

מערכות הפעלה

3

דיסקים

יחידות מידה

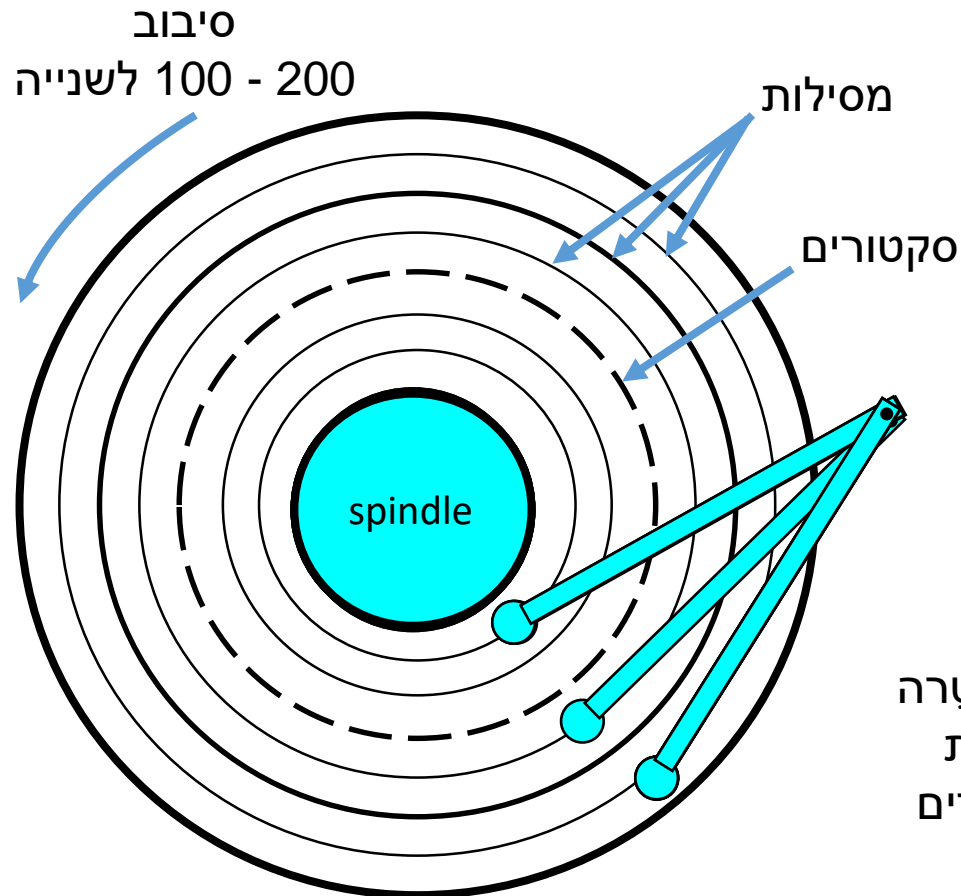
		בסיס דצימלי	בסיס בינרי		
Decimal term	Abbreviation	Value	Value	% Larger	
kilobyte	KB	10^3	2^{10}	2%	
megabyte	MB	10^6	2^{20}	5%	
gigabyte	GB	10^9	2^{30}	7%	
terabyte	TB	10^{12}	2^{40}	10%	

חלקי שניה		
מונח	קיצור	ערך
מילי שנייה	ms	$10^{-3} s$
מיקרו שניה	μs	$10^{-6} s$
ננו שניה	ns	$10^{-9} s$

