פרק 9

תכנון לוגי – מערכות צירופיות

9.1 מבוא .

מעגלים לוגיים עבור מערכות ספרתיות יכולים להיות צירופיים או סדרתיים.

מעגל צירופי מורכב משערים לוגיים אשר יציאותיהם בכל זמן נתון נקבעות ישירות מהצירוף של הכניסות באותו זמן וללא קשר אל הכניסות בעבר.

מעגל סדרתי מורכב משערים לוגיים וגם מיחידות זכרון. יציאות המעגל הן פונקציה של הכניסות ושל מצב יחידות הזיכרון. (מצב יחידות הזיכרון הוא פונקציה של כניסות בעבר) .

מעגל צירופי מורכב ממשתני כניסה, שערים לוגיים ומשתני יציאה. תרשים מלבני של מעגל צירופי :

n משתני הכניסה הבינריים (כל קו הוא '0' או '1') באים ממקור חיצוני. m - משתני היציאה מועברים ליעד חיצוני. ניתן לתאר מעגל צירופי באמצעות m פונקציות בוליאניות. (פונקציה אחת עבור כל משתנה יציאה. כל פונקציה מורכבת מ-n משתנים . (משתני הכניסה).

נוהל עיצוב (תיכנון) של מעגל צירופי.

עיצובו של מעגל צירופי מתחיל בתיאור מילולי של הבעיה ומסתיים בדיאגרמה של מעגל לוגי או בקבוצה של פונקציות שניתן לקבל מהן בקלות את הדיאגרמה הלוגית.

שלבי נוהל עיצוב.

א. תיאור הבעיה.

ב. קביעת מספר משתני הכניסה הקיימים ומספר משתני היציאה הנדרשים.

ג. סימון משתני כניסה ויציאות (שמות).

ד. בניית טבלת אמת.

ה. פישוט הפונקציות הבוליאניות (עבור כל יציאה זה פונקציה אחרת).

ו. שרטוט דיאגרמה לוגית.

את פונקציות הפלט אפשר לפשט מתוך טבלת האמת בצורות שונות כגון : שיטת המפה, שיטת הטבלה, אמצעים אלגברים ועוד. צורות הפישוט השונות גורמות לקבלת ביטויים שונים שמתוכם לבחור את המתאים לנו.

שיטה מעשית לפישוט חייבת לקחת בחשבון מספר אילוצים :

א. מספר מינימלי של שערים.

ב. מספר מינימלי של כניסות לשער.

ג. זמן השהיה מינימלי.

ד. מספר חיבורים פנימיים מינימלי.

ה. הגבלה על מספר החיבורים של האות לכניסות לשערים שונים . (בלועזית : FAN OUT)

חלק מהאילוצים סותרים לעיתים אחד את השני למשל א' וג' , לכן לא תמיד ניתן לקיים את כל האילוצים בו זמנית.

9.2 <u>מימוש פונקציות בשיטת מימוש ישיר</u>.

כללית : בשיטה זו מממשים באמצעות שערים (OR, AND ו- NOT)_ .

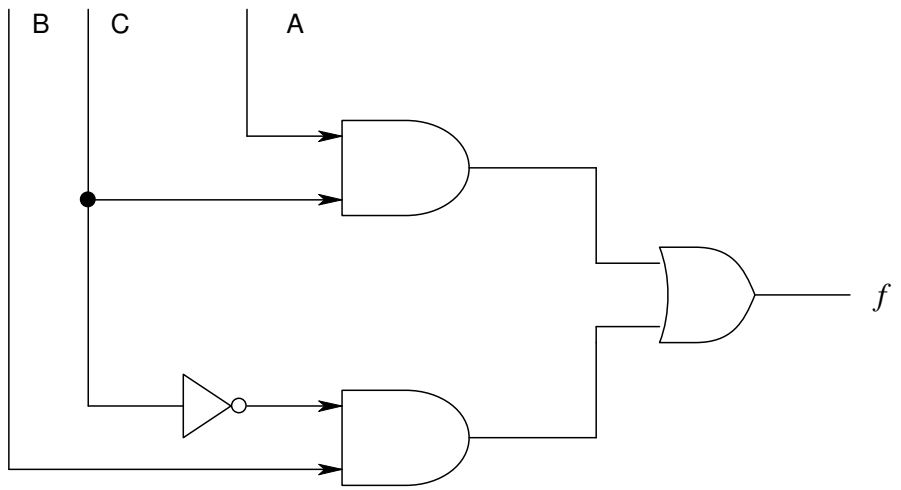
<u>דוגמא</u> : נתונה ה פונקציה : $f(A,B,C)=\sum 2,5,6,7$ A - סיבית M.S.B .

נדרש לממש את f באמצעות שערים בסיסיים NOT, OR, AND . כדי לחסוך בשערים, נפשט את f ע"י מפת קרנו.

C \ AB	00	01	11	10
0		1	1	
1			1	1

תוצאת הפישוט הינה : $f = B\overline{C} + AC$

מימוש הפונקציה המפושטת ע"י שערים:



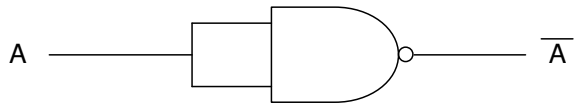
בשיטה שהצגנו כאן מימשנו את הפונקציה המפושטת בדיוק כפי שהיא כתובה. <u>יתרון</u> : פשטות במימוש.

<u>חסרון של מימוש זה</u> : השתמשנו כאן בשלושה מעגלים משולבים (שבכל אחד כמה שערים) למרות שניצלנו רק חלק מהשערים.

כדי לממש בצורה יותר יעילה (ניצול יעיל יותר של שערים בתוך הרכיבים) נלמד שיטות שונות שלאו דווקא נותנות מספר שערים מינימלי , <u>אבל</u> שיחסכו במספר המעגלים משולבים.

9.3 מימוש באמצעות שערי NAND בלבד.

שער NOT אפשר לממש באמצעות NAND באופן הבא :

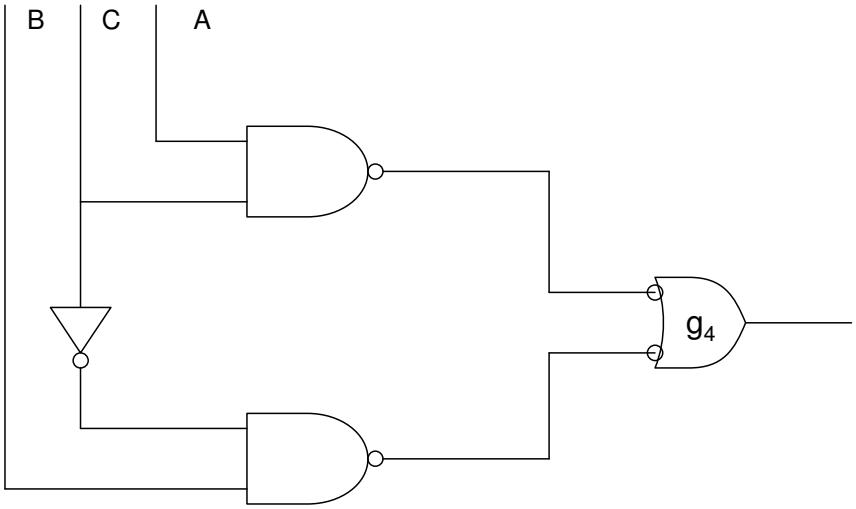


נתבונן במימוש הישיר לעיל, ונממש באמצעות שערי NAND בלבד. מעבר משער מסוים לשער NAND יעשה באמצעות שיטה הנקראת ההפוך הכפול. שיטה זו מתבססת על משוואה : $((X)')' = X$

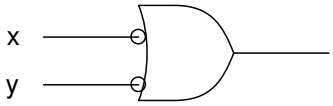
עקרון זה מתואר בציור הבא :

הסבר לשקילות :

