

תרגיל כיתה: Flip Flops

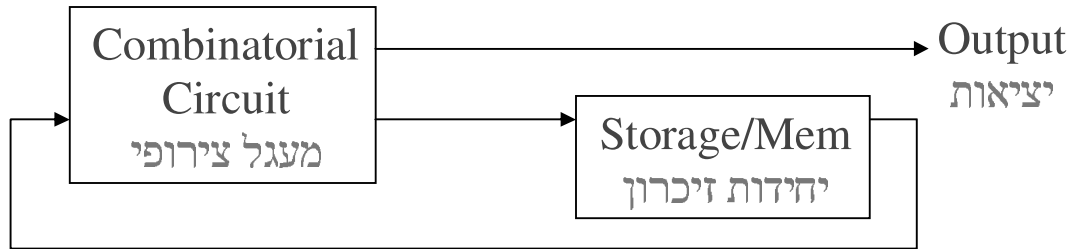
המטרה: לייצר מעגלים עם זיכרון (מצב פנימי) כך שהפלט של המעגל תלוי בערכי הכניסה ובמצב הפנימי. נרצה שקצב השינויים ביציאות ובמצב הפנימי יהיה מתואם עם שעון. המצב הפנימי מאפשר לבנות רכיבים שהתנהגותם תלויה בסדר (sequence) שבו משתנות הכניסות, ולא רק בערכן הנוכחי (הרחבה של יכולת החישוב). התאום עם השעון מקל מאוד על בעיות בתזמון שנוצרות בגלל קיום משוב.

Sequential Circuit: מעגל לוגי שמכיל קשרי משוב. כתוצאה מכך, ערכי היציאות של המעגל תלויים בערכי הכניסות ובמצבי המעגל (ערכי הרכיבים המרכיבים אותו) לאורך זמן הריצה.

Asynchronous Sequential Circuit: מעגל שמצבו אינו משתנה בתאום עם השעון. ערכי הכניסות והמשוב יכולים להשפיע על מצב המעגל בכל רגע.

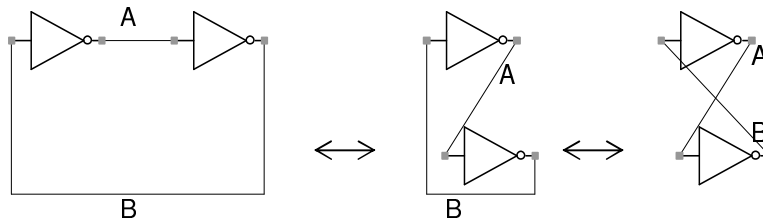
Synchronous Sequential Circuit: מעגל שהתנהגותו תלויה בערכי הכניסות והרכיבים הפנימיים רק בזמנים בדידים (תלויי שעון). המעגל משנה את מצבו התאום עם השעון.

תרשים של Sequential Circuit ניתן לחלוקה לשני חלקים: חלק הכולל מעגל צירופי וחלק הכולל רכיבי זיכרון, בעלי מצב פנימי.

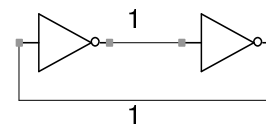


בתרגול זה נבין איך בנויה יחידת זיכרון ואיך מתארים את פעולתה. נתחיל מיחידות אסינכרוניות (Latch) ונשתמש בהן כדי לבנות יחידות סינכרוניות (Flip Flop). הערה: בספרות המונחים Flip-Flop משמש לפעמים גם לתיאור רכיבים אסינכרוניים.

נשים לב שישנם מעגלים עם מעגלים יציבים ומעגלים עם מעגלים לא יציבים. למעגל הבא יש 2 מצבים יציבים כש $A=B$:

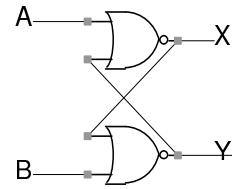


במעגל זה יש גם מצבים לא יציבים:



אין למעגל כניסות ולכן לא ניתן להשפיע על התנהגותו.