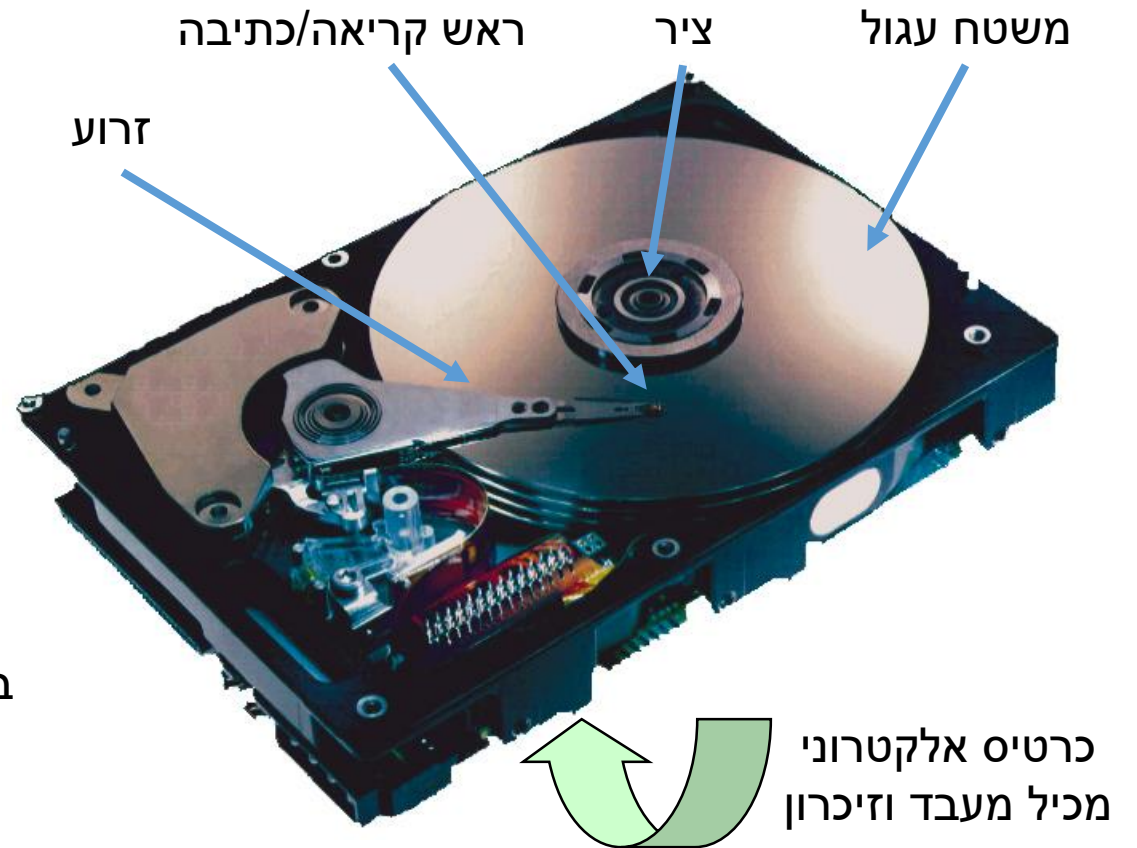
חזקות של 2	
חזקה	בתים (8 ביטים)
2^9	512 (0.5KB)
2^{10}	1024 (1KB)
2^{12}	4096 (4KB)

דיסק מכני

- הדיסק מכיל משטח עגול אחד או יותר מצופה בחומר מגנטי.
- כל משטח מחולק לטבעות הנקראות מסילות, כל מסילה מחולקת לסקטורים, בהם נשמר המידע.
- סיבוב הדיסק ותזוזת הזרוע מאפשרות למקם את ראש הקריאה/כתיבה מעל כל סקטור.
- סקטור הוא יחידת הזיכרון הקטנה ביותר שאפשר לכתוב או לקרוא, גודלו 512 בתים או יותר.