עיגולים מציינים הפוכים (NOT). שני עיגולים על אותו קו בשרטוט מבטלים זה את זה כלומר על קו כלשהו ניתן להוסיף תמיד מספר זוגי של הפוכים (עגולים) מבלי לשנות את הפונקציה . נפעיל את עקרון ההיפוך הכפול על הדוגמא בעמוד קודם :

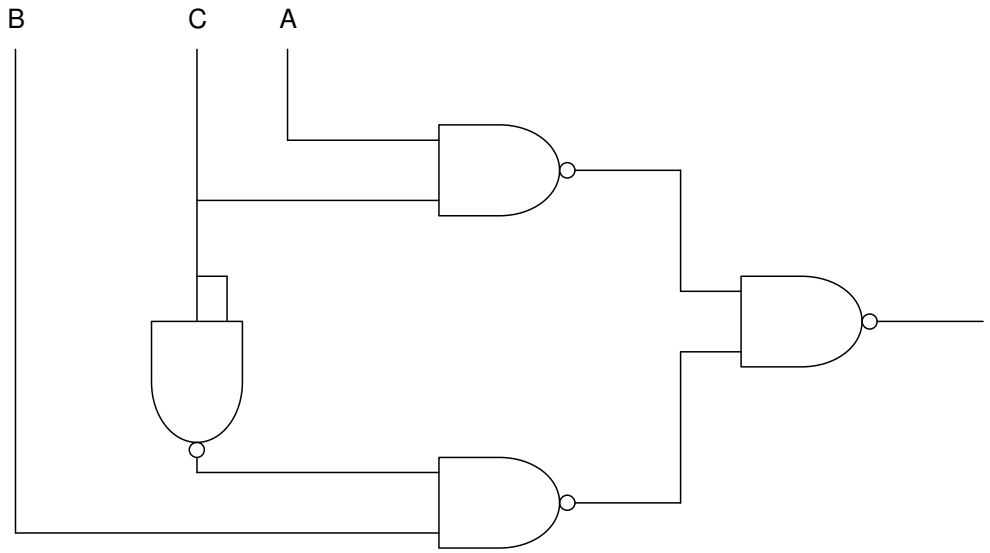


כעת : $\overline{x} + \overline{y} = \left(\overline{xy}\right)$



כלומר שער g_4 שקול לשער NAND .

ולכן קיבלנו :



סיכום תהליך מימוש באמצעות NAND בלבד :

א. פשט פונקציה לפי מפת קרנו לסכומי מכפלות SOP.

ב. שרטט מימוש ישיר באמצעות שערי AND, OR ו-NOT.

ג. הפעל את עקרון ההפוך הכפול על שערי AND ו-OR.

ד. המר שער NOT בשער NAND ע"י חיבור שתי כניסות יחד.

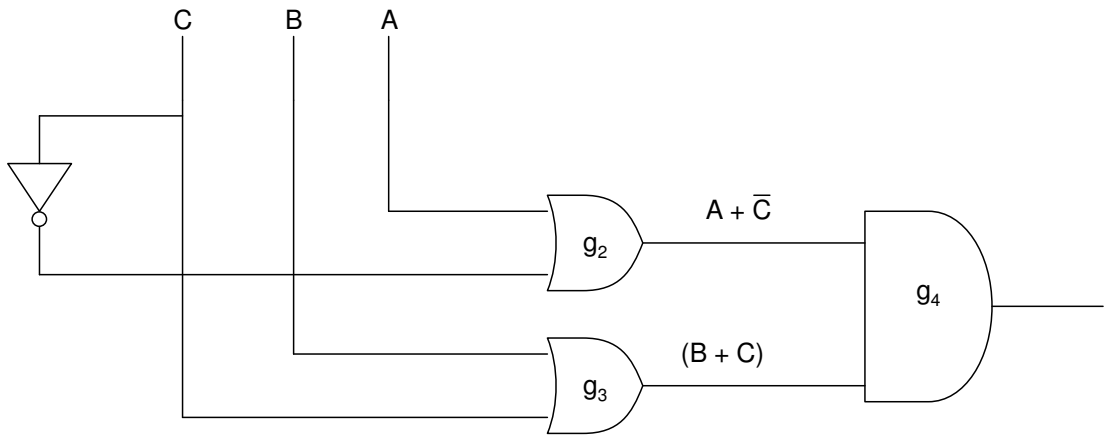
9.4 <u>מימוש פונקציות באמצעות שערי NOR בלבד.</u>

דוגמא : $f(A,B,C)=\sum 2,5,6,7$. נפשט את הפונקציה בצורת <u>מכפלת סכומים</u> (POS)

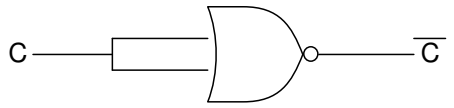
C \ AB	00	01	11	10
0	0			0
1	0	0		

$$f(A,B,C)=(B+C)(A+\overline{C})$$

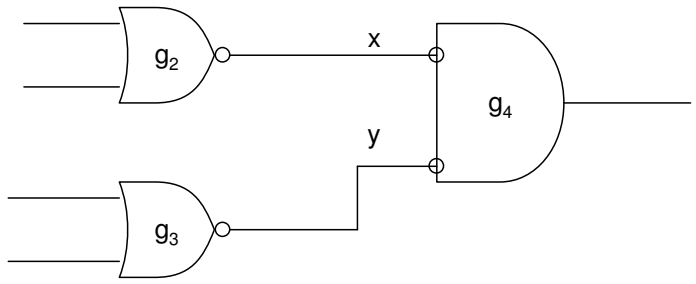
מימוש הפונקציה באמצעות שערי : OR ,NOT , AND



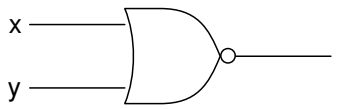
שער NOT אפשר לרשום כשער NOR ע"י חיבור האות לשתי כניסות :



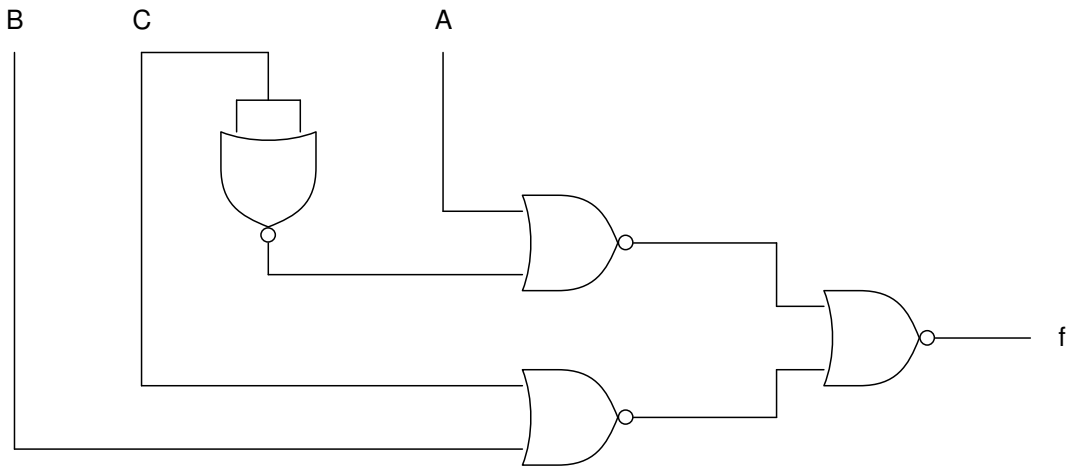
כעת לגבי שאר השערים נפעיל את שיטת הפוך :



כיוון ש- $\overline{x} \cdot \overline{y} = (x + y)'$ לכן השער g_4 שקול לשער :



המימוש באמצעות שערי NOR בלבד הינו :



<u>סיכום תהליך למימוש באמצעות שער NOR</u>

א. פשט את הפונקציה ותאר אותה כמכפלת סכומים (POS).

ב. שרטט מימוש (פשוט) ישיר באמצעות שערי AND, OR, ו-NOT.

ג. הפעל את עקרון ההפוך הכפול של שערי ה-AND וה-OR כך שתקבל שערי NOR.

