**דף נוסחאות ספרתיות**

**המרת מספרים**

מבסיס 2 – ניתן ישר לבסיס 4,8,16.

מבסיס 3 – ישר לבסיס 9,27 (עפ"י חוקי חזקות) (כל שתי ספרות בבסיס 3 = סיפרה אחת בבסיס 9 )

לבסיס 10 – הכפלה בבסיס עם חזקה שהיא אינדקס המספר(מתחיל מ0)

מבסיס 10 – חלוקה של הבסיס ורשימת השארית. לוקחים את המספר מהסוף.

ייצוגים – יש לשים לב בכמה סיביות עובדים!

סיבית הסימן- היא הMSB. בחיסור וחיבור מתעלמים מסיבית הסימן מחשבים את הערכים בערך מוחלט ולוקחים את הסימן של הגדול.

בשיטת משלים לr-1 לוקחים את המספר הגדול ביותר בכמות הספרות שצריך(לדוגמא אם המספר השלילי שאני רוצה לייצג במשלים ל10 בבסיס 11 מכיל 5 ספרות אני אחסר אותו מהמספר AAAAA כאשר A מייצג את המספר העשירי בבסיס 11) \*\*את הOverflow יש להוסיף לתוצאה.

בשיטת משלים לr מחשבים משלים לאחד ומוסיפים אחד.\*\*מתעלמים מהOverflow.

עדיף לוודא תוצאה בבסיס 10.

ייצוג מספרים שליליים בשיטת משלים ל9 בבסיס 10 בשלוש סיביות– 0-499 חיובי 500-999 שלילי (500 מייצג את -0)

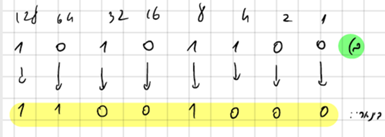
ייצוג מספרים שליליים בשיטת משלים ל10 בבסיס 10 בשלוש סיביות– 0-499 חיובי 500-999 שלילי (500 מייצג את -500 !)

המרות

BCD – המרת כל סיפרה מבסיס עשר לבסיס 2. לדוגמא 61 =0001 0110

בניית קוד גריי סטנדרטי –ע"י שיקוף והוספת אפסים לצד הישן ואחדות לצד החדש

**ממספר לקוד גרי סטנדרטי**

מעתיקים את הראשון, ואז עושים את פעולת הxor עם המספר משמאל

**מקוד גרי סטנדרטי למספר**

מעתיקים את הראשון אם 1 משמאל אי זוגי – מתהפך – זוגי נשאר כמו שהוא.->

קוד Hamming

P3 P2 M1 P1 M2 M3 M4 P3 מסתכל על מיקומים 1,3,5,7

7 6 5 4 3 2 1 P2 מסתכל על מיקומים 2,3,6,7 \*כמות אחדות זוגית

P1 מסתכל על מיקומים 4,5,6,7

מיקום אינדקס השגיאה מורכב מp1 p2 p3

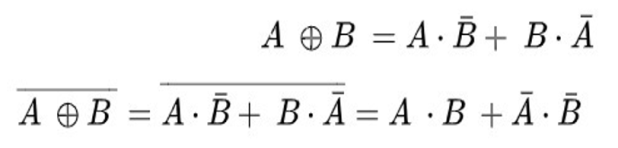
\*דורש יותר סיביות אבל יעיל יותר מבחינת הבחנה בטעויות של שתי ביטים ותיקון של שגיאה אחת.

סיבית הזוגיות תדע לזהות שגיאות אי זוגיות של ביטים בלבד.