(Set Reset) SR Latch



המצב הרגעי הבא של המעגל ניתן לתיאור ע"י הנוסחאות הבאות:

$$X(t+1) = (A + Y(t))'$$

$$Y(t+1) = (B + X(t))'$$

נתאר זאת בטבלת אמת את השינויים בערכי היציאות כפונקציה של הערכים הנוכחיים והכניסות

| A | B | X(t) | Y(t) | X(t+1) | Y(t+1) |
|---|---|------|------|--------|--------|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

כדי שבין מה קורה למערכת בצעדים הבאים – נתאר זאת שוב בצורה קצת שונה:

| | A=0 B=0 | | A=0 B=1 | | A=1 B=0 | | A=1 B=1 | |
|-----|------------|-----|------------|-----|------------|-----|------------|-----|
| | X=0 | X=1 | X=0 | X=1 | X=0 | X=1 | X=0 | X=1 |
| Y=0 | | | | | | | | |
| Y=1 | | | | | | | | |

הניתוח הנ"ל מאפשר לנו לתאר על אילו ערכים סופיים יתיצב המעגל:

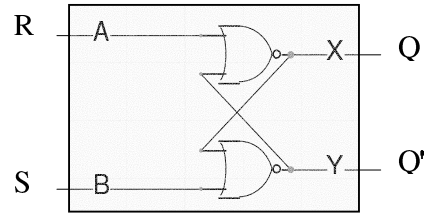
אם $A=1, B=1$ יציאות המעגל תתיצבנה על $X=Y=0$ ללא תלות בערכן הקודם.

אם $A=0, B=1$ יציאות המעגל תתיצבנה על $X=1, Y=0$ ללא תלות בערכן הקודם.

אם $A=1, B=0$ יציאות המעגל תתיצבנה על $X=0, Y=1$ ללא תלות בערכן הקודם.

אם $A=0, B=0$: אם $X=Y$ היציאות לא תשתנה ואם $X \neq Y$ לא ניתן לדעת על אילו ערכים יתיצב המעגל.

כעת נבין כיצד ניתן לעשות שימוש במעגל כרכיב זיכרון (אסינכרוני):



נביל את השימוש ברכיב למצבים בהם $X=Y'$ (ע"י כך שנתחיל ממצב שכזה ונאפשר רק חלק מערכי הכניסות האפשריים) ונסמן את שתי היציאות באופן שישקף זאת (Q, Q').
 נקרא למצב בו $Q=1$ כמצב בו הרכיב הוא SET ולמצב בו $Q=0$ כמצב ראשוני בו הרכיב הוא CLEAR.
 לכניסה B נקרא S (set) כי כאשר הכניסות הן $S=1, R=0$ אז Q עובר ל-1 (Q is SET).
 לכניסה A נקרא R (reset) כי כאשר הכניסות הן $S=0, R=1$ אז Q עובר ל-0 (Q is RESET/CLEAR).
 כל עוד נימנע מערכי כניסה $S=R=1$ אז $X=Y'$ ולכן – כאשר $R=S=0$ המעגל יציב וערכי היציאות לא משתנים.

ניתן לסכם את התנהגות הרכיב בשתי טבלאות שימושיות:

טבלת עירור \ טבלה אופיינית:

טבלה זו מסכמת עבור כל ערכי הכניסה האפשריים – מה יהיה ערך היציאה (החיובי) כפונקציה של ערכי הקודם:

| S | R | Q(t+1) |
|---|---|---------|
| 0 | 0 | Q(t) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | לא חוקי |

הטבלה הזו נובעת מטבלת המצבים שתוארה קודם ומתארת מה קורה למעגל בזמן ריצה לאחר שהתיצב.