ד. המר את שערי ה-NOT בשערי NOR בעל כניסות שמחוברות זו לזו.

9.5 <u>סימולים חלופיים של שערים</u>

השער	הפונקציה	סימול 1	סימול 2
NOT	$Y = \overline{A}$	A Y	A
AND	Y =AB	A B y	A B
OR	A + B	A B	A B
NAND	$\overline{AB}$	A B	A B
NOR	$\overline{A + B}$	A B	A B

הערה : אפשר לאמת את שקילות הסימולים לפי חוקי דה-מורגן.

9.6 מערכות צירופיות שימושיות

מבוא : בפרק זה נתאר מספר מערכות צירופיות נפוצות. מערכות אלו מתחלקות לשני סוגים עיקריים : מערכות חישוב ומערכות לניתוב נתונים.

9.6.1 מערכות חישוב.

9.6.1.1 מסכמים

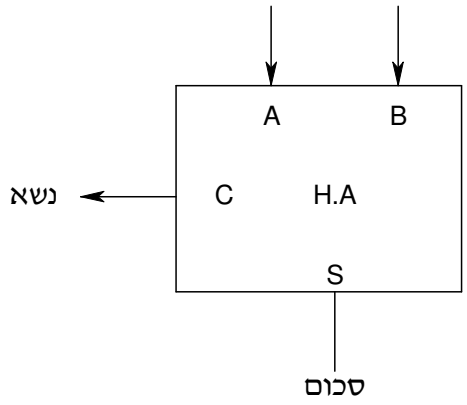
מסכם הינו מערכת צירופית אשר מבצעת פעולת חיבור בינרי. קיימים 2 סוגי מסכמים :

א. מסכם למחצה (H.A – HALF ADDER)

H.A זו מערכת שמבצעת חיבור בינרי על שתי סיביות בלבד. (אין "נשא" כניסה) .למערכת שתי כניסות נתונים : הסיביות שעליהן מתבצעת פעולת החיבור (A, B)

ושתי יציאות : סיבית הסכום S וסיבית נשא C.

תרשים מלבני של H.A :



טבלת האמת של ה- H.A

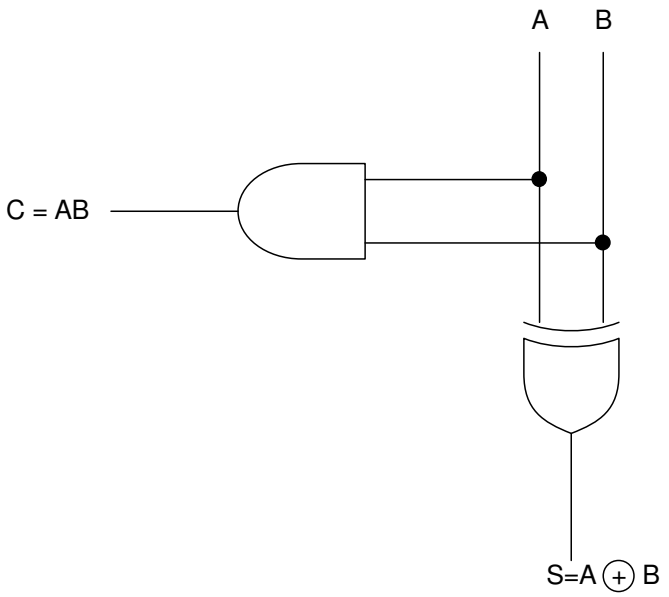
כניסות		נשא	סכום
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

מטבלת האמת מקבלים ישירות :

$$S = \overline{A} \cdot B + A\overline{B} = A \oplus B \qquad C = A \cdot B$$

הערה : C, S זה למעשה שתי סיביות התוצאה. (כאשר C הינה סיבית ה-MSB)

מימוש של ה- H.A בצורה ישירה:



ב. מסכם מלא- FULL ADDER (F.A)

-הקדמה : כאמור לעיל מסכם למחצה מבצע חיבור בינרי של שתי סיביות בלבד (ללא נשא קודם).

ניזכר במקרה כללי של חיבור שני מספרים בינריים X ו-Y כל אחד בן n סיביות.

	דרגה 0	דרגה 1	דרגה 2	.....	דרגה k	דרגה n-1
סיביות נשא ←		C_1	C_2		C_k	C_{n-1}
מחובר ראשון – X :	X_0	X_1	X_2		X_k	X_{n-1}
מחובר שני – Y :	Y_0	Y_1	Y_2		Y_k	Y_{n-1}
	S_0	S_1	S_2		S_k	S_{n-1}

הסבר : פעולת החיבור בין X ו- Y מתבצעת ע"י חיבור הסיביות המתאימות.

הסיביות הראשונות (X_0, Y_0) מחוברות ביניהן (אין שום "נשא" קודם עדין) סכומם הוא S_0 ונשא נרשם בתור C_1 . (רושמים מעל דרגה 1 חיבור 1).

שלב שני : מחברים את (X_1, Y_1) בתוספת (נשא מדרגה קודמת) C_1 התהליך חוזר על עצמו עד לחיבור של $X_{(n-1)}$ $Y_{(n-1)}$ ו- C_{n-1} (שהוא נשא מדרגה $n-2$). תוצאת החיבור היא S_{n-1} והנשא C_n הוא סיבית M.S.B : S_n .

דוגמא : חיבור בינרי 100 + 101

נשא ← 100
101
+
100
תוצאה ← 1001

מהדוגמא האחרונה ניתן לראות כי חיבור בין שתי סיביות ה- L.S.B X_0 ו- Y_0 הינו פשוט, כיוון שאין צורך לחבר נשא ולכן ניתן לממש דרגה ראשונה באמצעות H.A. כאשר הכניסות יהיו: Y_0, X_0, והיציאות: S_0 ו- C_1.

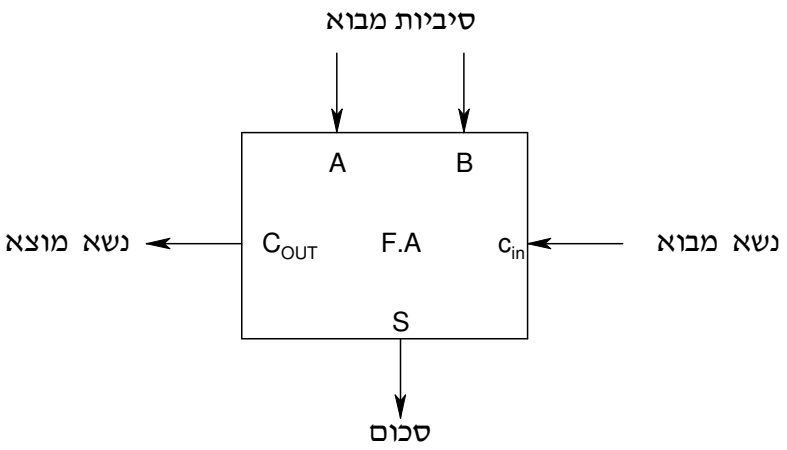
עמודת חיבור ראשונה נקראת דרגה 0 עמודה k נקראת דרגה k. חיבור של דרגה אחרונה הינו דרגה $(n-1)$.

בשאר הדרגות אחרי דרגה ראשונה יש צורך לחבר בנוסף לסיביות הכניסה גם נשא מדרגה קודמת.

- **<u>מסכם מלא</u> (F.A.- FULL ADDER)** הינו מערכת שמחברת שתי סיביות מבוא ונשא מבוא ומפיקה סיבית סכום וסיבית נשא. [בהתייחס למימוש של החיבור הכללי שהראנו: חוץ מדרגה 0 שהינה H.A הרי שכל שאר דרגות החיבור הינן למעשה מורכבות מ- F.A.]

תרשים מלבני של מסכם מלא F.A .