זהויות חשובות :

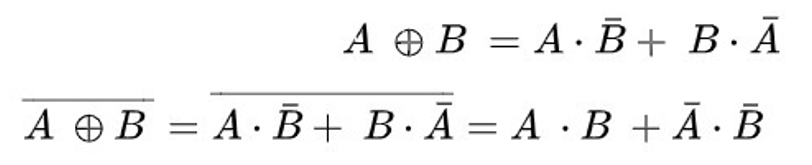
Xor –

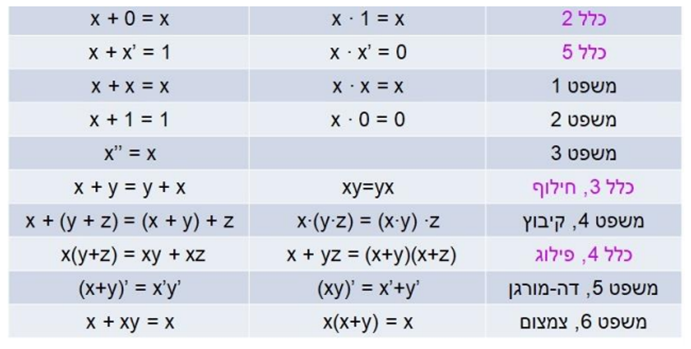
1. Xor(Xor(a,b),a)=b
2. Xor(x,0)=x

 ג (n>0) x^n=x.

ד.

Nxor-





דואליות נותן פונקציה משלימה במקרה שהמקורי היה כלל!

דרך פעולה - כל כפל הופך לחיבור וכל 1 הופך לאפס ולהפך.

* 1. F1 =

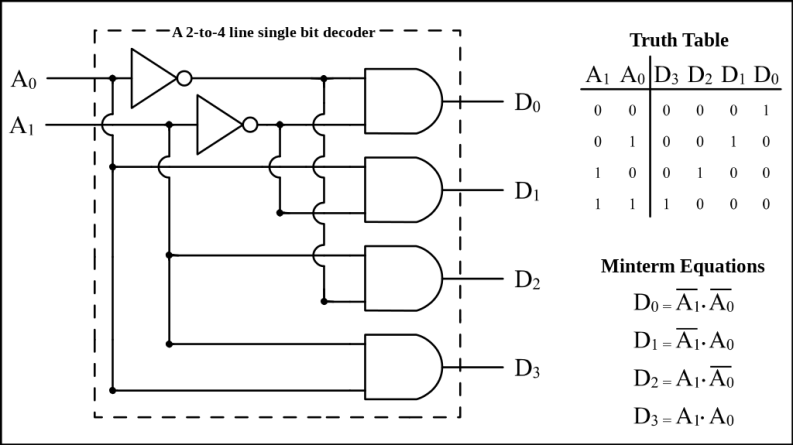
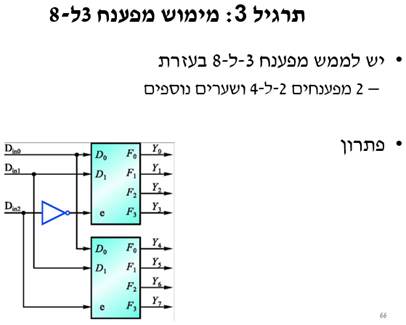
– ביטוי דואלי

דה מורגן כמו דואלי אך גם משנים את המשתנים עצמם לשלילה שלהם.

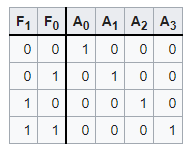
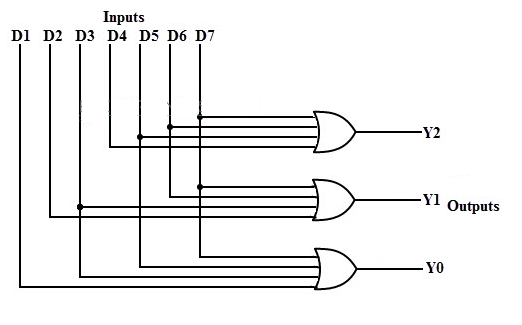
אפשר לבטא כל פונקציה בעזרת כפל ושלילה או חיבור ושלילה ע"י שימוש בדה מורגן. (נוסיף שתי שלילות)

רכיבים :

