12.02.08

מבנה מחשבים- הרצאה 4

חזרה על שיעור קודם-

מפות קרנו של 5 משתנים- יכול להיות שבמבחן תינתן פונקציה עם 5 משתנים שעל מנת לפשט אותה יש צורך במפת קרנו של 4 משתנים.

לדוגמה ב Seven Segments (=רכיב לייצוג ספרות דיגיטליות מ1 עד 9) זוהי פונקציה ב4 משתנים אך על מנת לפשטה אפשר להתייחס רק ל3 משתנים:

	A	B	C	D
(6-1)……				
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

מ10 ואילך יש לנו בטבלה don't care-ים כי המימוש הוא לספרות 1-9.

כשנרצה לפשט את הפונקציה, עבור הספרות 1-7 נעשה מפת קרנו רק עבור D,C,B (A תמיד 0), עבור 8 ו9 גם לא נצטרך להתייחס לכל הביטים, כלומר הפישוט יהיה מהצורה:

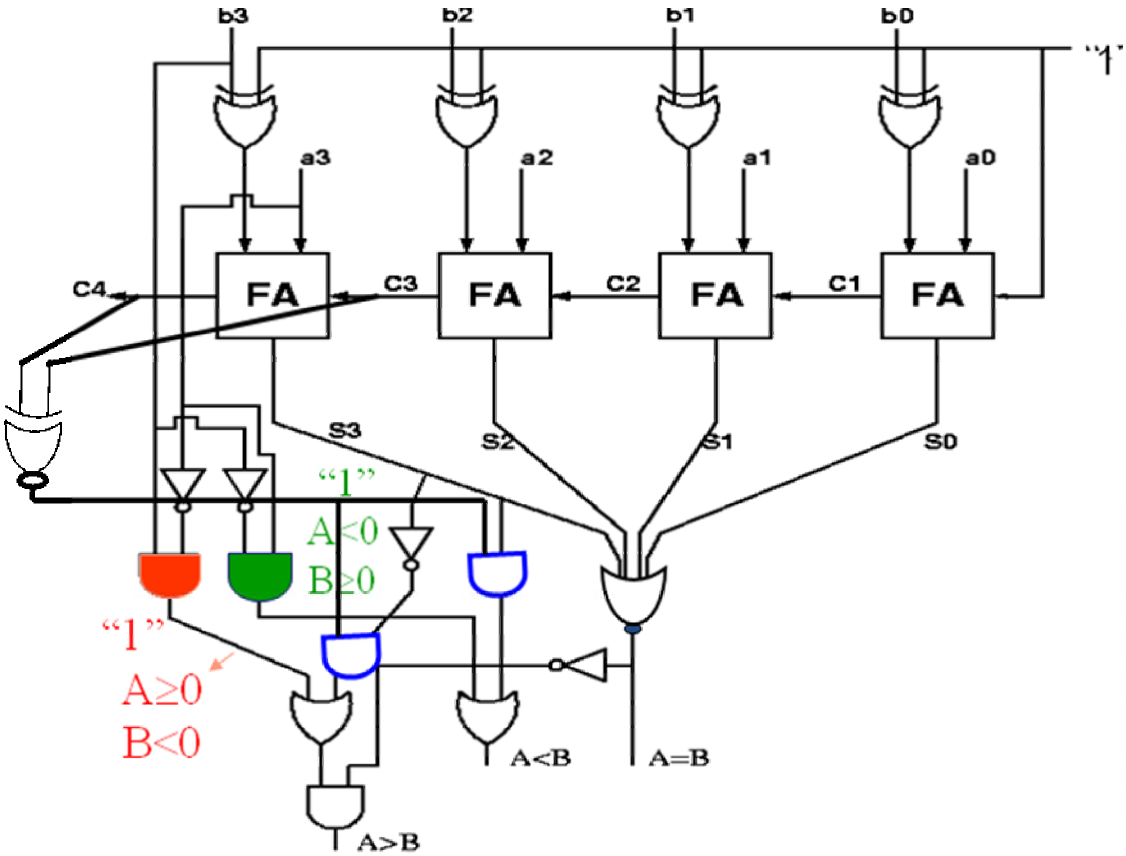
$$\overline{A}(\ldots\ldots\ldots) + A\square ?$$

הפישוט של הפונקציה לספרות 1-7 עם המשתנים B,C,D בלבד.

הפישוט של הפונקציה לספרות 8,9 כאשר '?' שווה ל 1 או ל D או ל $\overline{D}$

משווה גודל- ראינו בשיעור שעבר comparator שבהינתן שני מספרים A וB הוא בודק מי מביניהם גדול יותר או האם הם שווים.

נשים לב כי עבור המקרה A=B בדקנו רק ש $S_3=S_2=S_1=S_0=0$ **<u>ולא ביצענו בדיקת overflow.</u>** הסיבה לכך היא שתמיד עבור תוצאה 0 לא יהיה overflow (אפשר בקלות לבדוק את כל סוגי פעולות החיבור שייתנו תוצאה 0 ולהיווכח).



הערה לתרגיל: הגודל לא קובע ;-)

מימוש פונקציות בוליאניות ע"י מרבבים:

רעיון פשוט:

יש לנו פונקציה בינארית כלשהי ואנו רוצים לממש אותה באמצעות MUX.