בדיסק בנפח 1 טרה
כמיליון מסילות
כאלפיים סקטורים

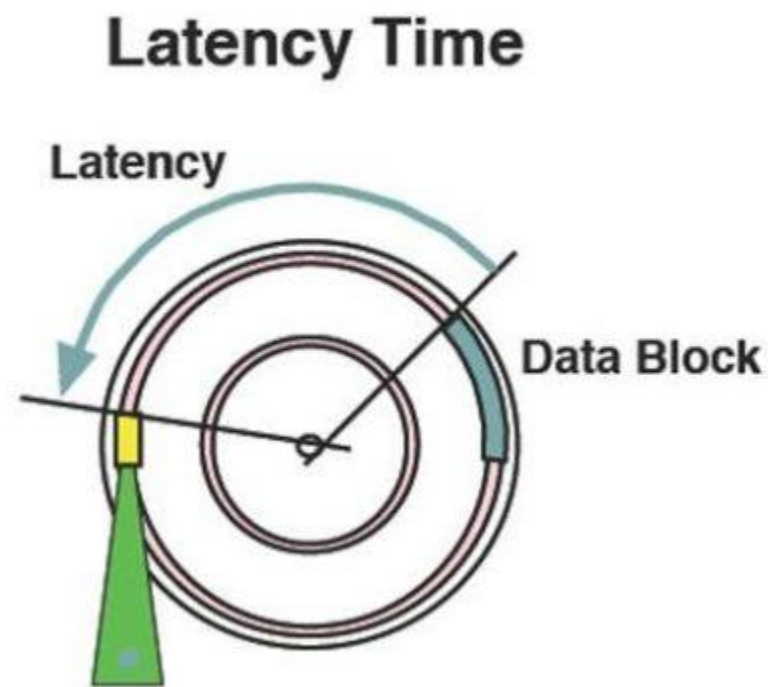
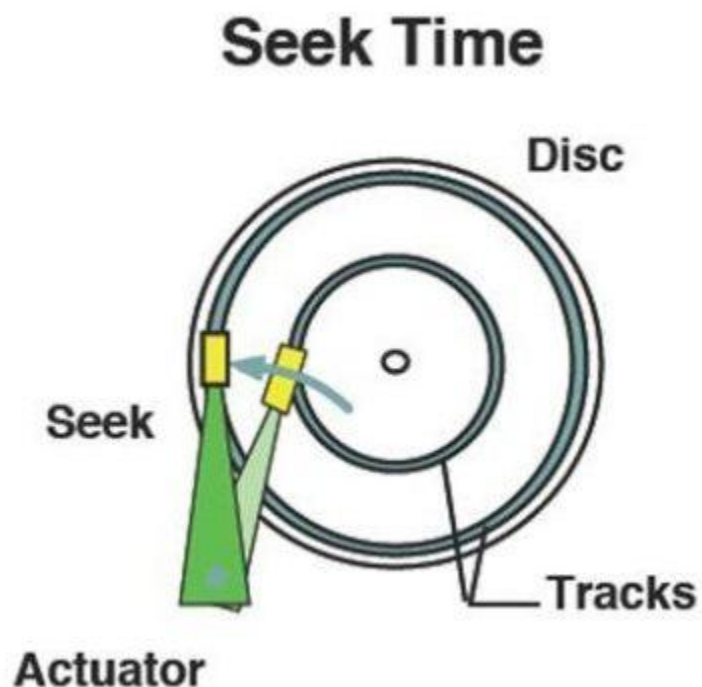


זמן הגישה לדיסק מכני

- הזמן הממוצע לקריאה או כתיבת סקטור בודד (Access) הוא סכום של שלושה זמנים :

$$\text{Access} = \text{Seek} + \text{Latency} + \text{Transfer}$$

1. **Seek** - הזמן שלוקח למקם את הראש מעל המסילה (כמה אלפיות השנייה).
2. **Latency** - זמן ההמתנה לתחילת הסקטור שיגיע לראש (כמה אלפיות השנייה).
3. **Transfer** - זמן המעבר של הסקטור מתחת לראש (קטן מאלפית השנייה).



זמן הגישה לדיסק מכני

- זמן הגישה נקבע בעיקר על ידי שני הזמנים הראשונים.
- אם הבלוקים של המידע מפוזרים על פני הדיסק במסילות שונות, יידרשו שני הזמנים הראשונים לכל בלוק.
- אם הבלוקים של המידע רציפים בדיסק, יידרשו שני הזמנים הראשונים רק לבלוק הראשון, ליתר הבלוקים יידרש רק הזמן הקצר של מעבר על הסקטורים.
- לכן מערכת ההפעלה קוראת וכותבת לדיסק ביחידות של כמה בלוקים רצופים.
- בדיסקים מודרניים מערכת ההפעלה קוראת וכותבת למערך של בלוקים לוגיים (LBA).
- הדיסק עצמו מחשב את המסילה, המשטח והסקטור של כל בלוק (CHS).
- זמני גישה לזיכרון ולדיסק:

Memory technology	Typical access time
SRAM semiconductor memory	0.5–2.5 ns
DRAM semiconductor memory	50–70 ns
Flash semiconductor memory	5,000–50,000 ns
Magnetic disk	5,000,000–20,000,000 ns

חישוב זמן הגישה לדיסק

לדיסק יש את הנתונים הבאים:

Sector size: 512 bytes

Surfaces: 4

Rotational rate: 10,000 RPM

seek: 5 ms

Number of sectors/track: 1,000

כמה זמן ייקח לקרוא קובץ בגודל 2000×512 (בקרוב 1MB) מתחילתו ועד סופו, כאשר הבלוקים מסודרים בצורה הגרועה ביותר ובצורה הטובה ביותר.