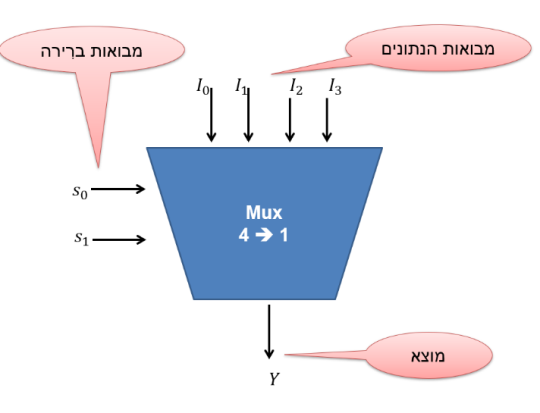
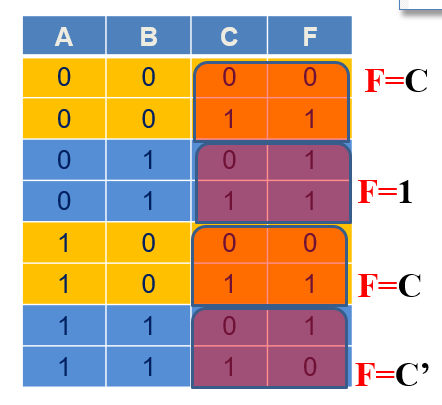
**מפענח** – מקבל n כניסות מוציא n^2 יציאות. נקרא גם מחולל מינטרמים כי הוא מוציא 1 במקום אחד והשאר אפסים. במקרה של צורך בצמצום- נכניס את כל המשתנים חוץ מהMSB לתוך שני מפענחים ונחבר בשער And את הMSB בצורה המתאימה **לכל הפלטים** ( אם לא ניתן להשתמש בו כ EN כמו בתמונה).



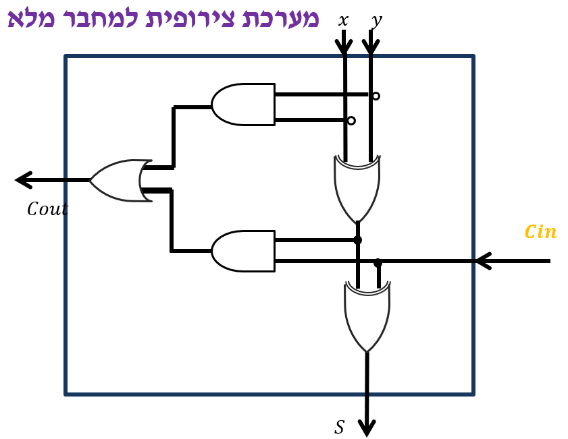
**מקודד**- בדיוק ההפך ממפענח. מקבל n כניסות מוציא logn יציאות. מפענח מוגדר לפעולה רק כשאחד מהקלטים הוא 1 והשאר אפסים. מוציא את הקומבינציה של המשתנים למיקום ה1 הזה.במקרה של צורך בצמצום- צריך לבדוק עם אחד הקלטים שנכנסים לרכיב נותנים 1 ושער הOr הזה משמש גם כmsb בתוצאה וגם כדי לבטל את התוצאה של המקודד השני כי הפלט שלו הוא DC.

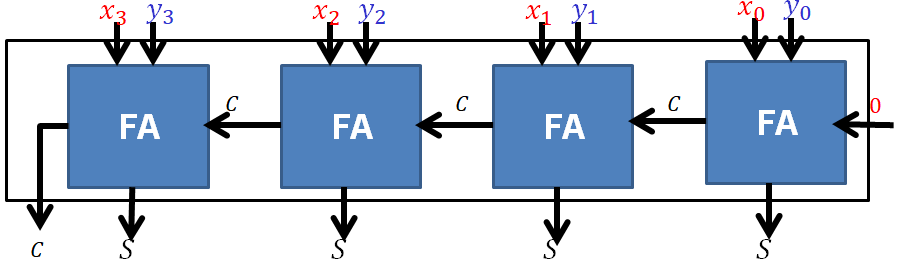


**מרבב**- מקבל את תוצאות הפונקציה כקלטים, את משתני הפונקציה כבוררים ומוציא רק תשובה אחת. במקרה של צורך בצמצום- נוציא את הLsb החוצה, ובמקרה הצורך נבטא את תוצאות הפונקציה באמצעותו. במקרה של צורך בצמצום נוסף, נוציא את הlsb אחריו, ונשתמש במקרה הצורך בשילוב שניהם (עם שערים) כדי לבטא את הפונקציה.

