

## מערכות זיכרון – Sequential Logic



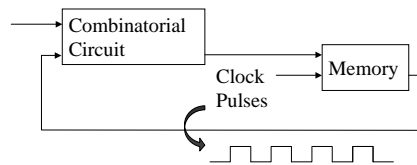
- עד כה טיפלנו במערכות צירופיות שהגן חסרות זיכרון.
- כל מחשב מכיל זיכרון המשמש לשמירת נתונים ולביצוע תוכניות.
- הזיכרון שנתייחס אליו בקורס הנוכחי מושג ע"י שימוש במשוב (feedback) של יחידות לוגיות.
- מעגל עם משוב איננו מעגל צירופי, כי מעגל צירופי הוא גרף חסר מעגלים.

## Synchronous Sequential Circuits

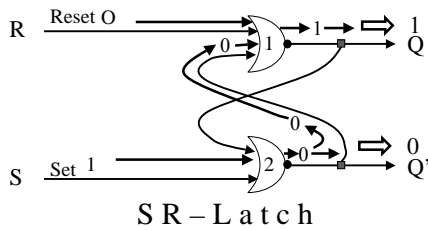
**יציב** פעולת המערכת מתבצעת ב"פיקודו" של שעון. אנו דוגמים את מצב המערכת רק ברגעי זמן דיסקרטיים, המוגדרים ע"י השעון.

## Asynchronous Sequential Circuits

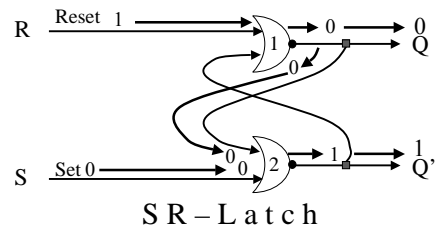
**מהיר** פעולות המערכת תלויות רק בסדר של שינוי הכניסות. ניתן לדגום את מצב המערכת בכל רגע נתון. בקורס הנוכחי נדון כמעט אך ורק במערכות סינכרוניות.



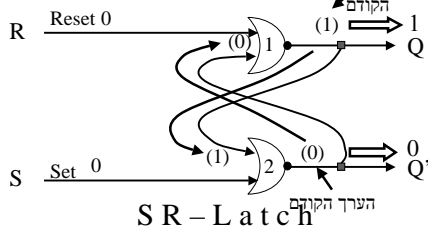
## LATCH



## LATCH

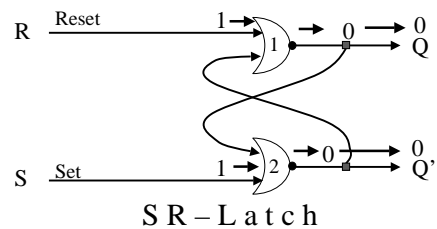


## LATCH



- נבנה שניתנה פקודת Set ,
- ז"א  $S=1, R=0$  ונתה אנו מאפסים את הכניסות  $(S=0, R=0)$ .
- ערכי היציאה  $Q, Q'$  "זוכרים" את פקודת ה-Set.
- מן הסימטריה ברור שבאותו אופן יזכרו את ה-Reset.

## LATCH



- כאשר  $S=R=1$ , נקבל  $Q=Q'=0$
- אם כעת מאפסים את הכניסות  $(S=0, R=0)$ , הערך של  $Q$  ו  $Q'$  תלוי באיזה קו ישתנה ראשון ← RACE Condition
- המצב הזה, מבחינתנו, הוא איננו חוקי.

דיאגרמת זמנים:

טבלת המצבים:

S	R	Q	Q'
0	1	0	1
0	0	0	1
1	0	1	0
0	0	1	0
1	1	0	0

Reset State

Set State

Illegal

- Latch איננו פונקציה בוליאנית, כי ערכי היציאה תלויים בפעולת Set או Reset האחרונה.
- עבור כניסות (0,0) ערך היציאה נשמר קבוע כל זמן שיש מתח.

SR Latch with NAND

S	R	Q	Q'
0	1	0	1
1	1	0	1
1	0	1	0
1	1	1	0
0	0	1	1

Reset State

Set State

Illegal

SR Latch מבוקר שעות

C	S	R	Next Q
0	∅	∅	No change
1	1	0	Q=1
1	0	1	Q=0
1	1	1	Illegal
1	0	0	No change

D (data) Latch

D (data) Latch

C	D	Next State of Q
0	∅	No change
1	0	Q=0 (Reset)
1	1	Q=1 (Set)

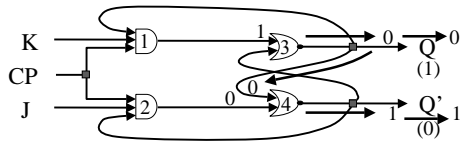
- D-Latch הנה יחידה שאוגרת ביט יחיד.
- זוהי אבן הבניין הבסיסית של אוגרים (Registers).

JK Latch

זוהו שכלול של SR Latch.

- הרכיב פעיל אך ורק כאשר CP=1, ואז:
- כאשר J=1, K=0, המעגל יתייצב על Q=1, Q'=0.
- כאשר J=0, K=1, נקבל Q=0, Q'=1.
- מה קורה כאשר J=1, K=1?