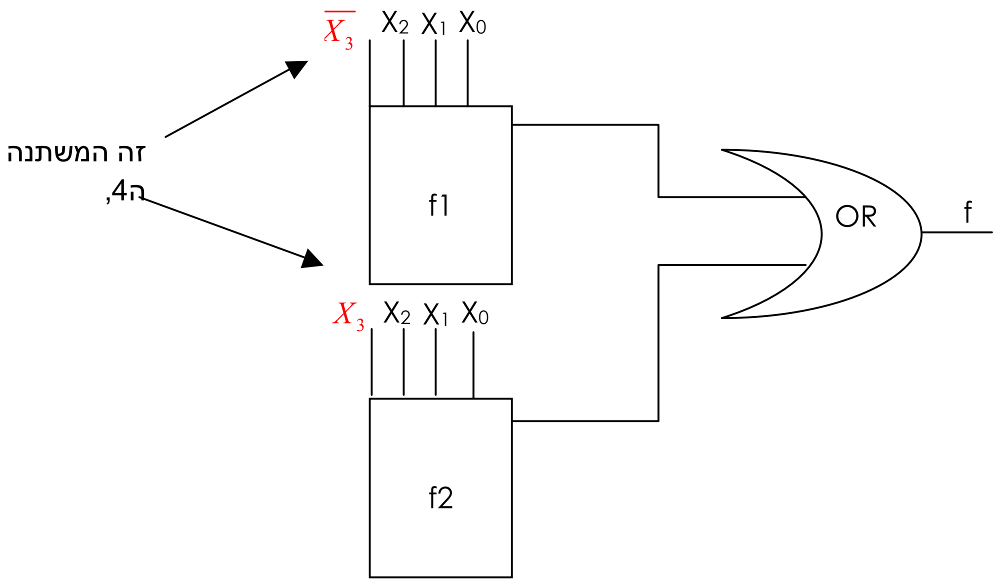
ניקח לדוגמה פונקציה עם 3 משתנים, לשם כך נצטרך MUX עם 2^3 **כניסות** (ממוספרות 0,...,7) **ראשיות** ו3 **כניסות בקרה**. עבור הכניסה הi, i בייצוג בינארי הוא מספר של 3 ביטים, וכל ביט מייצג ערך של משתנה אחד מתוך שלושת המשתנים (מכיוון שההסבר מסורבל נעבור לדוגמה):

נגיד יש לנו פונקציה F. נגיד אנו רוצים לחשב את F(0,0,0). אזי נסתכל על **כניסה ראשית** מספר 0 (הכניסה הראשית העליונה). 0 בייצוג בינארי הוא 000. לכן **בכניסה הראשית** שמספרה 0 יהיה ערך הפונקציה F(0,0,0).

בכניסות הבקרה נכניס את ערכי המשתנים שעבורם אנו רוצים לקבל את הפלט של F (נגיד עבור חישוב F(0,1,0) נכניס בכניסות הבקרה 0, 1, ו- 0). הביטים שהוכנסו **לכניסות הבקרה** יתורגמו למספר בינארי שייצג את מספר **הכניסה הראשית** שתעבור ליציאה וזה יהיה פלט הפונקציה.

(דוגמה בשקפים).

כעת נרצה לממש מעגלים לפונקציות של 4 משתנים באמצעות מעגלים של פונקציות ל3 משתנים, אז נעשה זאת באופן הבא:



אזי $f = \overline{x_3}(f_1) + x_3(f_2)$.

Flip-Flops

<u>הערה:</u> יש לעבור על דפים אלו כמה פעמים על-מנת להבין אותם יותר טוב. זהו חומר חדש וקשה בניגוד לחומר הקודם שהיה ברור יחסית.

עד כה עסקנו ברכיבים ושערים א-סיכרונים. כלומר רכיבים שהפלט שלהם תלוי בללט בלבד, אבל כעת הפלט עלול להיות תלוי גם במצב הקודם (בפלט הקודם) של המערכת. עתה נכניס להגדרתינו מרכיב חדש שילווה אותנו עד סוף הקורס, מרכיב הclock.

מה זה מרכיב הclock? ובכן זהו מרכיב שבעצם יראה לנו את "ציר הזמן" שלנו. הclock הינו קו כניסה שמוכנס למערכת שלנו (בלי הגבלת הכלליות הוא מאותחל להיות 0 בתחילת הפעולה) כך שבכל זמן קבוע הוא כל הזמן ישנה את ערכו מ"0" ל"1" ומ"1" ל"0". יש לשים פה דגש שהזמן שלוקח לשינוי מצבו של השעון הוא יחסית מינורי ביחס לזמן שבו השעון "נשאר במצבו" (ערכו לא משתנה). כמוכן, אין לנו שום השפעה על הclock כלומר זהו כניסה חוץ מערכתית.

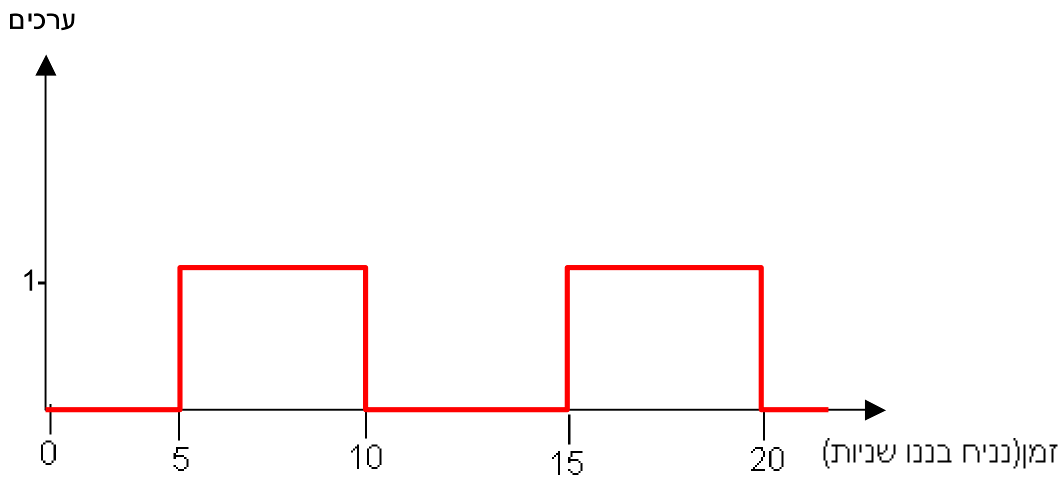
נגדיר את המושגים הבאים:

"עליית שעון": הזמן (המינורי) שבו הclock משנה את ערכו מ0 ל1.