$$\text{full_rotation} = 1/10,000 * 60 * 1000 = 6 \text{ ms}$$

$$\text{latency} = \text{full_rotation} * 0.5 = 3 \text{ ms}$$

$$\text{transfer} = \text{full_rotation} / 1000 \text{ sectors} = 0.006 \text{ ms}$$

הקובץ מכיל 2000 בלוקים, בסידור הגרוע:

$$(\text{seek} + \text{latency} + \text{transfer}) * 2,000 =$$

$$(5 + 3 + 0.006) * 2,000 = 16,012 \text{ ms}$$

בסידור הטוב:

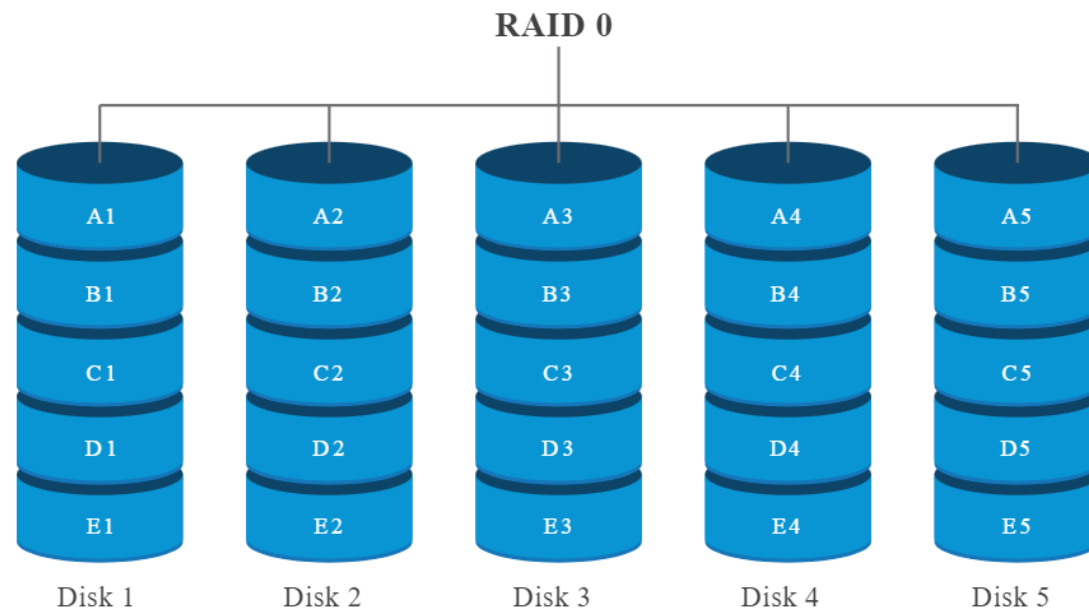
$$\text{seek} + \text{latency} + 2 * \text{full_rotation} = 5 + 3 + 12 = 20 \text{ ms}$$

RAID - Redundant Arrays of Inexpensive Disks

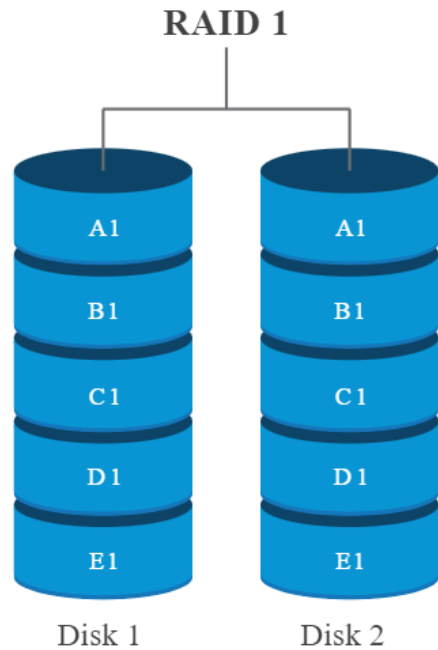
- ראינו שאפשר לחלק דיסק פיזי לדיסקים לוגיים, כעת נראה איך אפשר לחבר כמה דיסקים פיזיים לדיסק לוגי אחד (volume).
- היתרונות בחיבור דיסקים:
 - שיפור ביצועים - אפשר לגשת לדיסקים במקביל.
 - אמינות – אם יש תקלה בדיסק, אפשר לשחזר את הנתונים באמצעות מידע יתיר (redundant) שנשמר ביתר הדיסקים.
- RAID היא שיטה לחיבור כמה דיסקים לדיסק אחד.
- שיטת RAID כוללת כמה אפשרויות לחיבור דיסקים:
 - RAID 0, ... , RAID 6
 - להלן נראה את האפשרויות הנפוצות.