מבחינה <u>אריתמטית</u> $S = A + B + C_n$



טבלת האמת של המסכם המלא:

כניסות				יציאות	
נתונים	סיביות	נשא מבוא		נשא מוצא	סכום
A	B	C_{in}		C_{OUT}	S
0	0	0		0	0
0	0	1		0	1
0	1	0		0	1
0	1	1		1	0
1	0	0		0	1
1	0	1		1	0
1	1	0		1	0
1	1	1		1	1

ניתן להראות מטבלת האמת כי משוואות של S ו- C_{OUT} הינן :

$$S = A \oplus B \oplus C_{in}$$

$$C_{OUT} = AB + C_{in}\left(A \oplus B\right)$$

או בצמצום ממפת קרנו :

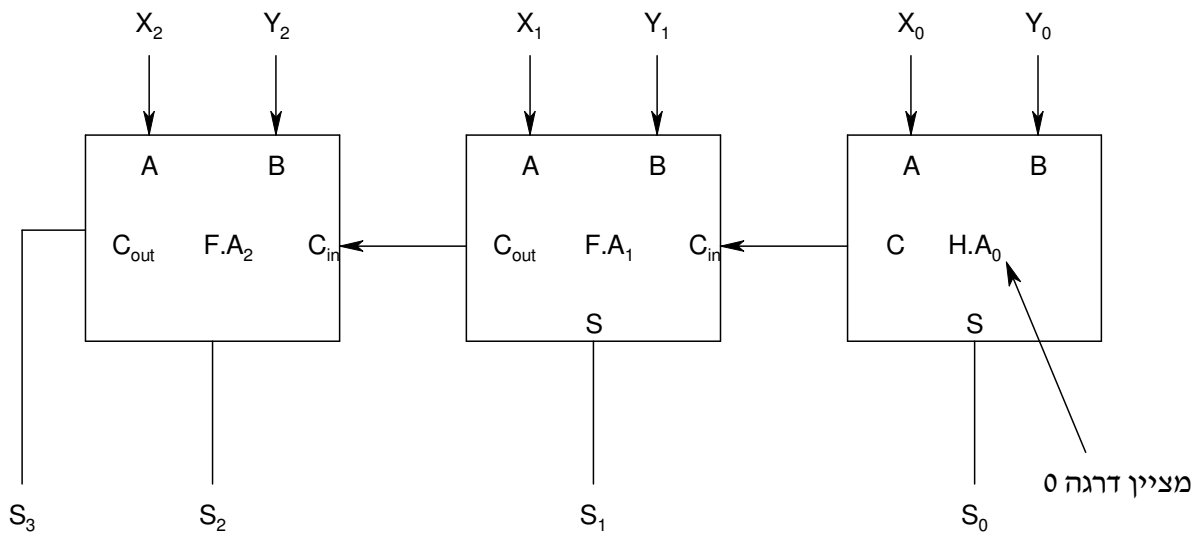
$$C_{OUT} = A \cdot B + C_{in}(A + B)$$

ניתן לממש משוואות אלו באמצעות שערי XOR, AND ו-OR.

ג. **מסכם גלי** (RIPPLE ADDER)

ניתן לחבר שני מספרים בעלי n סיביות כל אחד באמצעות : H.A אחד (לדרגה ראשונה – 0) ו- $(n-1)$ מסכמים מלאים F.A לדרגות הבאות .

דוגמא : מערכת לחיבור שני מספרים בעלי שלוש סיביות. למערכת זו קוראים מסכם גלי.



עבור $i \geq 1$ C_i - כניסה לדרגה i ונשא יציאה של דרגה ($i-1$).

במערכה זו דרגה 0 מסכמת את סיביות ה- L.S.B של X ו-Y, מפיקה סכום S_0, ונשא C_1.

דרגה 1 מסכמת את סיביות : X_1, Y_1 ו- C_1 ומפיקה את סכום S_1 ונשא C_2, וכן הלאה.

<u>שים לב לנקודה חשובה</u> : הכניסות X ו-Y של כל הדרגות מתקבלות באותו זמן, אולם כל עוד הנשא C_1 לא מתעדכן ע"י דרגה 0, הרי S_1 ו- C_2 לא יהיו נכונים.

בצורה דומה כל עוד C_2 לא מתעדכן ע"י דרגה 1 אזי S_2 ו- C_3 לא יהיו נכונים וכן הלאה. כלומר, נשאי המוצא השונים (של המסכמים) מתייצבים (כלומר מקבל ערך נכון שהוא <u>סופי</u>) לפי סדר מקומם החל ב- C_1 וכלה ב- C_n. תופעה זו נקראת <u>התפשטות גלית</u> של הנשא ומכן השם : מסכם גלי.

<u>דוגמא</u> : נתבונן במסכם הגלי של 3 סיביות ונבדוק כמה זמן עובר משעה שקיבלנו נתונים X ו- Y עד שהסכום הינו סופי נכון.

לשם כך נניח שכל דרגת חיבור יש לה זמן השהיה t_{PD} . נניח שזמן התחלה $t=0$ זה זמן הגעת X ו-Y.

א. אחר זמן $t=t_{PD}$, S_0 ו- C_1 נכונים (שים לב ששאר הדרגות למשל דרגה 1 לא מפיקה תוצאה נכונה כי C_1 עדיין לא נכון).

ב. אחרי זמן $t=2t_{PD}$ גם S_1 ו- C_2 נכונים.

ג. אחרי זמן $t=3t_{PD}$ כל תוצאה נכונה (כלומר סכום נכון).

<u>חסרון של מסכם גלי</u> : צריך להמתין עד שיוצרו כל הנשאים הנכונים ורק אז תוצאה נכונה.
כדי להתגבר על בעיה זו משתמשים בחישוב מהיר כפי שנראה בהמשך שבו מחשבים את כל ה"נשאים" במקביל כפונקציה של כניסות X ו- Y <u>בלבד</u> וללא תלות בנשאים קודמים.

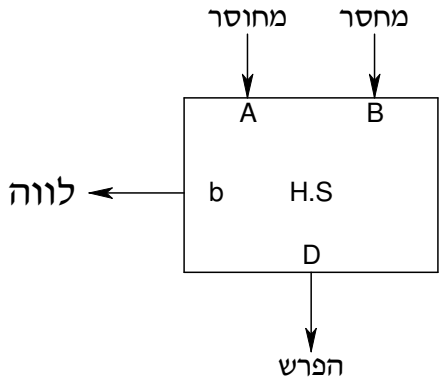
9.6.1.2 **<u>מחסרים.</u>**

קיימים 2 סוגי מחסרים : מחסר למחצה ומחסר מלא .

א. <u>מחסר למחצה</u> . H.S. – HALF SUBTRACTOR

זוהי מערכת צירופית בעלת שתי כניסות (מבואות) : סיבית מחוסר A וסיבית מחסר B. פעולה נדרשת : $(A-B)$ למערכת יש 2 <u>שתי יציאות</u> : סיבית הפרש D וסיבית ליוה – (BORROW) שמסומנת - b.

תרשים מלבנים של מחסר למחצה (H.S) . המחסר מבצע $A-B$. **התוצאה הינה : (b,D)**



טבלת האמת של ה- H.S

כניסות		יציאות	
A	B	b	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

אפשר להראות $D = \overline{A} \cdot B + A\overline{B} = A \oplus B$

$$b = \overline{A} \cdot B$$

הערה : $0 - 1 = (-1) = (11)_2$ ובמשלים ל-2 הערך של 1- הוא 11.

מימוש פשוט אפשרי באמצעות שערים : XOR, AND ו- NOT.

ב. מחסר מלא F.S.

בפעולת חיבור של שני מספרים בינריים בעלי n סיביות היה עלינו להתחשב בנשא אפשרי מדרגה קודמת לצורך חיבור הדרגה הבאה, בדומה לכך בפעולת חיסור של מספרים בגודל n סיביות עלינו להתחשב בלווה (BORROW) אפשרי (אם היה) מדרגת חיסור קודמת (ולא רק בסיביות המחוסרות).

