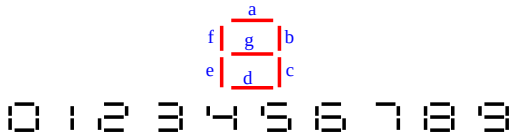
דוגמה: Seven -Segment - Decoder

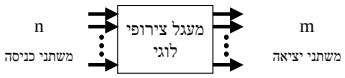


קלט: מספר בן 4 ביטים ב – BCD

פלט: 7 פונקציות בוליאניות כך שכל פונקציה הינה 1 אמ"מ ה- Segment המתאים צריך לדלוק.

- נבנה את טבלת האמת.
- נחשב את a,…,g ע"י מפות קרנו.
- נצמצם את המעגלים ע"י חיפוש שערים חוזרים.

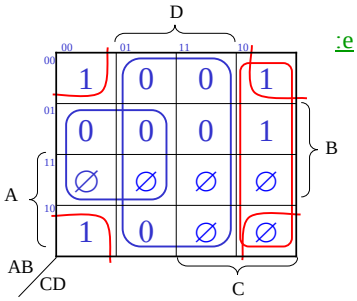
לוגיקה צרופית
Combinatorial Logic



שלבי התכנון של מעגל צירופי:

1. תאור הבעיה.
2. קביעת מספר משתני הכניסה הקיימים ומספר משתני היציאה הנדרשים.
3. התאמת סמלים למשתני הכניסה והיציאה.
4. בניית טבלת אמת המגדירה את היחסים הנדרשים בין הכניסות ליציאות.
5. פישוט הפונקציה הבוליאנית עבור כל יציאה.
6. "קיבוץ" ופישוט של הפונקציה הכוללת.
7. כתיבת הדיאגרמה הלוגית.

נחשב את e:

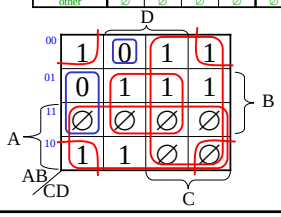


$e = D'B' + CD' = D'(B'+C)$

$e = (B'+C)D'$

טבלת האמת:

n	BCD IN				7 Seg Out						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
other	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅



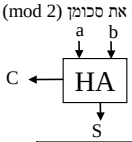
נחשב את a:

$a = B'D' + C + A + BD$

$a = (B'+D+C)(A+B+C+D')$

Half Adder – חצי מחבר

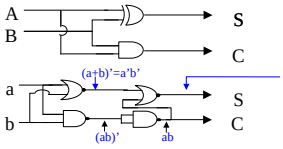
חצי מחבר: מקבל 2 סיביות ומחזיר את סכומן (mod 2) ואת הנשא.



a	b	s	c
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

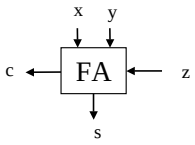
$S = a \oplus b$

$C = a \bullet b$



$(a'b' + ab)' =$
$=(a'b')'\bullet(a\bullet b)'$
$=(a+b)\bullet(a'+b')$
$= \cancel{aa'} + ab' + ba' + \cancel{bb'}$