"ירידת שעון": הזמן (המינורי) שבו הclock משנה את ערכו מ1 ל0.

"מחזור שעון": הזמן שבו הclock נמצא בין עליית שעון לעליית שעון שאחריה.

הציור הבא הינו ציור טוב שמשקף את המושגים שלנו.



הקו האדום מייצג את הערך של הclock בזמן מסויים.

בשניות 5 ו15 יש עליית שעון (השעון עולה) בשניות 10 ו20 יש ירידת שעון(השעון יורד) ובין שניות 5 ל15(כלומר מחזור של 10 שניות) יש מחזור שעון בודד.אין כאן לחשוב שעליית/ירידת השעון קורה בזמן אפסי (למרות מה שמצוייר בגרף) למרות שהוא זמן מינורי (פועל על קטע זמן מצומצם).

SR-LATCH

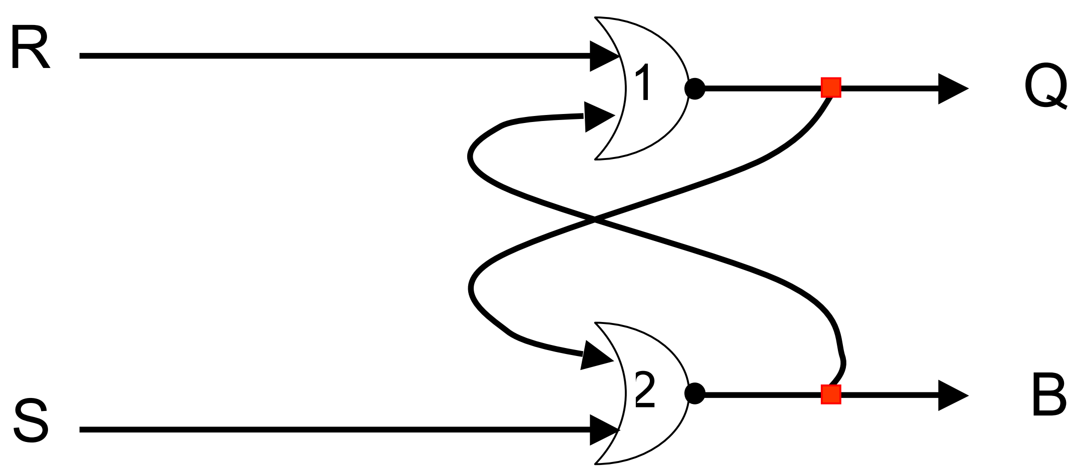
מטרתנו העיקרית פה בהרצאה זאת היא, לבנות רכיב (שיקרא "רכיב זכרון" SR-LATCH) שיחובר לשעון כך שבכל מחזור שעון ("יחידת הזמן") יהיה לו ערך קבוע ולא ישתנה, לא משנה מהם הכניסות שלו (בפרט אם השעון נמצא בירידה או בעלייה באותו מחזור השעון). איך נעשה זאת? נתחיל קודם מבניית רכיב זכרון שיקיים את התנאים הבאים:

1) הכניסות שלו יכולים להשתנות אך ורק במהלך עליית השעון (יעבור ממצב 0 ל1 וההפך) אך בכל מחזור שעון הוא קבוע.

2) הפלט הוא **<u>פונקציה</u>** חד ערכית שתלויה בכניסות ובמצב הקודם בלבד. (בהמשך תיראו למה דרישה זאת לא תמיד מתקיימת).

3) הפלט שלנו יהיה תלוי גם בפלט הקודם שהיה במחזור השעון שלפניו. כמוכן בכל מחזור שעון הפלט שלנו יהיה קבוע ולא ישתנה.

מימוש- (1 ו2 הינם שערי NOR)



S היא מהמילה Set וR מהמילה Reset (ראה טבלה שלמטה).

טבלת אמת (t הינו המחזור שעון מספר t שלנו)

S	R	Q_{t+1}	B
0	0	Q_t	$\bar{Q}_t$
0	1	0	1
1	0	1	0
1	1	מצב אסור (יוסבר בהמשך)	מצב אסור (יוסבר בהמשך)

<u>הערה1:</u> אנו מניחים כי Q_0 (כלומר הערך ההתחלתי של הפלט) הינו ערך זבל(כנ"ל לגבי B_0). לכן נוכל להניח כי Q_0 הינו הערך 0 (אלא אם צויין אחרת). הנחות לגבי ערכי התחלה לכל היציאות של הרכיב תלוייה באיך רשומה השאלה (ובתשובתו של המרצה במהלך המבחן).

