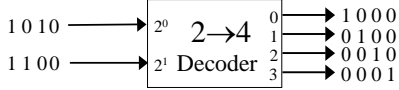


**רכיבים סטנדרטיים**

**מפענח – Decoder**

- n כניסות מייצגות מספר בינארי בתחום  $0 \dots 2^n - 1$
- $2^n$  יציאות כאשר יציאה i הינה "1" אם"ם בכניסה יש ערך i.

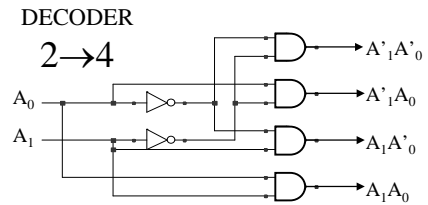


קלט		פלט			
A <sub>1</sub>	A <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

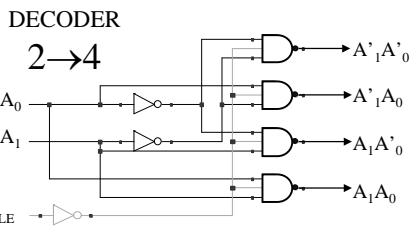
**רכיבים סטנדרטיים**

**מפענח – Decoder**

שימו לב שהפונקציה שמחשבת היציאה ה- i של המפענח היא ה- m<sub>i</sub> minterm.



**הרחבות למפענחים**



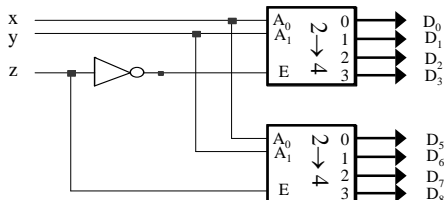
**הרחבות למפענחים**

- היציאות יכולות להיות מהופכות (שימוש בשערי NAND במקום AND).
- קיים קו Enable אשר "מפעיל" או "משתק" את כל הרכיב. טבלת האמת עבור שערי NAND ו- Enable מופעל ע"י אות "0".

E	A <sub>1</sub>	A <sub>0</sub>	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	∅	∅	1	1	1	1

**שימוש ב- Enable**

נממש מפענח  $2^n \rightarrow n$  ע"י שני מפענחים  $2^{n-1} \rightarrow n-1$ , ושימוש בקו Enable:



- כאשר  $Z=0$  יופעל ה-Decoder העליון. במקרה זה ה-Decoder התחתון יוציא אפסים.
- התפקוד משתנה (עליון יוציא 4 אפסים) כאשר  $Z=1$ .

**מימוש סכום מכפלות ע"י מפענחים**

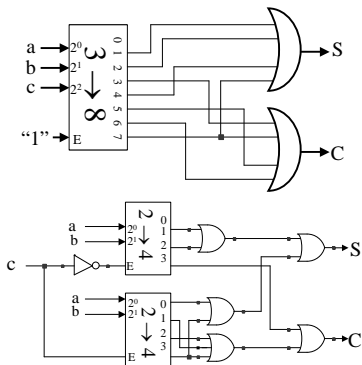
- ע"י לקיחת OR של היציאות המתאימות נוכל לממש סכום מכפלות.
- מכיוון שכל המכפלות הינן יציאות ה-Decoder ניתן לממש יותר מפונקציה אחת ע"י Decoder יחיד.

דוגמא: מימוש FA ע"י Decoder.

$$S = A \oplus B \oplus C = \Sigma(1,2,4,7)$$

$$C = AB + BC + AC = \Sigma(3,5,6,7)$$

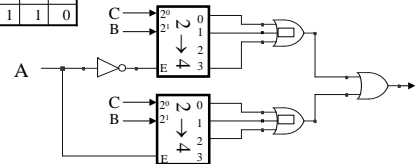
**מימוש FA ע"י מפענה 3→8**



**מימוש פונקציות ע"י מפענחים - חשיבות הבחירה של סדר המשתנים**

דוגמה:  
 $f(A, B, C) = \sum(0, 1, 3, 4, 5, 6)$

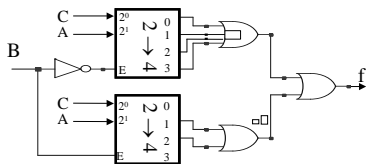
A	B	C	f
0	0	0	1
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



**חשיבות הבחירה של סדר המשתנים - המשך**

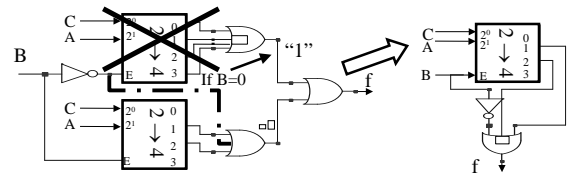
נשנה את סדר המשתנים:

B	A	C	f
0	0	0	1
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	0



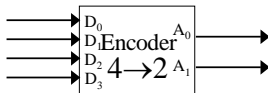
**חשיבות הבחירה של סדר המשתנים - המשך**

נקבל מימוש יותר יעיל:



**מקודדים - Encoders**

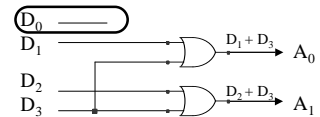
- מממש פונקציה הפוכה למפענח.
- $2^n$  קוי כניסה.
- n קוי יציאה.