תרשים מלבני של מחסר מלא (F.S)

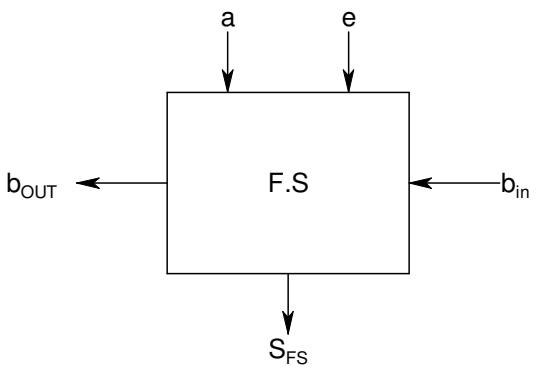
רכיב זה מבחינה אריתמטית מבצע את הפעולה : $a - e - b_{in}$

a – נקרא מחוסר e – מחסר b_{in}- לווה (BORROW) בכניסה.

התוצאה בצורה בינרית מורכבת משתי סיביות : (S_{FS} , b_{out}), כאשר :

b_{out} - לווה יציאה

S_{FS} - סיבית ההפרש



<u>טבלת האמת של המחסר המלא</u> :

תוצאה בינרית		תוצאה עשרונית			
S_{FS}	b_{OUT}		b_{in}	e	a
0	0	0	0	0	0
1	1	-1	1	0	0
1	1	-1	0	1	0
0	1	-2	1	1	0
1	0	1	0	0	1
0	0	0	1	0	1
0	0	0	0	1	1
1	1	-1	1	1	1

שים לב שתוצאה בינרית מורכבת משתי סיביות (b_{out}, S_{FS}) וזה מתאים לתוצאה העשרונית הרצויה בשיטת משלים ל-2 למשל: (-1) במשלים ל- 2 : 11 -2 במשלים ל- 2 : 10.

<u>מפת קרנו של</u> S_{FS} :

b_{in} \ ae	00	01	11	10
0		1		1
1	1		1	

ניתן לאמת שערכו S_{FS} הינו : $S_{FS} = a \oplus e \oplus b_{in}$

מפת קרנו של b_{OUT} :

b_{in} \ ae	00	01	11	10
0		1		
1	1	1	1	

ממינמזציה של המפה מקבלים:

$$b_{OUT} = \bar{a}e + \bar{a}b_{in} + e\,b_{in}$$

ניתן להראות כי הביטוי הבא הינו גם נכון :

$$b_{OUT} = b_{in} \cdot e + \bar{a} \cdot \left(e \oplus b_{in}\right)$$

באמצעות המחסרים המלאים וחצי מחסר ניתן לחסר מספרים בינריים בני n סיביות כל אחד. מערכת זו נקראת <u>מחסר גלי</u>. (בדומה למסכם גלי שראינו).

9.6.1.3 נשא צפוי מראש { CARRY LOOK AHEAD }.

נתיחס למסכם מלא עבור זוג סיביות (A_i, B_i) עם נשא כניסה C_i ונשא יציאה C_{i+1} הסכום כאמור לעיל הינו : $S_i = A_i \oplus B_i \oplus C_i$. $C_{i+1}=A_i \bullet B_i + C_i(A_i \oplus B_i)$

נגדיר שתי פונקציות חדשות P_i ו- G_i בצורה הבאה :

$P_i = A_i \oplus B_i$ - נקראת פונקצית "קדום נשא" .

$G_i = A_i \cdot B_i$ - נקראת פונקצית "יצירת נשא" .

לפי הסימון החדש נקבל : $C_{i+1} = G_i + C_i \cdot P_i$

הסברים לשמות הפונקציות :

א. G_i - **יצירת נשא**

כאשר $A_i=B_i=1$ ($G_i=1$) אזי $C_{i+1} = 1$, כלומר $G_i=1$ מבטא את העובדה שנוצר נשא במסכם i והוא : C_{i+1} , ללא תלות בנשא כניסה C_i.

ב. P_i - **קדום נשא**

כאשר $P_i = A_i \oplus B_i = 1 \Leftarrow$ כניסה אחת 0 ושנייה 1 (כוונה לכניסות A_i B_i) ולכן $G_i = A_i \cdot B_i = 0$

ולכן $C_{i+1} = G_i + C_i \cdot P_i = 0 + C_i = C_1$

כלומר $P_i = 1$ מבטא את העובדה שמסכם i מקדם (מעביר) את נשא כניסה C_i ליציאה C_{i+1}.

כעת נחשב את כל ה- C_i כפונקציות של C_0. (נשא כניסה לדרגה ראשונה אן לכל המערכת)

$i=0 \quad C_1=G_0 + P_0 \bullet C_0$

$$i = 1 \rightarrow C_2 = G_1 + P_1 \cdot C_1 = G_1 + P_1(G_0 + P_0 \cdot C_0)$$

$$C_2 = G_1 + G_0 P_1 + C_0 \cdot P_0 \cdot P_1$$

$$i = 2 \rightarrow C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + G_0 P_1 + C_0 \cdot P_0 \cdot P_1)$$

$$= G_2 + G_1 \cdot P_2 + G_0 \cdot P_1 \cdot P_2 + C_0 \cdot P_0 \cdot P_1 \cdot P_2$$

$$i = 3 \rightarrow C_4 = G_3 + P_3 \cdot C_3 = G_3 + P_3[G_2 + G_1 P_2 + G_0 P_1 P_2 + C_0 \cdot P_0 P_1 P_2]$$

$$= G_3 + G_2 \cdot P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 \cdot P_3 + C_0 \cdot P_0 P_1 \cdot P_2 \cdot P_3$$

רואים חוקיות הבאה :

$$** \; C_{i+1} = G_i + G_{i-1} \cdot P_i + G_{(i-2)} \cdot P_{i-1} \cdot P_i + G_{i-3} P_{(i-2)} \cdot P_{(i-1)} \cdot P_i + \ldots + G_0 \cdot P_1 \cdot P_2 \cdot \cdot P_i + C_0 \cdot P_0 \cdot P_1 \cdot P_2 \ldots P_i$$

מן הביטוי האחרון רואים שכל C_i תלויים בכניסות (ביטויים G_i P_i) ובנשא הכניסה C_0 ולכן ניתן לחשב את כל ה- C_i במקביל.

חישוב זמן של מציאת נשא של כל דרגה . (שיחושב במקביל עם שאר הדרגות).

הנחות:

נניח שזמן T זה זמן ההשהיה עבור כל רמת שערים וגם זמן השהיה עבור ה- F.A (FULL ADDER)

תהליך החישוב:

שלב א': P_i ו- G_1 מחושבים מקבילית, ולוקח רמת שערים אחת T.

שלב ב': כדי לחשב את C_{i+1} צריך עוד שתי רמות: רמה אחת לחשוב כל המכפלות בביטוי C_{i+1} ורמה שניה לחשוב סכום כל המכפלות (OR בין מכפלות) ולכן שלב זה לוקח זמן 2T.

שלב ג': חשוב תוצאה S_1 צריך שימוש ב- F.A עוד T ספייר (S_1 רק אחרי שחשבנו את C_1).

סה"כ: זמן של 4T לחשוב תוצאה ללא תלות במספר דרגות. לעומת מסכם גלי $\cong n \cdot T$ (n – מספר דרגות).

דוגמא :

תכנן מסכם למספרים בגודל 2 סיביות בשיטת נשא צפוי מראש.