<u>הערה2:</u> נשים לב כי בעצם לפי הטבלה פה בכל מחזור שעון (מלבד המקרה של 1=R=S והמקרה ההתחלתי) כי B=$\bar{Q}$. לכן בהרצאה שינו את היציאה B ל$\bar{Q}$.אם זאת,כדי למנוע בלבול רשמתי כך את היציאה B כי יתכן שבהתחלה $B_0=1=Q_0$ ובמקרה של 1=R=S יתקיים כי $B_t=0=Q_t$ (יוסבר בהמשך).

למה זה כך? ובכן אם לדוגמא אנו במחזור שעון t ונניח כי 0=R=S והחלטנו לשנות את S ל1 (ואת R אנו לא משנים) אזי הפלט בשער השני היה 0 לא משנה מה היה Q_t (שכן $\overline{1+Q_t}=\overline{1}=0$) ולכן $B_{t+1}=0$ ואז שער 2 יוציא בעצם 1 (NOR של Q' עם R כלומר NOR של 0 ו-0). כלומר Q_{t+1} שווה ל1.

השאלה המתבקשת פה מטבלת האמת שלמעלה היא למה כאשר 1=R=S יש לנו מצב אסור (כלומר אין לנו יותר התחייבות למה שיקרה אחר-כך)?

ובכן לא תמיד נוכל לבטיח שאם S וR ישנו את ערכם אזי הם יקבלו את ערכם החדשים בדיוק בסיום העליה, ז"א אם לדוגמא S=1=R ואנו משנים את ערכם של R וS ל0 אזי יתכן כי R קודם "יגיע" ל0 מהר יותר מאשר שS "יגיע" ל0(הזמן שלוקח לעליית שעון לעלות הינו איטי יותר מהזמן שיקח לS לעלות(מ1 ל0) וזה יותר איטי מהזמן שיקח לR לעלות מ1 ל0). נשים לב כי כאשר S=1=R אזי Q וB שווים ל0 (כי NOR של x עם 1 תמיד היה שווה ל 0). כאשר נגיע למצב שR=0 אזי עדיין 1=S (נוכל להניח כי הזרם שיעבור דרך S לא ישתנה בקטע זמן שאנו מורידים את הזרם של R) ואז Q יקבל את הערך 1 וB יקבל את 0 (ניתן לבדוק את זה). אח"כ כאשר נוריד את הזרם של S מ1 ל0 אזי התוצאה שלנו לא תשתנה.

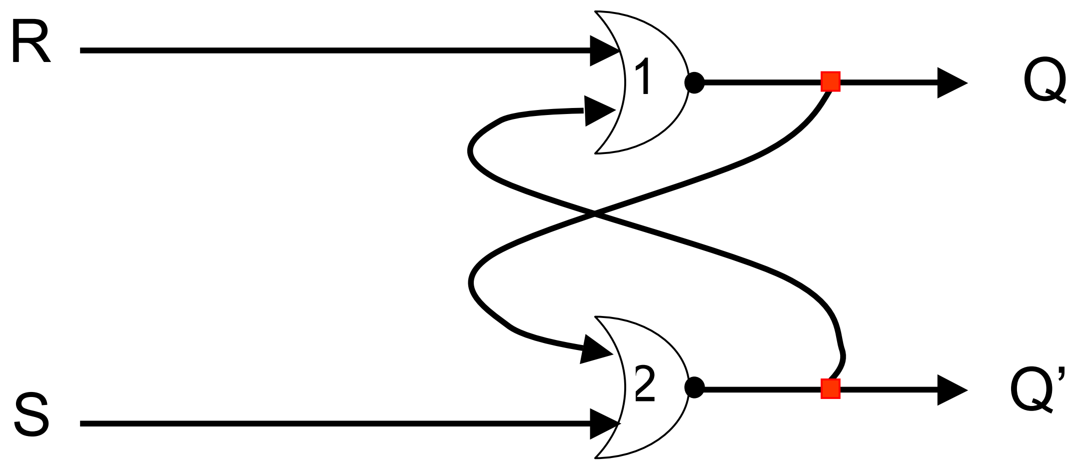
אם זאת, אם S תירד יותר מהר מR כלומר היה קטע זמן מסויים שבו 0=S ו1=R אזי Q=0 וB=1 אח"כ כאשר נוריד את R מ1 ל0 נקבל כי עדיין Q=0 וB=1 .

לפיכך, אם בעליית שעון נשנה את מצבו של R מ1 ל0 ואת מצבו של S מ1 ל0 אזי נקבל תלות בקצב ירידה של R וS (אם S תירד יותר מהר מR או ההפך).אנו <u>**נדרוש**</u> כי קצב העליה\ירידה של S ושל R לא תשפיע לנו על התוצאה הסופית(בסתירה לסעיף 2 שכתוב למטה).

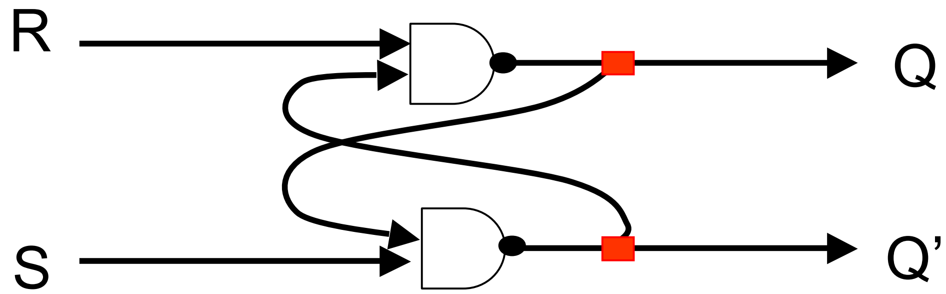
<u>הערה1:</u> נניח שאנו במצב שבו 1=S ו0=R ואנו רוצים לשנות אותו למצב שבו 0=S ו1=R.האם פה יש לנו תלות בקצב העלייה של R ובקצב הירידה של S? ובכן ניתן לבדוק שני מצבים שונים אלו ולקבל תשובה זהה הרי לא משנה מהם הערכים של Q וB בזמן עליית השעון, בסוף עליית השעון עצמה 0=S 1=R ואז הערכים של Q ושל B ישתנו ל0 ול1 בהתאמה. כנ"ל אם 0=S ו1=R יהפוך לS=1 ו0=R