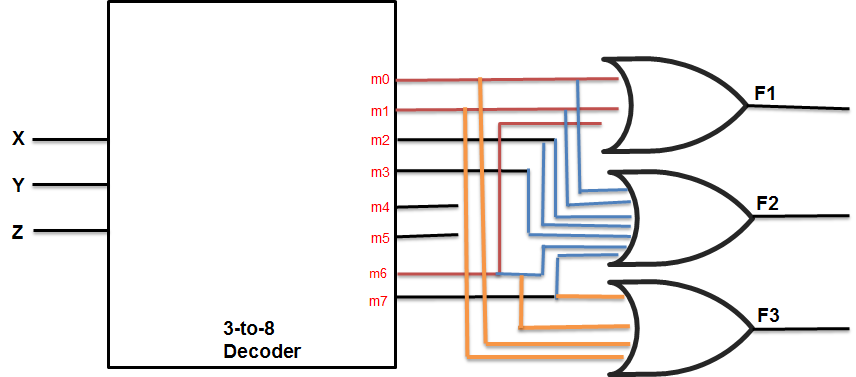
****

**מפלג**- ההפך ממרבב מקבל קלט אחד ומוציא הרבה פלטים. רק מפלט אחד(שיבחר ע"י מבואות הברירה) הוא יוציא את הקלט ומהשאר הוא מוציא רק אפסים.

**מחבר**-למחבר מלא יש שלוש כניסות:

* + שתי הסיביות המחוברות
  + הנשא המתקבל מחיבור שתי הסיביות הקודמות cin
* תוצאת הסכום כוללת שתי סיביות:
  + סיבית הסכום S
  + הנשא היוצא cout

דוגמה למימוש של מפענח בפונקציות שונות.

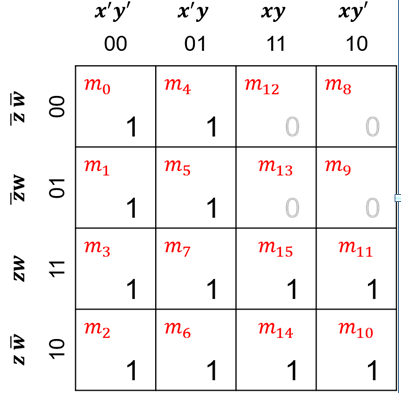


**משווה חלקי** – XSOR בין זוגות ביטים וAND לפלטים שלהם.

**משווה מלא-** A0\*B0' 🡺 A>B

A NXOR B 🡺 A=B

A0'\*B0 🡺 A<B

דרך פתירת תרגיל סטנדרטי – טבלת אמת , הצבת הערכים במפת קרנו, צמצום ויצירת מעגל.