מספרים $A = (A_1, A_0)$ $B = (B_1, B_0)$

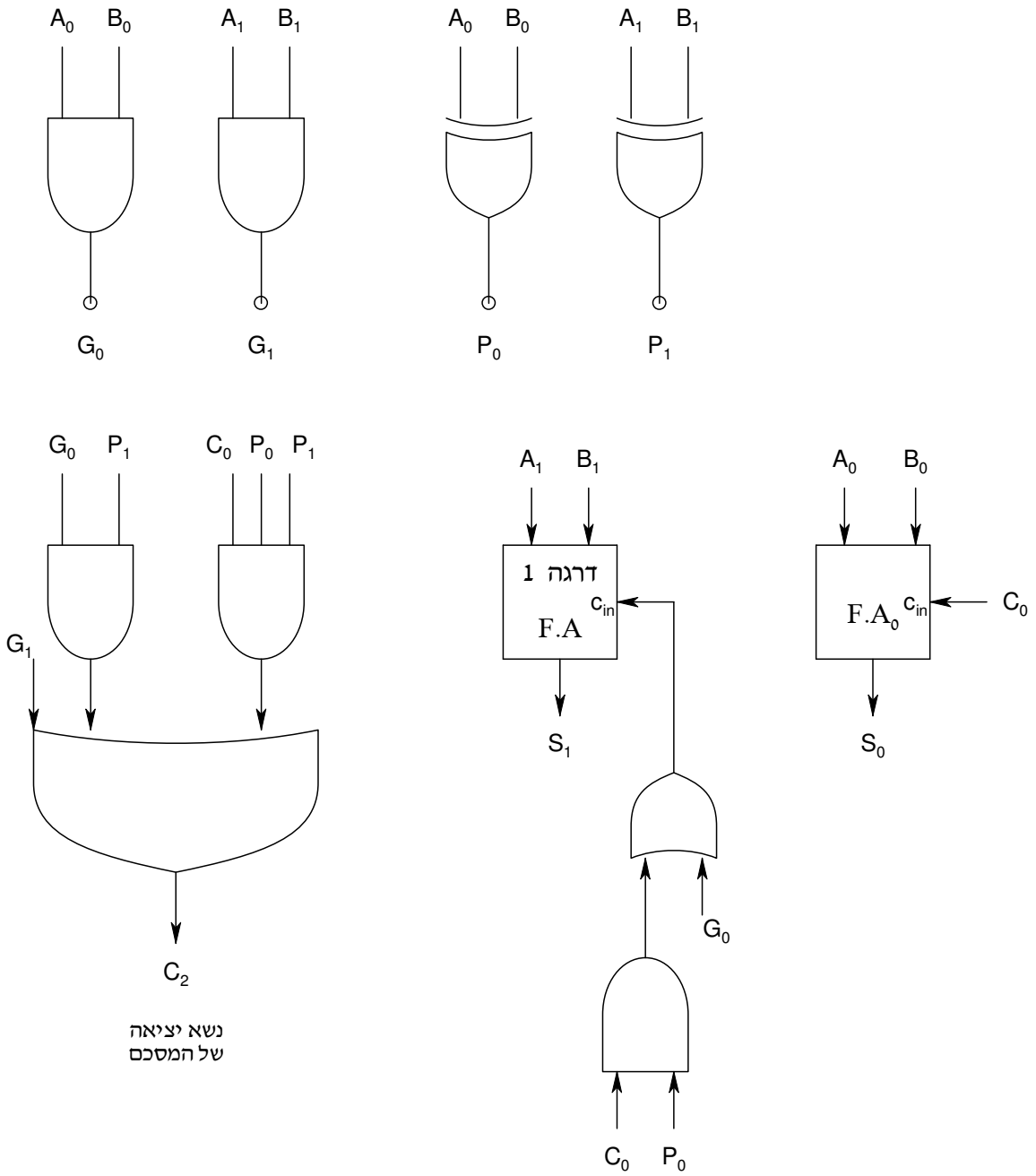
לפי משוואת $C_1 = G_0 + P_0 \cdot C_0$

$$C_2 = G_1 + G_0 \cdot P_1 + C_0 \cdot P_0 \cdot P_1$$

$$P_1 = (A_1 \oplus B_1) \qquad P_0 = (A_0 \oplus B_0)$$

$$G_1 = A_1 \cdot B_1 \qquad G_0 = A_0 \cdot B_0$$

מימוש המסכם בשיטת הנשא הצפוי מראש .

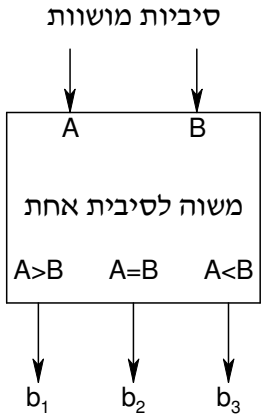


<u>משוה גודל COMPARATOR עבור מספרים חסרי סימן.</u>

ההשוואה בין שני מספרים קובעת האם מספר אחד גדול, קטן או שווה למספר האחר. "משוה" זו מערכת צירופית המשווה שני מספרים A ו-B וקובעת את היחס ביניהם. תוצאת ההשוואה מבוטאת באמצעות שלושה משתנים בינריים (סיבית אחת לכל משתנה) המציינים אם $A > B$ $A = B$ או $A < B$.

נתיחס בתחילה למקרה פשוט בו משווים שני מספרים שכל אחד הוא בעל סיבית אחת. משוה זה נקרא <u>משוה לסיבית אחת</u> (בצורה דומה משוה ל-n סיביות : משוה בין שני מספרים כל אחד בעל n סיביות).

תרשים מלבנים של המשוה :



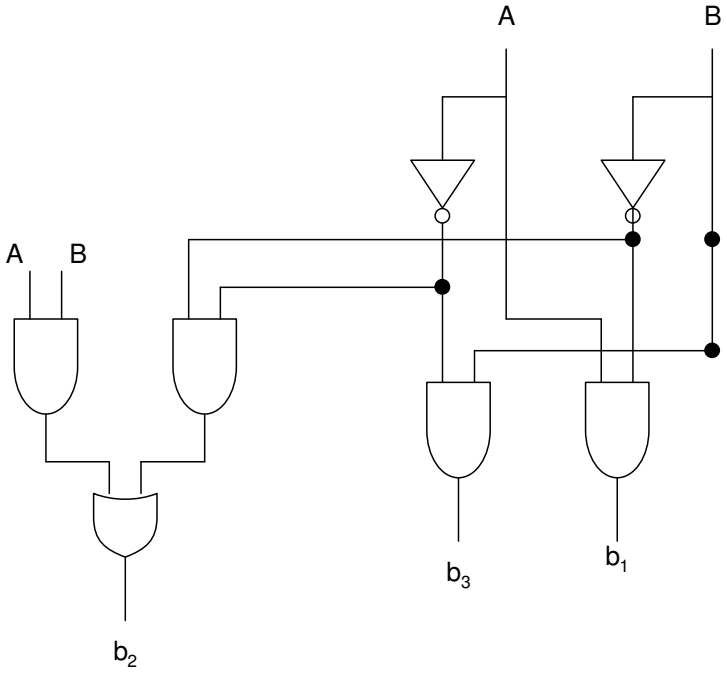
נרשום טבלת אמת של המשוה :

סיביות מושוות			מוצאים		
A	B		b_1	b_2	b_3
0	0		0	1	0
0	1		0	0	1
1	0		1	0	0
1	1		0	1	0

$$b_1 = A\overline{B}$$

$$b_2 = \overline{A} \cdot \overline{B} + A \cdot B = \overline{A} \oplus B = A \oplus \overline{B}$$

$$b_3 = \overline{A}B$$



ככל שהמשוה יותר גדול כן יותר מסורבל לממש, כי טבלת האמת גדלה . אם המשווה הוא של n ביט אז לטבלת האמת יש 2^{2n} שורות (כי הקלט הוא 2 מספרים כל אחד בגודל n סיביות).

בתרגול נלמד כיצד לתכנן משווים בשיטות יעילות יותר .

9.6.2 מערכות צירופיות לניתוב נתונים .