<u>הערה 2:</u> בכל המקרים,מלבד שני המקרים שיש ב<u>הערה 1</u>, אנו משנים אך ורק ביט אחד והביט השני הוא קבוע בעליית השעון, ולכן אין אנו צריכים לעסוק האם לS יש קצב ירידה\עלייה יותר מהיר מקצב הירידה\עלייה של R בניגוד לסעיפים הקודמים.

אחרי שנשנה את B ל $\bar{Q}$ נקבל בעצם את SR Latch הרגיל שאנו מכירים:



ניתן גם לממש את רכיב הSR Latch באמצעות שערי NAND:

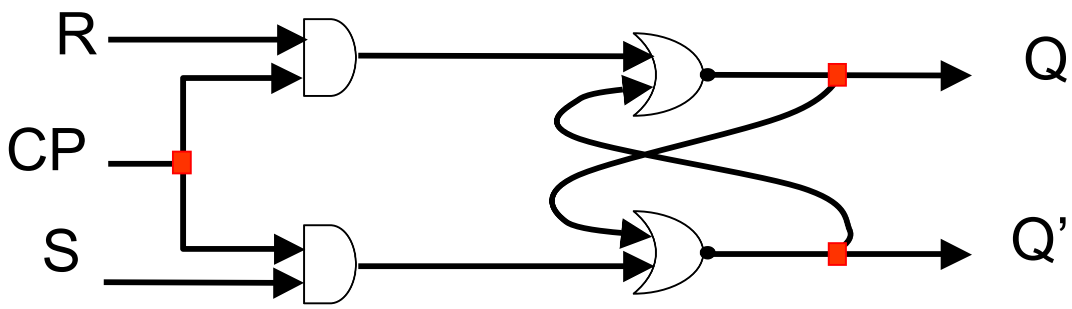


כאן הבעייתיות היא כאשר 0=S=R ואז נקבל כמו כאשר הSR Latch מובע ע"י שערי NOR

כמובן כאשר 1=R=S המצב נשמר.

SR Latch מבוקר שעון

אנו נרצה לפתח יותר את הרכיב שפיתחנו. עתה ניצור רכיב SR כך שכאשר השעון יקבל 0 הוא ישאר במצבו, לא משנה אילו ערכים אנו נכניס לכניסות, אך כשהשעון יקבל 1, זה לא יהיה כך. ברור, שזה לא מה שאנו רוצים (שכן יתכן שכאשר השעון יקבל 1 הכניסות ברכיב ישתנו בתדירות גבוהה ואז לא יובטח לנו מה היה הפלט), אבל זה יקדם אותנו שלב אחד קדימה בדרך למה שאנו רוצים. כמוכן רכיב זה מבוסס על רכיב הזכרון הקודם.



טבלת אמת:

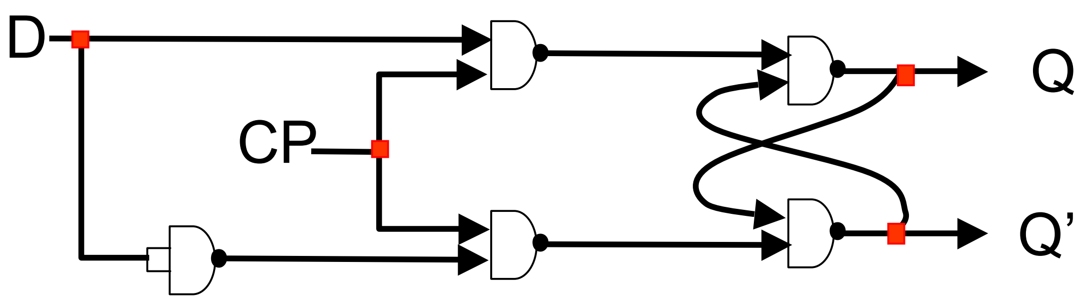
C	S	R	Q_{t+1}
0	$\varnothing$	$\varnothing$	Q_t
1	1	0	1
1	0	1	0
1	1	1	לא מוגדר
1	0	0	Q_t

כאשר "יחידת הזמן" שלנו אורכה כאורך הזמן שלוקח לקלט שלנו להכנס עד שיצא הפלט. ברור שזה זמן מאוד מינורי, ולוקח הרבה פחות זמן מאורך מחזור השעון.

D Latch

עתה נרצה לבנות את אחד מאבני היסוד של האוגרים, הD FF (נבנה זאת בהמשך הדרך). פעולתו של רכיב זה הוא שכאשר הוא מקבל בכניסתו ערך מסויים בעליית שעון, הפלט של הרכיב יהיה קבוע לאורך כל מחזור השעון. כמו כן ב D LATCH הפלט הינו תלוי בפלט שהיה במחזור השעון הקודם, כלומר הוא "זוכר" ביט אחד במהלך מחזור שעון. במקרה פה נבנה את הD Latch אשר יעשה פעולה דומה, חוץ מכך שכאשר השעון הוא 0, הערך הקודם ביציאות ישמר. רכיב זה מבוסס על הרכיב של ה SR מבוקר שעון.

