL.4

PAL = Programmable Array Logic

כשישנם הרבה משתני כניסה, ופונקציות הפלט משתמשות רק במעט מה minterms, עדיף להשתמש ב-PAL. מעגל PAL לפני צריבה נראה כך:



שימו לב כי F1 ניתן ל"שימוש חוזר".

דוגמא 3:

בתונות ארבעת הפונקציות:

$$W(A, B, C, D) = \Sigma(2, 12, 13),$$

$$X(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15),$$

$$Z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$$

נממש אותן באמצעות PAL.

נפשט את הפונק' ונקבל:

$$W(A,B,C,D)=ABC'+A'B'CD'$$

$$X(A,B,C,D)=A+BCD$$

$$Y(A,B,C,D)=A'B+CD+B'D'$$

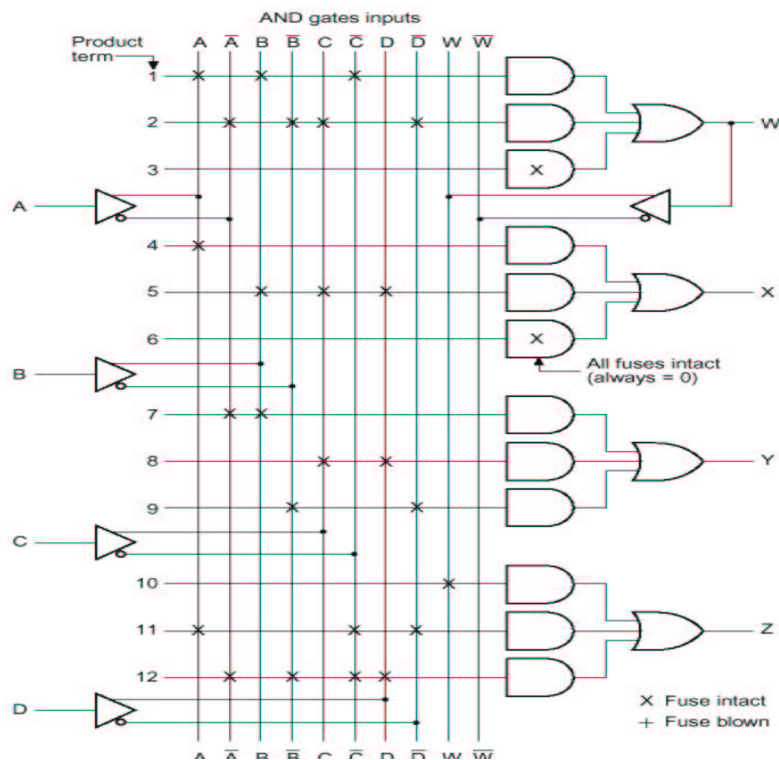
$$Z(A,B,C,D)= ABC'+A'B'CD' +AC'D'+A'B'C'D$$

$$\rightarrow Z(A,B,C,D)=W+AC'D'+A'B'C'D$$

טבלת ה PAL המתקבלת:

Product term	AND Inputs				W	Outputs
	A	B	C	D		
1	1	1	0	—	—	$W = \overline{ABC} + \overline{A}BCD$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$X = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	$Y = \overline{AB} + CD + \overline{B}D$
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$Z = \overline{W} + \overline{ACD} + \overline{A}BCD$
11	1	—	0	0	—	
12	0	0	0	1	—	

שימו לב כי הפלט W משמש לחישוב Z. המעגל המתקבל:



4. תרגיל לדוגמא

נתונה הפונק' הבאה:

$$f(t+1,b) = ((b+1)f(t) + b) \text{ mod } 7$$

כאשר b הינו ביט קלט בינארי.

- א. רשום את טבלת המעברים של f.
- ב. שרטט אוטומט המממש את f.
- ג. ממש את האוטומט באמצעות ROM 16x4 ושלשה D-FF.
- ד. נתח את התנהגות האוטומט.

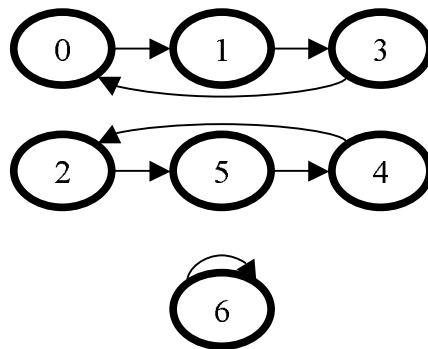
פתרון:

א.

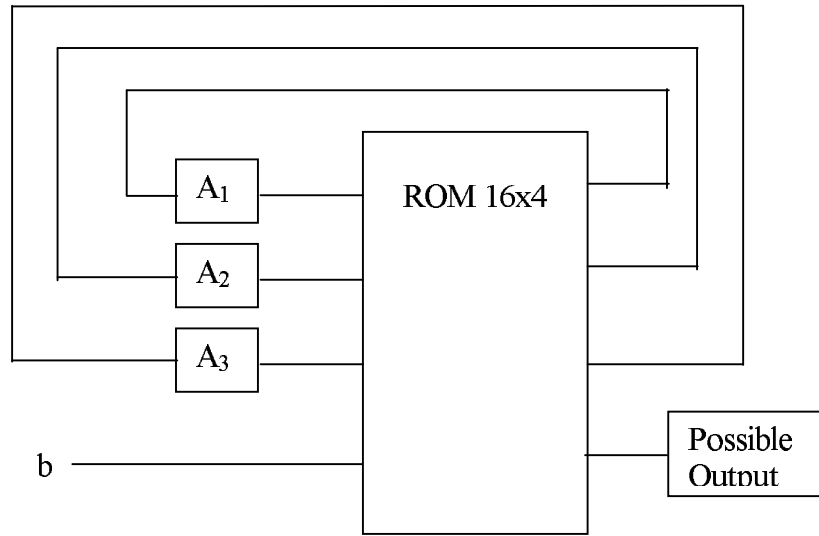
תחילה נשים לב כי אם $b=0$ אז $f(t+1,0) = f(t)$ ולכן נרשום את הטבלה עבור $b=1$ בלבד.

f(t)	b	f(t+1)	Possible Output
0	1	1	
1	1	3	
2	1	5	
3	1	0	
4	1	2	
5	1	4	
6	1	6	

ב. מכיון שקלט 0 משאיר את המצב הקודם, מתוארות הקשתות עבור קלט 1 בלבד. בכל מצב יש להוסיף לולאה המחזירה לאותו המצב עבור הקלט 0.



ג.



ד. למעגל שלש "לולאות" בלתי תלויות:

(a) מצבים 0, 1, 3

(b) מצבים 2, 4, 5

(c) מצב 6

כפי שנואמר למעלה, על קלט 0 המעגל נשאר באותו המצב. עבור קלט 1, לעומת זאת, המעגל יעבור על סדרת המצבים הכלולים ב"לולאה".