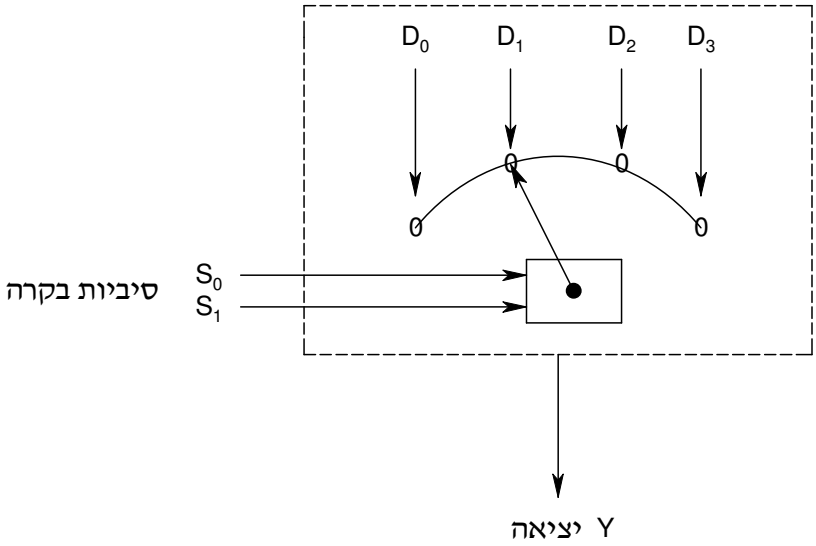
9.6.2.1 מרבב [שמות נוספים : בורר , SELECTOR , MUX] .

מרבב (MUX - MULTIPLEXER) הינו מערכת צירופית שבוררת כניסת נתונים אחת מבין מספר כניסות, ומעבירה ליציאת המערכת את כניסת הנתונים שנבחרה.

שימוש לדוגמא : מרכזית טלפונים בה נדרש להעביר מספר שיחות טלפון (לא באותו רגע) שמגיעות מקווים שונים, אל יעד אחד.

נתאר את פעולת המרבב בציור הבא :

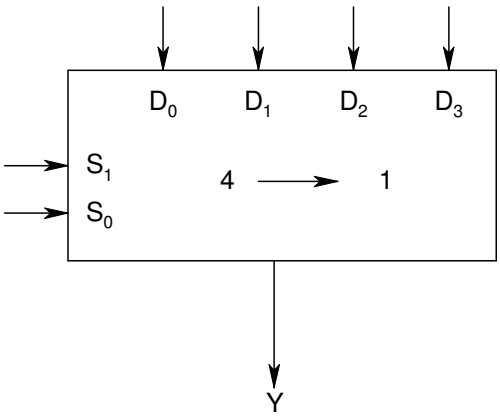
נתייחס למרבב $4 \rightarrow 1$ (4 כניסות ויציאה אחת)



למרבב זה 4 כניסות נתונים $D_0 ... D_3$ שתי כניסות בקרה ויציאה אחת.

פעולת מערכת: צירוף בינרי בכניסות הבקרה קובע את הכניסה הנבחרת (אחד מ- D_i) כניסה זו תועבר ליציאה (שים לב להתאמה, כיוון שרוצים לבחור 1 מ-4 כניסות מספיק 2 סיביות בקרה).

תרשים מלבנים :



<u>טבלת פעולת המערכת</u> .(זה לא טבלת אמת).

מבואות בקרה S_1	S_0	מוצא Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

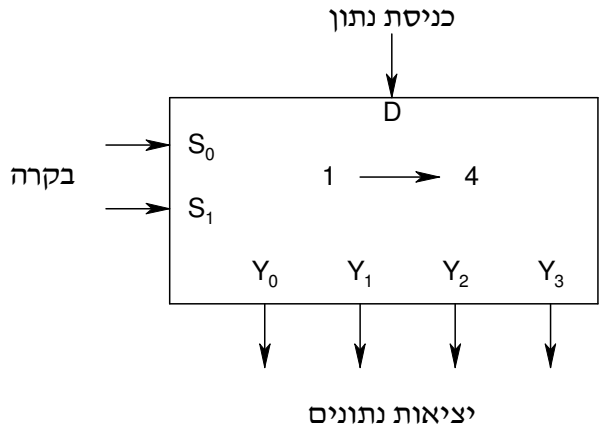
מן הטבלה רואים כי צירוף של כניסות הבקרה $S_1\ S_0 = 00$ $\Leftarrow$ כניסה 0 דהיינו D_0 עוברת ל- Y .

כאשר ערך בקרה 2 (כלומר 10) כניסה D_2 עוברת ל- Y וכן הלאה .

<u>הערה : בתרגול נלמד לממש פונקציות באמצעות מרבב.</u>

9.6.2.2 <u>מפלג (DE-MUX)</u> (פעולה הפוכה ממרבב) .

מפלג זו מערכת צירופית שמנתבת נתון אחד לאחד מתוך מספר מוצאים (המוצא יבחר לפי סיביות בקרה).
תרשים מלבנים :

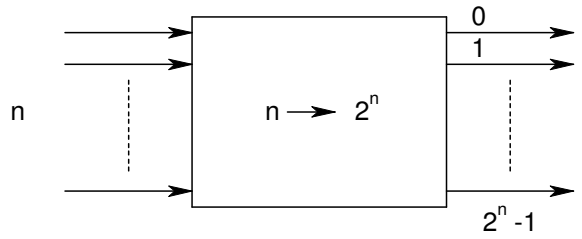


טבלת הפעולה של המערכת :

S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

9.6.2.3 מפענח .(DECODER)

מפענח הינו מערכת צירופית בעלת n כניסות (הנקראות בקרה) ו- 2^n יציאות.



המספרים שעל קוי היציאה מציינים את "מספר" היציאה.

פעולת המפענח : מתרגמים את מילת הכניסה בת n סיביות לערך עשרוני. נניח שהערך העשרוני הינו N.

<u>תוצאה</u> : יציאה מס' N "<u>נבחרת</u>" ושאר היציאות אינן נבחרות.

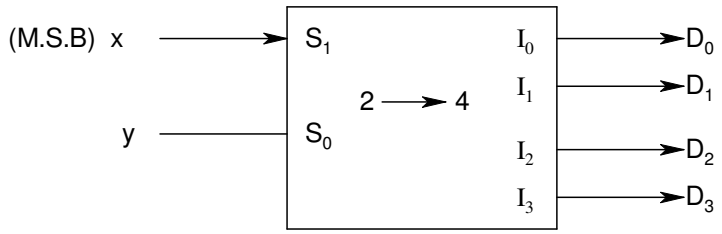
<u>קיימים</u> מפענחים שעבורם יציאה נבחרת ערכה : 1 (ויציאות שלא נבחרו ערכן 0) .

וקיימים מפענחים שעבורם יציאה נבחרת ערכה : 0

לכן עבור כל מפענח יש לציין את ערך היציאה הנבחרת.

<u>הערה</u> : יציאה נבחרת נקראת **יציאה פעילה**.

לדוגמא : מפענח $2 \to 4$ בעל <u>יציאה פעילה בערך 1.</u>



פעולת המפענח : כאשר $xy = 00$ (שזה ערך עשרוני 0) נבחרת יציאה D_0 [כלומר $D_0 = 1$ וכל שאר ה-D-ים הינם 0].

כללית אם ערך עשרוני שמתאים ל-xy הינו i אזי יציאה D_i נבחרת.

טבלת האמת של מפענח $2 \to 4$ (בעל יציאה פעילה 1) הינה :

x	y	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

משוואות היציאות :