מימוש:



טבלת אמת:

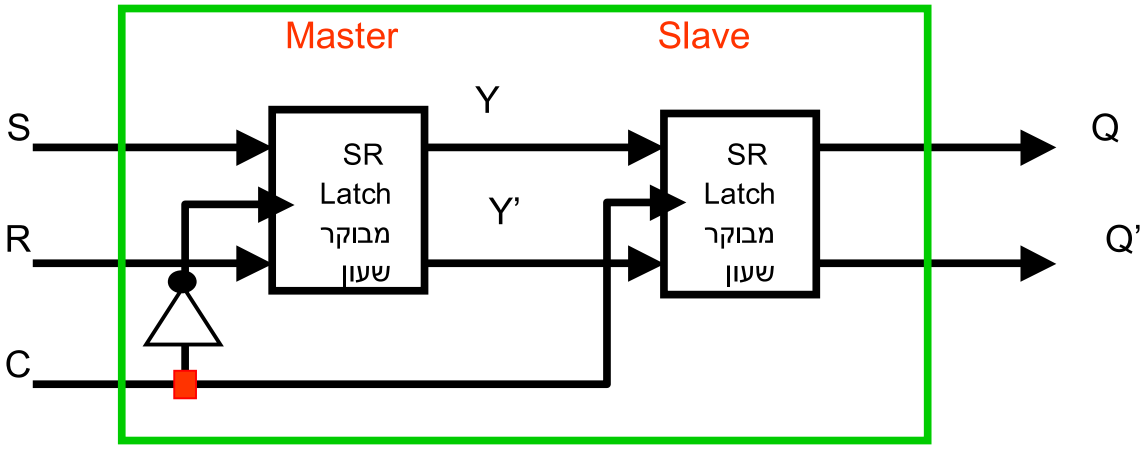
C	D	Q_{t+1}
0	$\varnothing$	Q_t
1	1	1
1	0	0

הבעייתיות היא כאמור שכשהשעון ב1 ואנו משנים את הקלט של הרכיב כאשר אזי הפלט ישתנה, (אם זאת כאשר השעון יקבל 0 אזי הפלט לא ישתנה לא משנה מה יהיה בכניסה שלו).

עתה ננסה ליצור את הflip flop, גירסה יותר מתקדמת מהlatch אשר יקיים שהמערכת תוציא לנו בכל מחזור שעון משהו קבוע . גירסה זאת תתבסס על מודל הMaster-Slave כפי שיוצג בדף הבא.

SR Flip Flop

תרשים:



כלומר הSR Latch הראשון יהיה הMaster ו הSR Latch השני יהיה הSlave

הפעילות של הMaster Slave פועל כך:

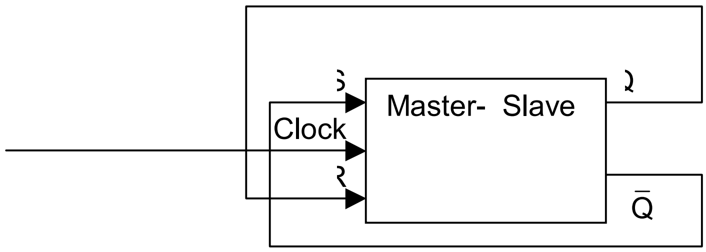
כאשר השעון הוא בעלייה אזי הSlave יקבל את הפלט מהMaster והMaster יסגר, ולכן הSlave יהיה קבוע בעת שהשעון ב 1 וכאשר השעון בירידה הSlave נסגר ולכן הוא לא יקבל ערכים מה Master. כמובן כאשר השעון ב 0 גם כן הSlave לא ישתנה. וזה בדיוק מה שאנו רוצים.

הערות:

1)בהרצאה הMaster-Slave שהוצג הוא יוציא פלט קבוע לא בין שני עליות שעון אלא בין שני ירידות שעון.

2)יש בעייה כאשר S=R=0 , בסוף ירידת השעון. Y יקבל את ערך הזבל שהיה כאשר השעון קיבל 0, לפני תחילת עליית השעון. זה תוצאה לא רצוייה, אם היינו רוצים להתחשב אך ורק בכניסות שאנו נקבל בתחילת עליית השעון. לפיכך נניח כי כאשר השעון ב1 הכניסה קבועה. אם זאת כאשר S=1 וR=0 או S=0 וR=1 אין לנו בעיה זאת (S=R=1 לא מוגדר).

<u>דוגמא למימוש של הMaster Slave-מונה ל2(מונה שסופר מודלו 2)</u>

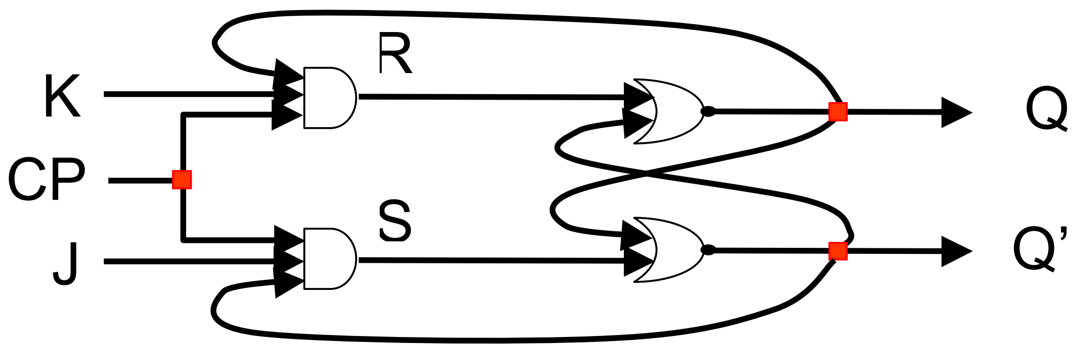


כאשר בהתחלה Q יהיה 0 וב$\bar{Q}$ יהיה 1. אזי במחזור השעון הראשון, Q יהיה 1 ובמחזור שעון השני יהיה 0, במחזור שעון השלישי יהיה שוב 1 וכך הלאה....