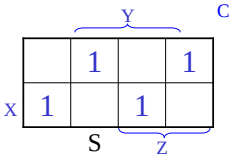
Full Adder – מחבר מלא

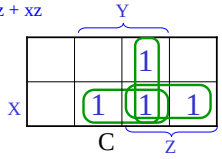


x	y	z	c	s
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1

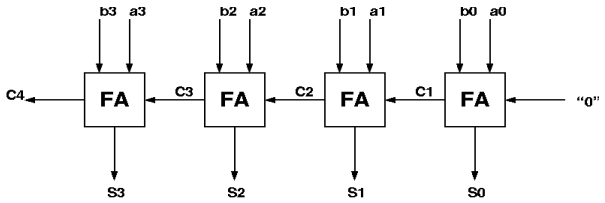
$S = x'y'z + x'yz' + xy'z' + xyz$

$C = xy + yz + xz$

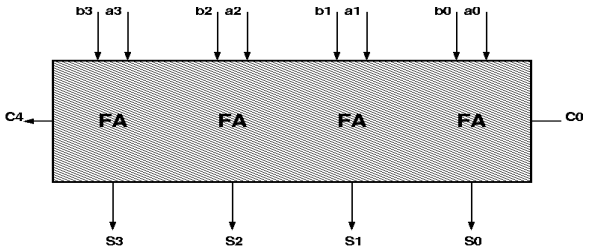




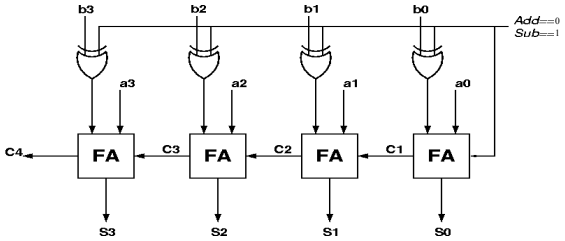
Ripple Carry Adder



4-Bit Adder

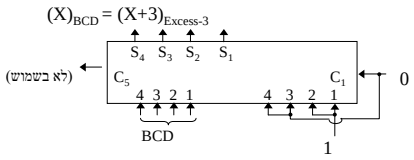


מחבר / מחסר

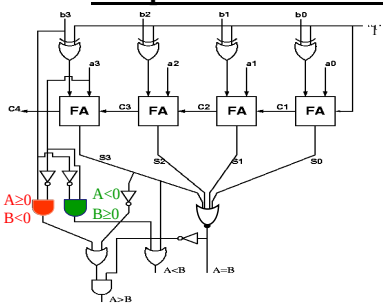


שמוש במחבר להמרת קודים

	BCD				Excess-3			
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

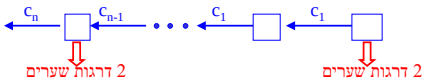


Comparator - משווה גודל



Carry Look Ahead – חיבור מהיר

- במחבר רגיל הנשא C_n מיוצר ע"י 2n רמות של שערים:

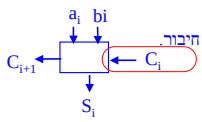


לכן זמן החיבור תלוי בגודל המספר.

פתרון אפשרי, אבל מוגבל מבחינה טכנולוגית: שימוש במעגלים מהירים יותר לצורך החיבור.

פתרון אלגוריתמי: משתמש בתכונות של חיבור.

נסתכל על שער i:



א. אם $a_i=1$ ו-$b_i=1$ אזי יהיה נשא ($c_{i+1}=1$) ללא תלות בערך c_i.

ב. אם $a_i=1$ או $b_i=1$ (אך לא שניהם) קיים "פוטנציאל" לקיום נשא: אם $c_i=1$ אזי $c_{i+1}=1$.

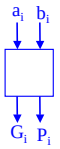
מחברים מהירים – המשך

נשא ודאי: <= $b_i=1, a_i=1$

נגדיר $G_i = a_i \cdot b_i$

נשא אפשרי: <= $a_i \oplus b_i=1$

נגדיר $P_i = a_i \oplus b_i$



נרשום עתה את S_i, C_{i+1} כפונקציה של C_i,G_i,P_i:

$S_i = P_i \oplus C_i$

$C_{i+1} = G_i+P_iC_i$

- P_i, G_i מחושבות מייד, ללא תלות בנשא מדרגה קודמת.
  נפתח עתה את c_i באופן רקורסיבי.

פיתוח רקורסיבי:

$C_1 = G_0 + P_0 \cdot \overset{0}{C_0} = G_0$

$C_2 = G_1 + P_1 \cdot C_1 = G_1 + P_1 \cdot G_0$

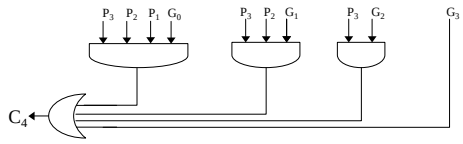
$C_3 = G_2 + P_2 \cdot C_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0) = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0$

$C_4 = G_3 + P_3 \cdot C_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0)$

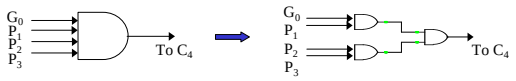
$= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$

$C_{i+1} = G_i + P_iG_{i-1} + P_iP_{i-1}G_{i-2} + \ldots + P_iP_{i-1} \ldots P_1G_0 = \sum_{k=0}^{i} \left(\prod_{j=k+1}^{i} P_j \right) G_k$

מימוש C_4:

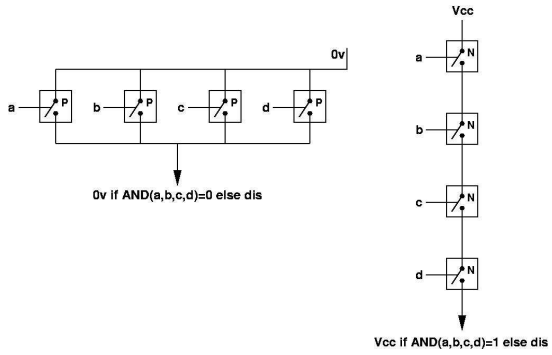


מימוש ריבוי כניסות ע"י שערים עם דרגת כניסה 2

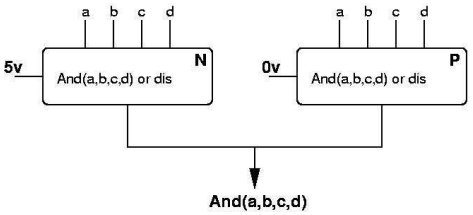


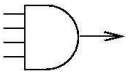
- מימוש בשערים רגילים מצמצם את זמן החישוב של הנשא (ולכן זמן החיבור הכולל) מ - $O(n)$ ל – $O(\log_2 n)$.
- אלטרנטיבה: ניתן לממש שערים עם דרגת כניסה גבוהה (אך קבועה) ע"י שימוש בתכונות של מתגים.