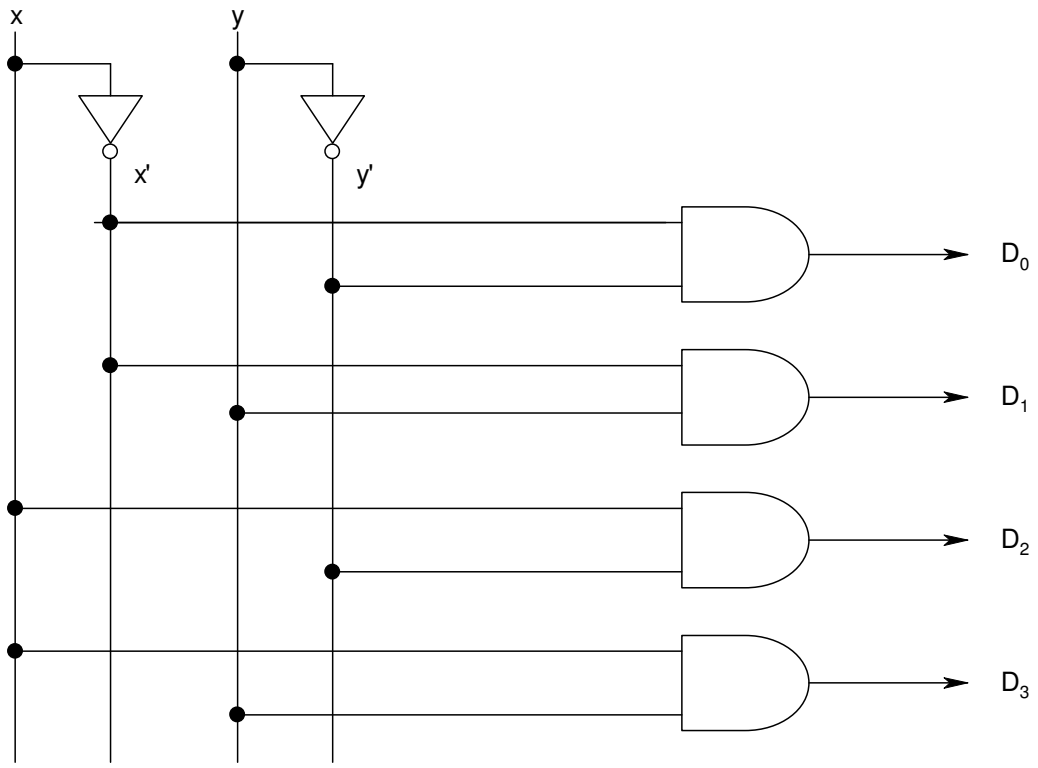
$D_0 = x' \cdot y'$

$D_1 = x' y$

$D_2 = xy'$

$D_3 = xy$

מימוש המפענח באמצעות שערים :



מימוש פונקציות באמצעות מפענח

טענה : באמצעות מפענח $n \to 2^n$ ניתן לממש כל פונקציה בת n משתנים.

מקרה א': נניח שמפענח הינו בעל יציאה פעילה בעלת ערך 1.

מימוש :1. את n המשתנים יש לחבר ל-n כניסות המפענח.

2. בצע פעולת OR על יציאות המפענח שמתאימות למכפלות סטנדרטיות (מינטרמים) שמהם מורכבת הפונקציה. (לשם כך יש לרשום את הפונקציה בצורת Σ).

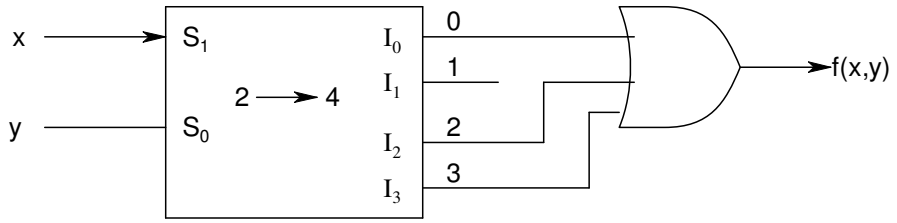
דוגמא : נתונה הפונקציה הבאה : $f(x, y) = x + x' y'$. נרשום את טבלת האמת של הפונקציה :

(M.S.B) x	y	f
0	0	1
0	1	0
1	0	1
1	1	1

מטבלת האמת נקבל: $f(x,y)=\sum 0,2,3$.

כיוון שהפונקציה בעלת שני משתנים צריך מפענח $2 \rightarrow 4$.

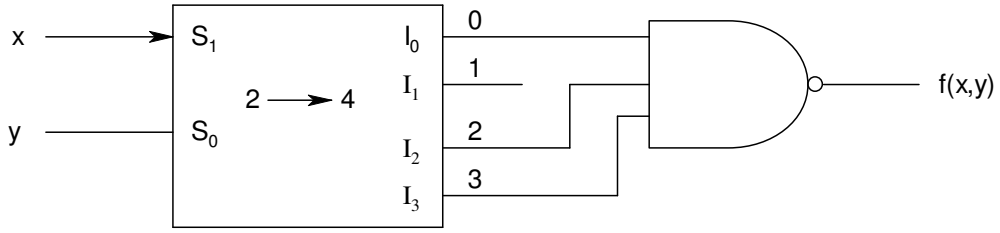
נשתמש במפענח בעל יציאה פעילה – 1.



בצענו פעולת OR על יציאות המפענח 0, 2, 3.

שים לב: מימוש זה מתאים בדיוק לטבלת האמת.

<u>מקרה ב':</u> נניח שמפענח הינו פעיל ב- 0. מימוש זהה לזה שבמקרה א', אלא שהפעם במקום שער OR נשתמש בשער NAND.



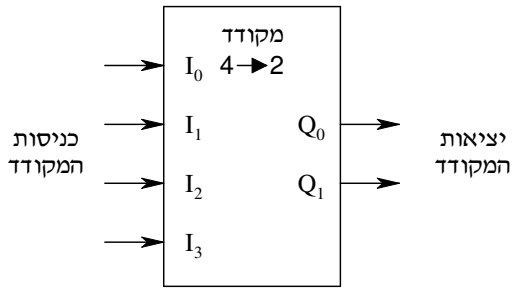
גם הפעם מימוש זה מתאים לטבלת האמת.

9.6.2.4 <u>מקודד עדיפות</u> (ENCODER)

המקודד מבצע פעולה הפוכה לזו של המפענח.

למקודד $2^n \rightarrow n$ יש n יציאות ו- 2^n כניסות. ברגע מסוים, כניסה אחת פעילה וכל יתר הכניסות אינן פעילות. ביציאות המקודד מופיע באותו רגע הקוד הבינארי המתאים לכניסה הפעילה.

באיור הבא מתואר למקודד בעל ארבע כניסות ושתי יציאות:



<u>נניח</u> כי כניסה פעילה ערכה 1 וכניסה לא פעילה ערכה – 0.

<u>דוגמא לשימוש במקודד הנ"ל</u>:

נתון שהכניסות של המקודד באיור הינן : $I_3I_2I_1I_0 = 0100$. מה ערך היציאות Q_1Q_0 ?

<u>פתרון</u>:

הכניסה הפעילה היא I_2 ושאר הכניסות אינן פעילות, לכן ביציאות המקודד יופיע צירוף שמתאים למספר 2 כלומר : $Q_1Q_0 = 10$.

עד כה תיארנו מצב שבו הותר לכניסה אחת בלבד – מבין כניסות המקודד – להיות פעילה.
מה יקרה אם נתיר לכמה כניסות להיות פעילות באותו זמן?

נניח, לדוגמא, שבאיור הנ"ל נתיר מצב שבו $I_1 = 1$ ובאותו זמן גם $I_3 = 1$. הרי לא יתכן שנקבל ביציאות את שני הקודים באותו זמן $Q_1Q_0 = 01$, $(Q_1Q_0 = 11)$. השאלה היא מה הקוד שיופיע ביציאות ?

כדי להתגבר על בעיה זו עלינו להגדיר סדר עדיפויות בין כניסות המקודד, כלומר לקבוע איזו מבין הכניסות היא העדיפה מכולם, איזו השנייה בעדיפותה וכך הלאה, עד לכניסה בעלת העדיפות הנמוכה ביותר.מקודד הפועל על פי סדר עדיפויות מוגדר של מבואותיו, נקרא מקודד עדיפות (priority encoder).

<u>עקרון הפעולה של מקודד עדיפות הינו</u>: אם כניסה אחת או יותר פעילות – יופיע ביציאות המקודד הצירוף הבינארי של הכניסה בעלת העדיפות הגבוהה ביותר מבין הכניסות הפעילות.

נתייחס למקודד עדיפות בעל ארבע כניסות ושתי יציאות דלעיל. נגדיר את הכניסה I_3 כבעלת העדיפות הגבוהה ביותר, ואת I_0 - כבעלת העדיפות הנמוכה ביותר. אם ברגע מסוים I_1 הוא המבוא הפעיל היחיד, יתקבל ביציאות המקודד $Q_1Q_0 = 01$. אם לעומת זאת, I_2 ו- I_1 פעילים יחדיו, הרי I_2 הוא העדיף ביניהם ותפוקת המקודד תהיה $Q_1Q_0 = 10$.