קלט	קלט	פלט	פלט		
$D_3$	$D_2$	$D_1$	$D_0$	$A_1$	$A_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

• קלט שאינו "אונארי" יביא ליציאה שגויה או לא מוגדרת.

**מימוש מקודדים**



(קו  $D_0$  אינו מחובר).

ישנן הרחבות למקודדים מורכבים וכלליים יותר:

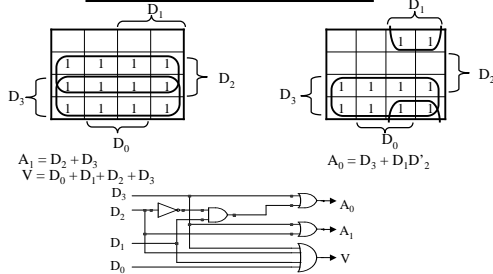
**מקודד עדיפויות - Priority Encoder**

- $2^n$  כניסות.
- n יציאות + יציאה נוספת (יציאת Valid) הבודקת את תקינות הקלט.

### מקודד עדיפויות – Priority Encoder

הפלט מציין את הבית הראשון (MSB) שהינו "1".

D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	A <sub>1</sub>	A <sub>0</sub>	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	∅	0	1	1
0	1	∅	∅	1	0	1
1	∅	∅	∅	1	1	1



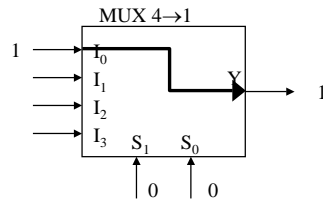
### מרבב – Multiplexer

• 2<sup>n</sup> קוי כניסה.

• n קוי ברירה / בחירה / מיעון.

• קו יציאה אחד.

מרבב ספרתי בורר קו יחיד מבין מספר קווי כניסה ומכוון את המידע הבינארי אל קו היציאה.



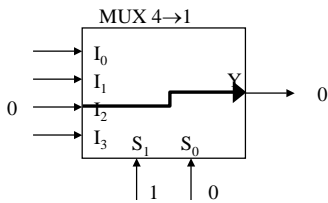
### מרבב – Multiplexer

• 2<sup>n</sup> קוי כניסה.

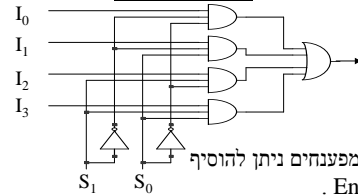
• n קוי ברירה / בחירה / מיעון.

• קו יציאה אחד.

מרבב ספרתי בורר קו יחיד מבין מספר קווי כניסה ומכוון את המידע הבינארי אל קו היציאה.

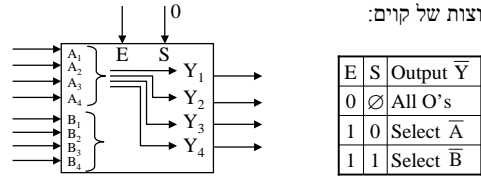


### מימוש מרבב



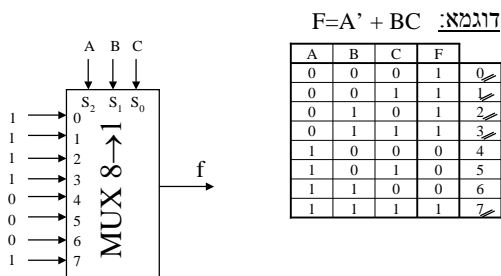
• בדומה למפענחים ניתן להוסיף קו Enable.

• ניתן גם לתכנן מרבב הבורר בין קבוצות של קוים:



### מימוש פונקציות בוליאניות ע"י מרבבים

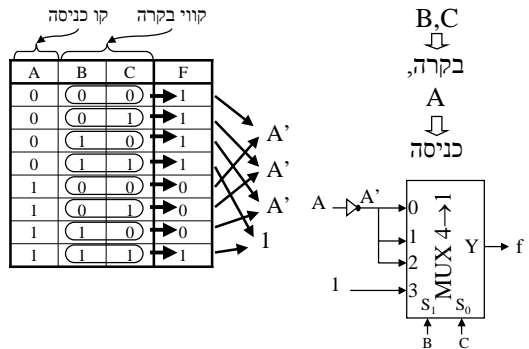
גישה "ישירה": עבור פונקציה בעלת n משתנים השתמש במרבב בעל n כניסות בקרה. רשום הפונקציה כסכום מכפלות. הכנס 1 לכניסה המתאימה למכפלה אשר מופיעה בסכום ו-0 אחרת.



### גישה חסכונית

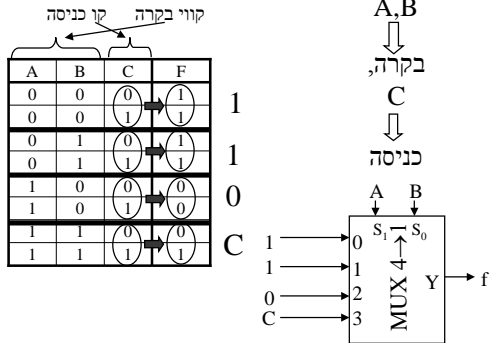
• חלוקת משתני הפונקציה בין קווי הבקרה וקווי הכניסה.

• חלוקה סטנדרטית:



### גישה חסכונית - המשך

- חלוקת משתני הפונקציה בין קוי הבקרה וקוי הכניסה.
- חלוקה סטנדרטית:



### דוגמא: חלוקה ל-2 קווי בקרה ו-2 קווי כניסה