מימוש ריבוי כניסות ע"י מתגים חשמליים



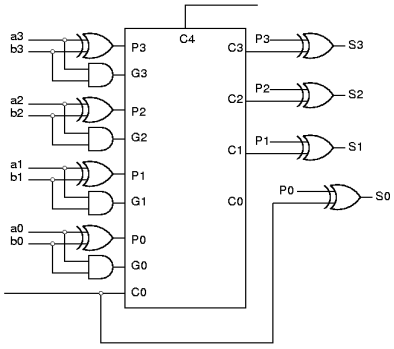
מימוש ריבוי כניסות ע"י מתגים חשמליים





ממשש את השער:

Carry Look Ahead Adder



מימוש מחבר BCD ע"י FAs

$A \in \{0,1,2,..,9\}$

$+ \; B \in \{0,1,2,..,9\}$

(S_1, S_0)

$\in(0,1) \quad \in\{0,...9\}$

• לרשותנו 4-bit FAs ושערי AND, OR

שלבים:

א. חבר $A+B \rightarrow T$ ע"י 4-bit FA.

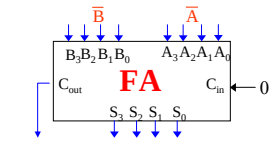
ב. מקרה א': $0 \le T \le 9$. סיימנו, התוצאה היא $(S_1,S_0)=(0,T)$.

ג. מקרה ב': $10 \le T \le 18$.

I. $S_1 \leftarrow 1$.

II. עלינו להפחית 10 מ – T כדי לקבל את התוצאה הנכונה.

שלב א':



$10 \le T \le 18$:

C_{out}	S_3	S_2	S_1	S_0	
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16
1	0	0	0	1	17
1	0	0	1	0	18

$\Longleftrightarrow 10 \le T$

$C_{out} = 1$ OR $S_3 = 1$ AND $(S_2=1$ OR $S_1=1)$

לכן ספרת העשרות היא $C_{out} + S_3 \cdot (S_2 + S_1)$

שלב ב':

אם $T \ge 10$ עלינו להפחית 10

$$\begin{array}{r} (C_{out}\ S_3\ S_2\ S_1\ S_0)_2 \\ - \quad (10)_{10} \\ \hline (X_3\ X_2\ X_1\ X_0)_2 \end{array}$$

מספר ≥ 8

ממוש מלא ב 2'sCo:

0	C_{out}	S_3	S_2	S_1	S_0
1	1	0	1	1	0
0	0	X_3	X_2	X_1	X_0

$\Longrightarrow$ "-10"

2s' Comp

• מספיקים 4 ביטים לייצג את התוצאה.

• שני ביטי MSB בשני המספרים אינם משפיעים על התוצאה ב – $(X_3X_2X_1X_0)$.

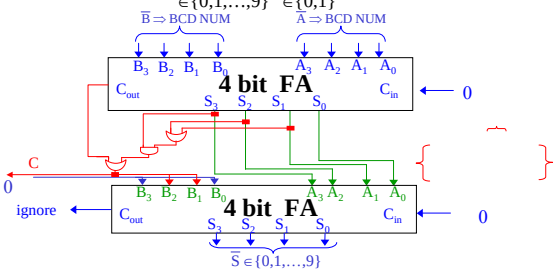
$\Longleftarrow$ ניתן לבצע החיסור ע"י החיבור של $(S_3S_2S_1S_0) + (0110)$!

ממוש מחבר BCD ע"י FAs

$A + B = (S,C)$

$\in\{0,1,\ldots,9\} \quad \in\{0,1\}$

$\bar{B} \Rightarrow$ BCD NUM $\quad \bar{A} \Rightarrow$ BCD NUM



כפל מספרים בינאריים - המשך

$$\begin{array}{r} a_2\ a_1\ a_0 \\ \times\ b_2\ b_1\ b_0 \\ \hline ?\ ?\ ?\ ?\ ? \end{array}$$

$$\begin{array}{r} 0 \quad 0 \quad b_0 \\ + \quad 0 \quad b_1 \quad 0 \\ b_2 \quad 0 \quad 0 \\ \hline b_2 \quad b_1 \quad b_0 \end{array}$$