RAID 0 - Striping (פסים)

- ראינו שמערכת ההפעלה רואה את הדיסק כמערך של בלוקים.
- בשיטת RAID 0 מחלקים את הדיסקים לפסים, בלוק 0 הוא בדיסק הראשון, בלוק 1 בדיסק הבא וכן הלאה.
- ביצועים: אפשר לגשת לדיסקים במקביל, נותן את הביצועים הטובים ביותר של RAID.
- לדוגמה, אפשר לקרוא או לכתוב קובץ בגודל שני בלוקים בשני דיסקים בו זמנית.
- נפח: צרוף הדיסקים לא מפחית מהנפח הכולל של הדיסקים.
- אמינות: אין יתירות, אם יש תקלה בדיסק אחד, המידע שעל כל הדיסקים משתבש, יותר גרוע מדיסקים ללא RAID.



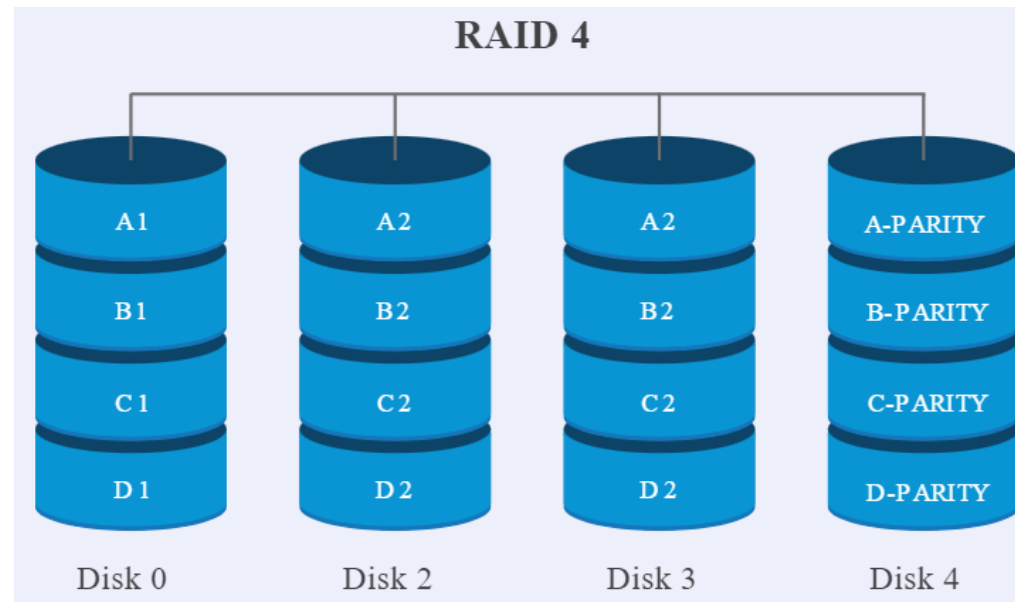
RAID 1 – Mirroring (שכפול)



- בשיטת RAID 1 כל הנתונים נכתבים לשני דיסקים (ראי).
- ביצועים: קריאה היא יותר מהירה כי אפשר לקרוא משני הדיסקים במקביל.
- נפח: בשיטה זו אפשר לנצל רק חצי מנפח הדיסקים.
- אמינות: אם יש תקלה בדיסק, אפשר לקרוא מהדיסק השני.

RAID 4 – Parity (זוגיות)

- בשיטת RAID 4 מחלקים את הדיסקים לפסים.
- בדיסק האחרון נשמר מידע (זוגיות) שמאפשר שחזור הנתונים במקרה של תקלה באחד מהדיסקים האחרים.
- ביצועים: משפר את הביצועים, אפשר לגשת לדיסקים במקביל.
- נפח: דיסק אחד משמש לזוגיות ולכן הנפח לשמירת נתונים הוא כמספר הדיסקים פחות אחד.
- אמינות: אם יש תקלה בדיסק אחד, אפשר לשחזר את הנתונים מיתר הדיסקים.