<u>JK Latch</u>

הJK Latch הוא מעיין "עידון" של הSR Latch מבוקר השעון, בכך שבמצב ששני הכניסות יקבלו 1 אזי אנו נקבל את ההיפוך של המצב הקודם. נתבסס על כך ש $NOR(Q,\bar{Q})=1$. J מתנהג כמו S וK מתנהג כמו R.

מימוש (כתבתי את R ואת S בשביל טבלת האמת)



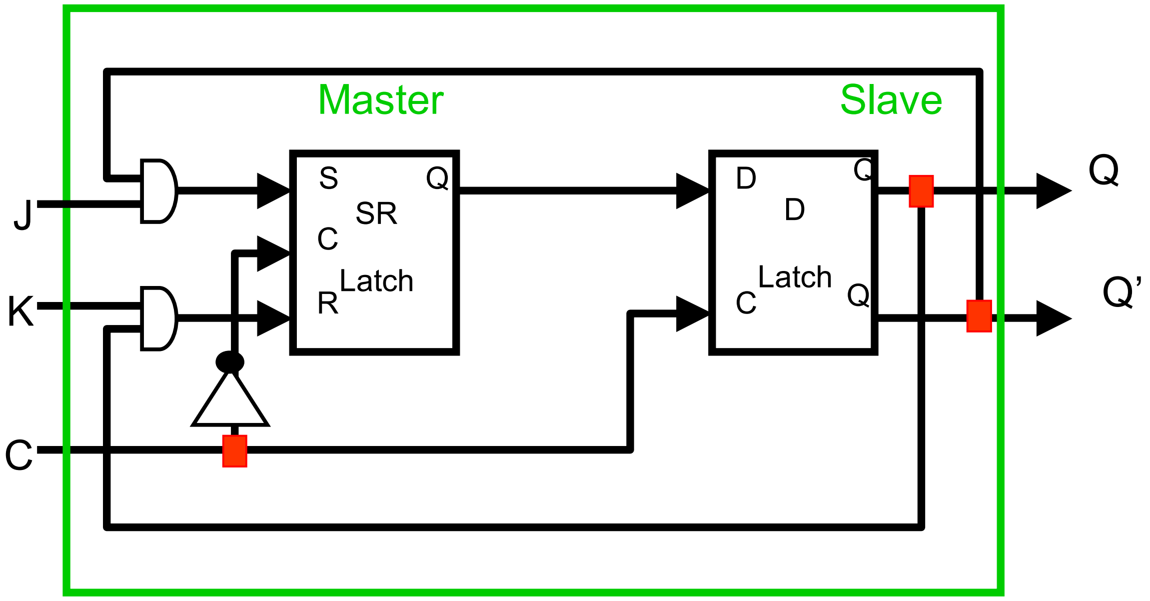
טבלת אמת:

C	J	K	R	S	Q_{t+1}
0	$\varnothing$	$\varnothing$	0	0	Q_t
1	0	0	0	0	Q_t
1	1	0	0	$\bar{Q}_t$	1
1	0	1	Q_t	0	0

1	1	1	Q_t	$\bar{Q}_t$	$\bar{Q}_t$

למרות זאת יש בעייה קטנה כאשר J=K=1 (בכל הקטע שבו השעון מקבל 1) שכן, הרכיב יחליף כל הזמן את הפלט שלו,מ0 ל1 ומ1 ל0.

איך נפתור בעיה זאת? נפתור זאת בעזרת Master-Slave.



הערות:

1)כמו בMaster-Slave SR, בהרצאה הMaster-Slave שהוצג יוציא פלט קבוע לא בין שני עליות שעון אלא בין שני ירידות שעון

2)כמו בMaster-Slave SR כאשר K=J=0 , בצורה זהה J=K=1, נוצרה לנו אותה בעייה שתוארה ב Master-Slave SR.