### JK Latch



מה קורה כאשר  $J=1, K=1$ ?  
מצב Q מתהפך.  
אם לאורך זמן  $J=K=1$ , מצב Q יתהפך שוב ושוב.

### טבלת אמת עבור JK-Latch

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

דיאגרמת זמנים:



### T (trigger)-Latch

- מתקבל ע"י חיבור J,K ב JK-Latch למקור אחד:
- $T = 0$  - אין שינוי במצב
- $T = 1$  - היפוך מצב
- היפוך מצב זה הנו יחיד אם משך הזמן בו  $T = 1$  קצר מספיק.

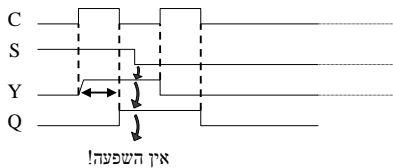
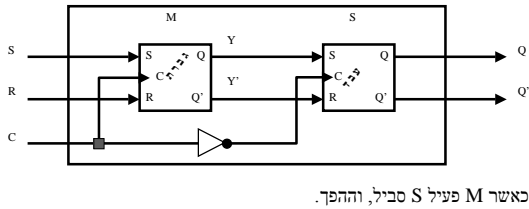
$Q_t$	T	$Q_{t+1}$
0	0	0
0	1	1
1	0	1
1	1	0

$Q_t$	D	$Q_{t+1}$
0	0	0
0	1	1
1	0	0
1	1	1

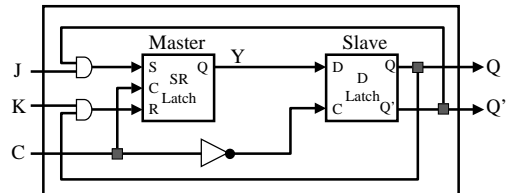
### Flip - Flops (דלגלים)

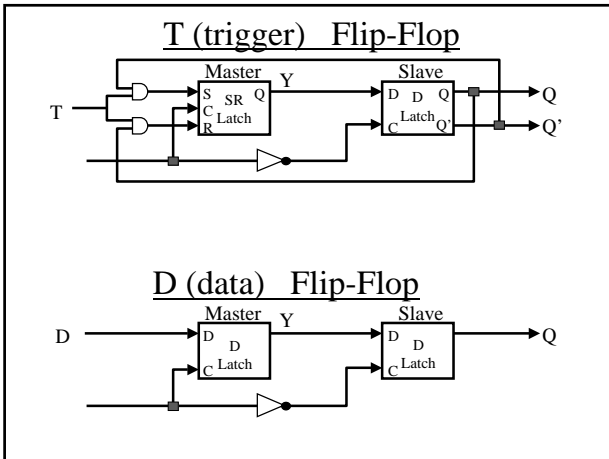
- המוצא של רכיב זכרון אינו צריך להיות תלוי בתזמון, וצריך להיות יציב לפרק זמן מובטח.
- Flip-Flop הוא פתרון המבוסס על Latch. זהו רכיב זכרון של ביט יחיד.
- Edge-Triggered Flip-Flop -1 Master-Slave Flip-Flop הם שני פתרונות שימושיים.
- Master-Slave Flip-Flop הוא פתרון המבוסס על שימוש בשני Latch, בצורה שמבטיחה שהפלט יהיה מבודד מהכניסות.

### Master - Slave SR Flip - Flop



### JK Flip-Flop





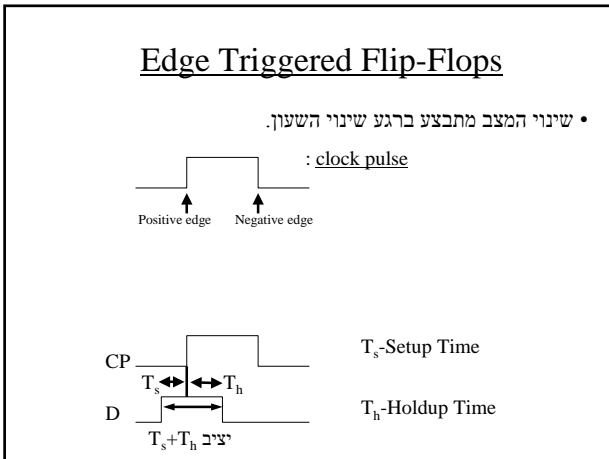
### טבלאות המצבים

J	K	Q(t+1)	
0	0	Q(t)	No Change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

S	R	Q(t+1)	
0	0	Q(t)	No Change
0	1	0	Reset
1	0	1	Set
1	1	?	Illegal

D	Q(t+1)	
0	0	Reset
1	1	Set

T	Q(t+1)	
0	Q(t)	No Change
1	Q'(t)	Complement



### כניסות ישירות (Direct Inputs)

- כאשר מפעילים את המחשב, המצב  $Q(t=0)$  של ה-Flip Flops, איננו מוגדר. לכן יש לבצע אתחול (כי  $Q(t+1)$  הוא פונ' של  $Q(t)$ ).
- אתחול מבוצע ע"י כניסות ישירות אשר קובעות ישירות את המצב (Preset).

PreSet	CP	J	K	Q	Q'
0	∅	∅	∅	0	1
1	↑	0	0	no change	
1	↑	0	1	0	1
1	↑	1	0	1	0
1	↑	1	1	flip state	