זוגיות (Parity)

- זוגיות היא פונקציה שמחשבים על הבלוקים של הנתונים של כל פס ושומרים את התוצאה בבלוק שבדיסק האחרון.
- הפונקציה שמחשבים היא XOR על הביט הראשון של כל הבלוקים של הפס ושמירת התוצאה בביט הראשון של בלוק הזוגיות, XOR על הביט השני ושמירת התוצאה בביט השני של בלוק הזוגיות, וכן הלאה עבור כל ביט בבלוק.
- התוצאה של XOR היא 0 אם מספר הביטים שהם 1 הוא זוגי, התוצאה היא 1 אם מספר הביטים שהם 1 הוא אי-זוגי.
- יוצא שמספר הביטים שהם 1 בכל הבלוקים כולל בבלוק הזוגיות הוא זוגי עבור כל ביט.
- בדוגמה הבאה מספר הדיסקים הוא חמש ובכל בלוק ארבעה ביטים:

Block0	Block1	Block2	Block3	Parity
00	10	11	10	11
10	01	00	01	10

חישוב הזוגיות

- בכל כתיבה של בלוק נתונים יש צורך לעדכן את בלוק הזוגיות.
- 1. אפשרות אחת היא לקרוא את כל הבלוקים של הפס, לחשב את הזוגיות ולכתוב בבלוק הזוגיות.
- 2. אפשרות יותר יעילה היא להשוות את הביטים (בהתאמה) של הבלוק שרוצים לכתוב לתוכן הקודם שלו.
- אם הביטים שווים אזי ביט הנתונים לא השתנה ולכן אין צורך לשנות את ביט הזוגיות.
- אם הביטים שונים צריך להפוך את ביט הזוגיות.
- אפשר לבצע את החישוב באופן הבא:

זהויות עבור XOR:

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

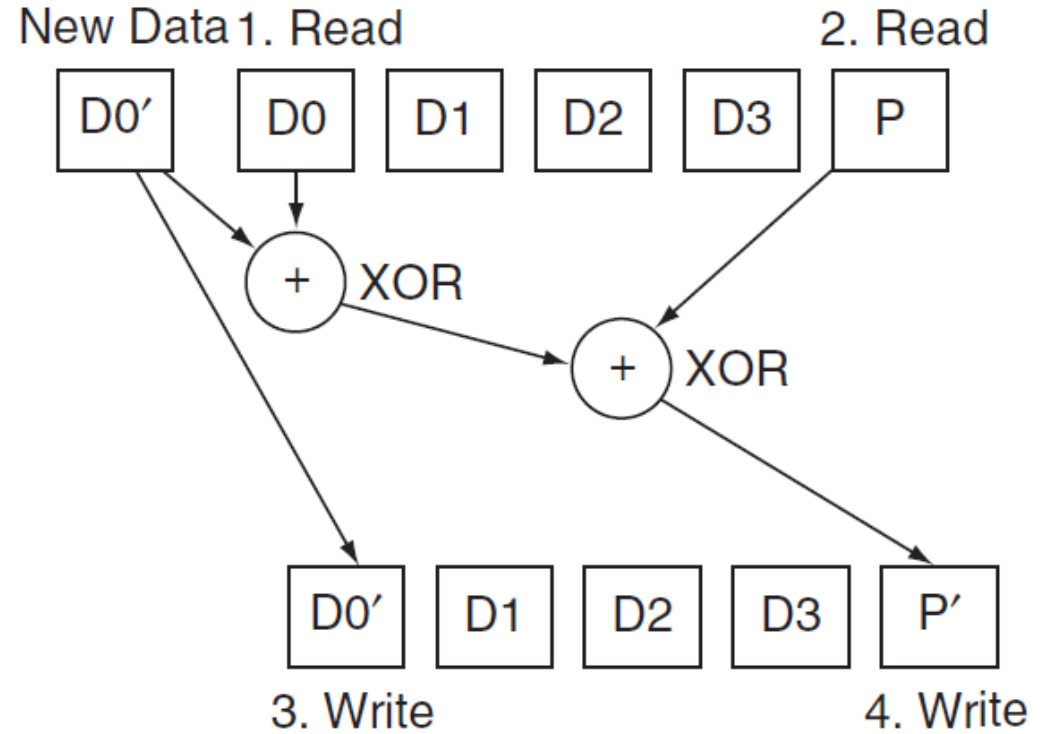
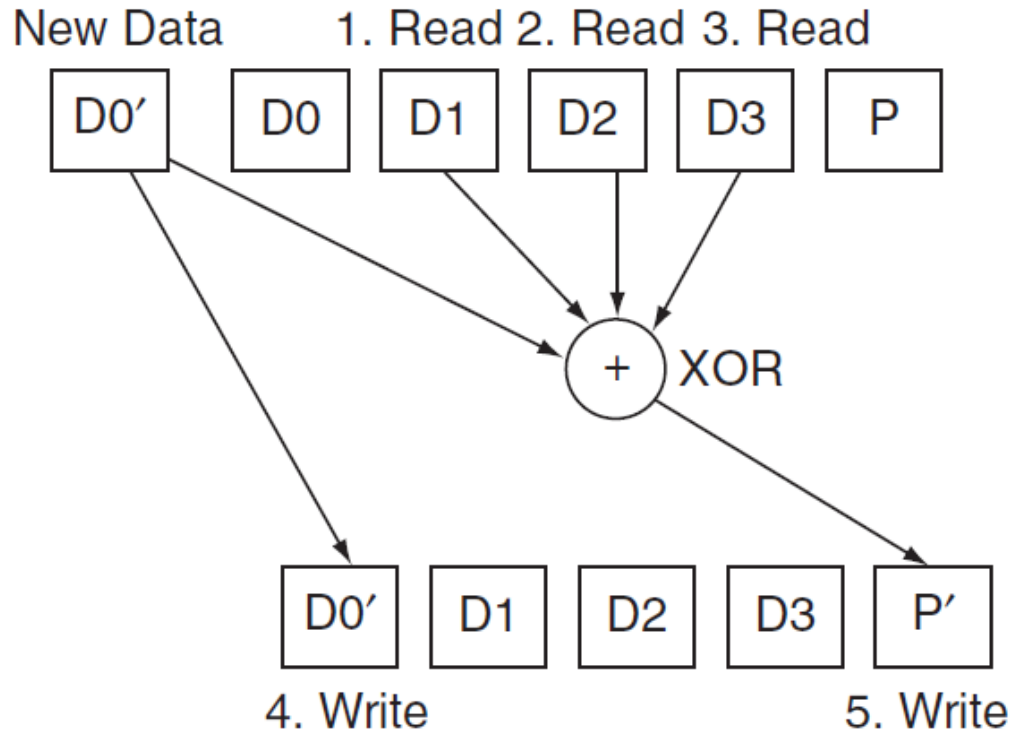
$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$P_{new} = (C_{old} \oplus C_{new}) \oplus P_{old}$$

חישוב הזוגיות

- הציור הבא מדגים את שתי האפשרויות:



שחזור נתונים

- אם ארעה תקלה בדיסק, כדי לשחזר את הנתונים בדיסק חדש, נבצע עבור כל בלוק, XOR של כל הדיסקים הנותרים כולל דיסק הזוגיות.
- הסבר:
 - ראינו שבכל הדיסקים כולל דיסק הזוגיות מספר הביטים שהם 1 הוא זוגי.
 - פעולת XOR על הדיסקים הנותרים תיתן 1 אם מספר הביטים הוא אי זוגי, ואז נכתוב בדיסק החדש 1 ונקבל שוב מספר זוגי.
 - פעולת XOR על הדיסקים הנותרים תיתן 0 אם מספר הביטים הוא זוגי, ואז נכתוב 0 בדיסק החדש כי מספר הביטים הוא כבר זוגי.

זוגיות בסבב על כל הדיסקים - RAID 5 – Rotating Parity

- בשיטת RAID 4, בכל כתיבה של בלוק לדיסק כלשהו צריך לעדכן את הבלוק שבדיסק הזוגיות.
- זה הופך את דיסק הזוגיות לצוואר בקבוק בכתיבות לדיסק.
- בשיטת RAID 5, כדי לשפר את הביצועים, הבלוקים של הזוגיות מפוזרים בסבב על כל הדיסקים.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
*4	5	6	7	+P1
8	9	10	11	P2
12	*13	14	15	+P3