<u>T Flip-Flop</u>

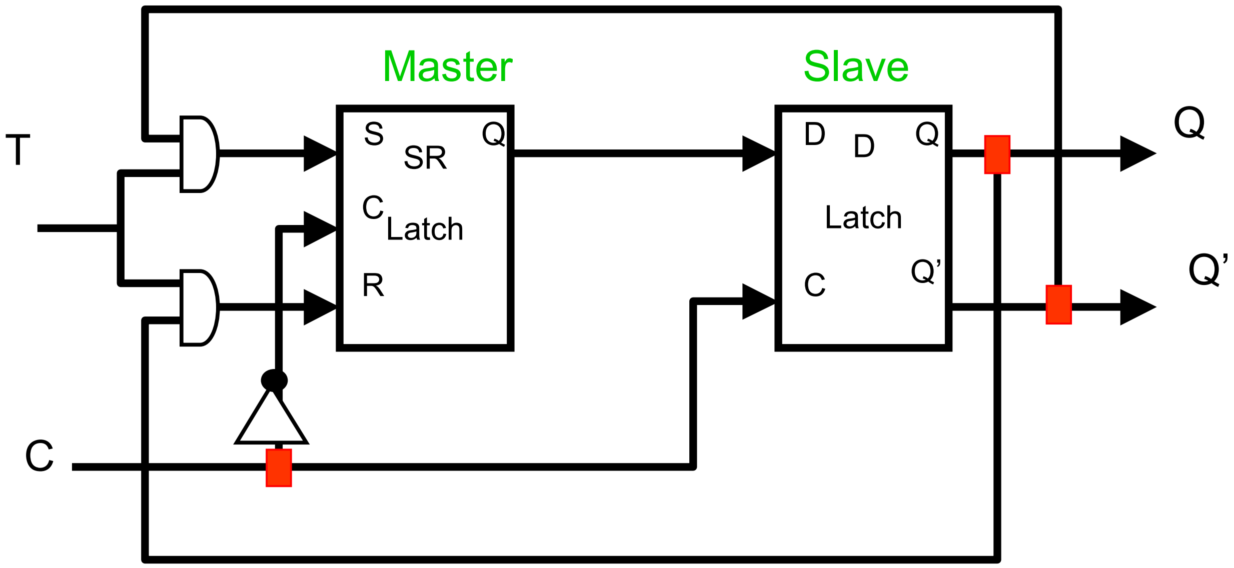
נשים לב כי כאשר נחבר את הכניסות J וK לכניסה משותפת, נקרא לה T, נקבל את טבלת האמת הבאה:

T	Q_{t+1}
∅	Q_t
0	Q_t

1	$\bar{Q}_t$

כלומר לפי ערך הכניסה הרכיב יחליט האם "להפוך" את המצב הקודם.

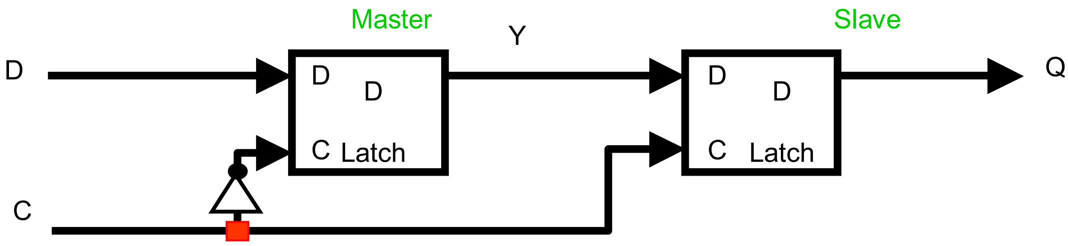
מימוש:



<u>D Flip-Flop</u>

נעשה עתה את המערכת הסיכרונית לD Latch (שוב באמצעות הMaster-Slave)

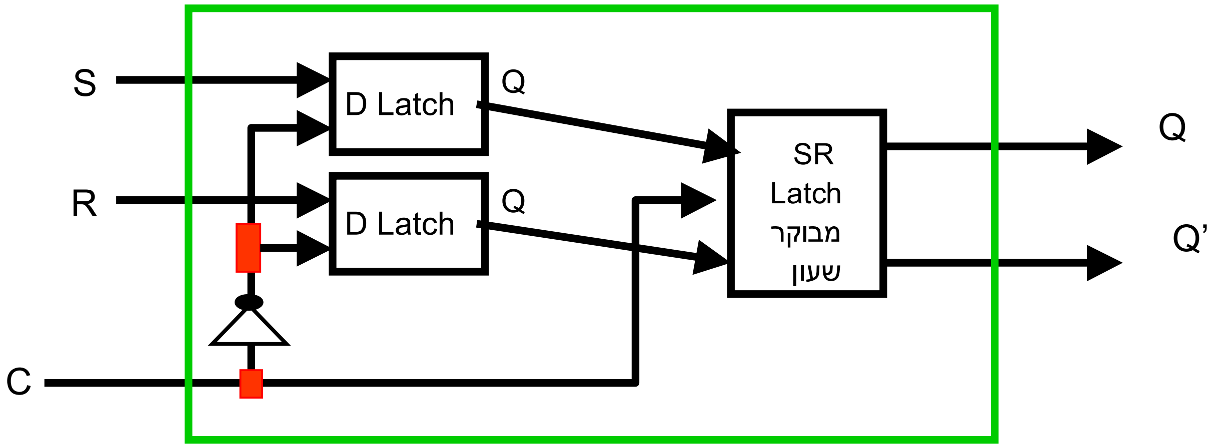
מימוש:



<u>Edge Triggered Flip-Flop(נתן אינטרטור עבר על זה)</u>

בדוגמאות הקודמות של אמרנו כי היינו רוצים לבנות Master-Slave שלנו,כך שרק הכניסות שהוא יקבל בעליית השעון ישפיעו על הפלט הסופי.

על מנת לעשות כך אפשר לעשות שהכניסות יעברו תחילה דרך D latch-ים כך שכל אחד מהם יהיה מחבור לשלילה של השעון(קו הזרם של השעון שעליו יש את השער NOT).לדוגמא ניתן לממש את SR FF בדרך הבאה:



<u>Direct Input</u>

ניתן להכניס לFF כניסה ישירה אשר מאתחלת את הFF.נקרא לכניסה זאת בשם clear.ניתן לממש לדוגמא SR Flip-Flop עם כניסה ישירה של clear. תרגיל טוב למבחן היה לממש Flip-flop כזה.ניתן גם להגדיר כניסות אחרות לFF,כמו Preset אשר יקבל 1 אם"ם השעון בעלייה (נתן לא התייחס לזה).