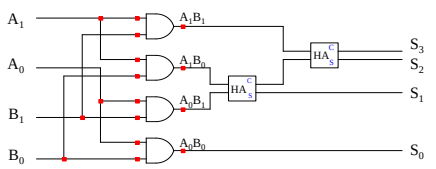
דוגמא: $(1011)_2 = (11)_{10}$

$$\begin{array}{rr} 1000 & 8 \\ +\ 0010 & 2 \\ +\ 0001 & 1 \end{array}$$

• הכפל יתבצע ע"י n חיבורים של הכפלים: $\begin{array}{r} a_{n-1}\ a_{n-2} \ldots \ldots a_0 \\ \times\ 0\ 0 \ldots 0\ b_1\ 0 \ldots 0 \end{array}$

ממוש כפל ע"י מחברים

$$\begin{array}{rrrr} & & B_1 & B_0 \\ & \times & A_1 & A_0 \\ \hline & 0 + & A_0B_1 & A_0B_0 \\ & A_1B_1 & A_1B_0 & 0 \\ \hline S_3 & S_2 & S_1 & S_0 \end{array}$$



לכן כפל מספרים ניתן לבצע ע"י חיבור:

$$
\begin{array}{lr}
 & \times\ a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0 \\
 & \times\ b_{n-1}\ b_{n-2}\ \dots\ b_1\ b_0 \\
\hline
b_0a_{n-1}b_0a_{n-2}\dots b_0a_0 & \times\ a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0\ \ b_0 \\
 & +\ \ 0\ \ 0\ \ \dots\ 0 \\
b_1a_{n-1}b_1a_{n-2}\dots b_1a_0 0 & \times\ a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0\ \ b_1\ \ 0 \\
 & +\ \ 0\ \ 0\ \ \dots\ b_1\ \ 0 \\
 & \vdots \\
b_{n-1}a_{n-1}\dots b_{n-1}a_0\dots 0 & a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0 \\
 & b_{n-1}\ \ 0\ \ \dots\ 0\ \ 0 \\
\hline
\end{array}
$$

$$
\begin{array}{lllll|lllll}
\dots & 0 & 0 & & & a_{n-1}b_0 & \dots & \dots & a_1b_0 & a_0b_0 & + \\
\dots & 0 & a_{n-1}b_1 & & & a_{n-2}b_1 & \dots & \dots & a_0b_1 & 0 & + \\
 & & & & & & & & \vdots & \vdots & \\
a_{n-1}b_{n-1} & \dots & a_1b_{n-1} & & & a_0b_{n-1} & \dots & \dots & 0 & 0 & +
\end{array}
$$

סה"כ יידרשו 2n ביטים לתוצאה.

הסבר: $(2^n-1)(2^n-1)=2^{2n}-2^{n+1}+1$ וזהו מספר גדול מדי לייצוג בעזרת 2n-1 ביטים.

קל להיווכח שהתוצאה של

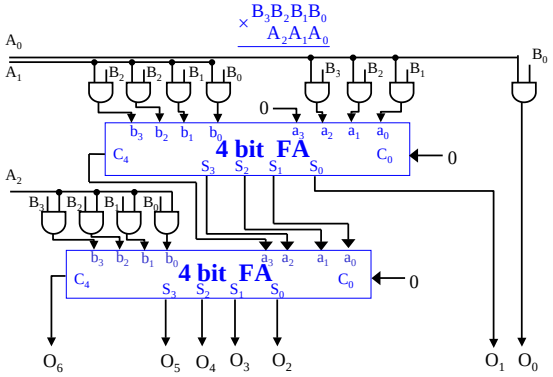
$$
\begin{array}{r}
a_{n-1}\ a_{n-2}\ \dots\ a_1 a_0 \\
\times\ 0\ 0\ \dots\ 0\ b_i\ 0\ \dots\ 0
\end{array}
$$

הינה

א. 0, אם $b_i=0$.

ב. $a_{n-1}a_{n-2}\dots a_1a_0\underbrace{00\dots0}_{i\ \text{אפסים}}$, אם $b_i = 1$.

דוגמא – כפל מספר 3B במספר 4B



שני המספרים אינם חייבים להיות בעלי אותו מספר ביטים.

לדוגמה:

$$
\begin{array}{r|c|c|c|c|c|c|c}
 & & & & 1 & 0 & 0 & 1 \\
\times & & & & 1 & 0 & 1 & \\
\hline
 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
+ & 0 & 0 & 0 & 0 & 0 & 0 & \\
+ & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\hline
 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
 & & 1 & 0 & 1 & 1 & 0 & 1
\end{array}
$$

באופן כללי אם נכפול מספר בעל n ביטים במספר בעל m ביטים נזדקק ל - n + m_ביטים לייצוג התוצאה.