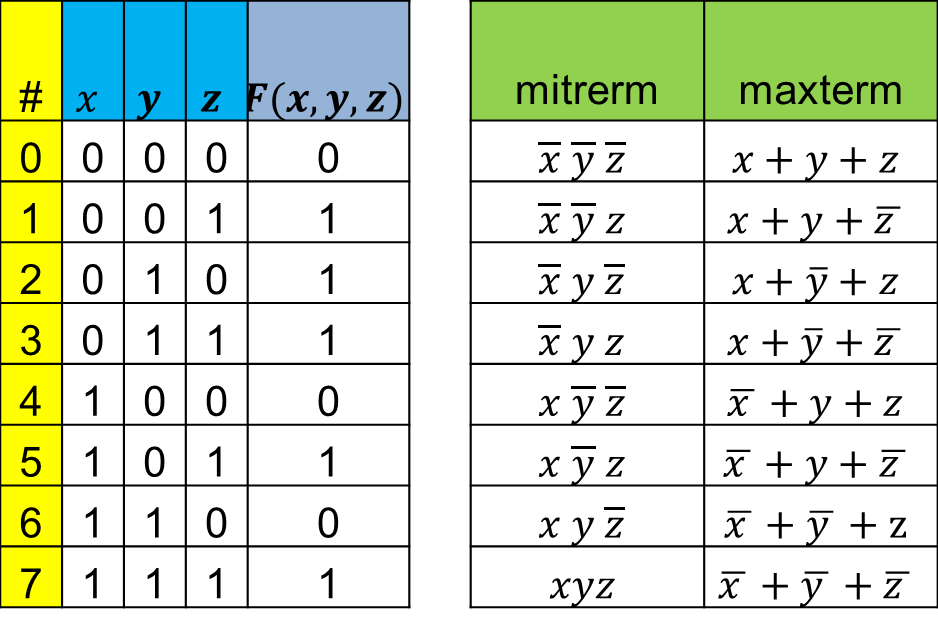
דוגמא למפת קרנו

מכפלת סכומים CNF : Conjunctive Normal Form

(+++)(’+’+)(+’+)

סכום מכפלות DNF : Disjunctive Normal Form

* + + ’ ’ + +’ …

*\*יש לשים לב שזה כמו מינטרם ומקסטרם אבל לא בדיוק. בביטויים פה לא חייבים להיות קומבינציה של כל המשתנים. במינטרם ומקסטרם כן.*

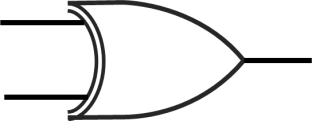
ליטרלים- כמות הופעת המשתנים בפונקציה.

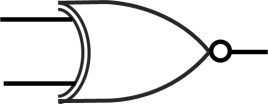
פונקציות זהות- פונקציות שמחזירות את אותו הערך עבור אותן ההצבות.

שערים:

שער OR מחזיר 0 רק כאשר שני המשתנים הם אפסים.(קל יותר לקבל אחד)

שער AND מחזיר 1 רק כאשר שני המשתנים הם אחד. (קל יותר לקבל אפס)

שער NOT מחזיר את השלילה של הליטרל

שער XSOR מחזיר 1 רק כשאר הביטים שונים.

שער NXSOR – ההפך מXSOR מחזיר 1 רק כשאר הביטים זהים.

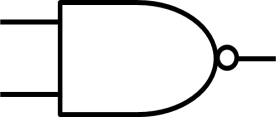
שער NOR – ההפך מOR מחזיר אחד רק כשאר שני המשתנים הם אפס.

מסומן גם כ

🡺 NOR(NOR, NOR) OR

NOR(x,x) =0 (x=1) 🡺 שלילה בNOR

NOR(x,x) =1 (x=0)

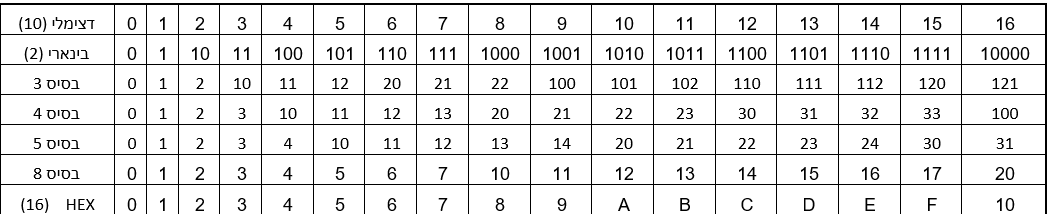
שער NAND – ההפך מAND מחזיר אפס רק כשאר שני המשתנים הם אחד. מסומן גם כ

🡺 NAND(NAND, NAND) AND

NAND(x,x) =0 (x=1) 🡺 שלילה בNAND

NAND(x,x) =1 (x=0)

טבלת המרות בבסיסים שונים :



כתיבת סכום מכפלות בקיצור (לא לשכוח לכתוב את המשתנים)

בסוגריים כותבים את מספרי המינטרמים

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 |  | 1 |
|  |  | 1 |  |
|  | 1 |  | 0 |
| 1 | 1 | 0 |  |

מכפלת סכומים :

בסוגריים כותבים את מספרי המקסטרמים

**בהצלחה !**