אפשר גם להציג את הטבלה האופיינית במפת קרנו ובנוסחת נסיגה (מצומצמת):

| SR | | Q | | | |
|----|---|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q | 0 | | | X | 1 |
| | 1 | 1 | | X | 1 |

(זה מימוש אפשרי נוסף) $Q(t+1) = S + R'Q(t)$

טבלת מעברים:

טבלה זו מסכמת עבור כל שינוי אפשרי של ערך היציאה (החיובי) מה צריכים להיות ערכי הכניסה המתאימים לשינוי:

| From Q(t) | To Q(t+1) | S | R |
|-----------|-----------|--------|--------|
| 0 | 0 | 0 | ϕ |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | ϕ | 0 |

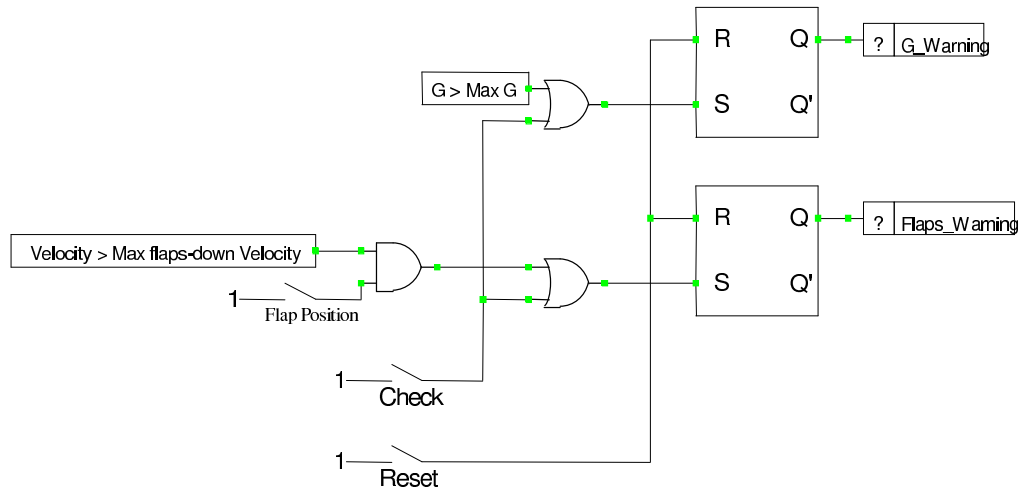
הטבלה הזו עוזרת לתכנן מעגל עם זיכרון. ראשית קובעים את מצבי הזיכרון, משמעותם והמעברים הרצויים ביניהם. בהמשך דואגים שערכי הכניסות לזיכרון יגרמו למעברים הרצויים (ע"י שימוש בטבלה).

דוגמא לשימוש ב RS-Latch

נבנה שברצוננו לממש מערכת במטוס שמדליקה נורת אזהרה אם המטוס ביצע תמרון שחורג ממעטפת הביצועים המותרת. במטוס יש חיישנים שנותנים – 1 ברגע שה-G חרג מהמהירות המותרת (Max G) וברגע שהמהירות (Velocity) גדולה מהמהירות המירבית המותרת עם מדפים מטה (Max flaps-down velocity).

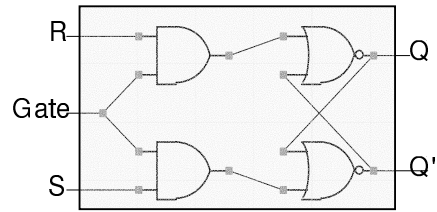
נרצה שנורת אזהרה תדלק במעבר מגבלת G ותשאיר דלוקה עד לביצוע Reset ע"י הטייס.
 נרצה שנורת אזהרה נוספת תדלק במעבר מהירות מדפים מירבית, אך רק עם המדפים הם במצב מטה ותשאיר דלוקה עד לביצוע Reset.
 נרצה גם מתג בדיקה לנורות.

להלן מבנה אפשרי:



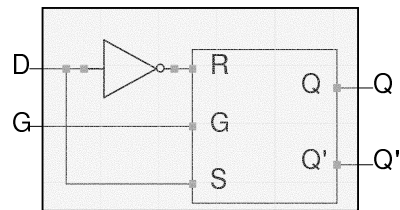
מה קורה אם לוחצים Reset במצב של חריגה ממעטפת הביצועים?
 תשובה: מצב אסור. הנורה תהבהב ותתיצב במצב לא ידוע (דלוק או כבוי).

Gated SR Latch



כאשר ה- Gate=1 פועל כמו SR Latch. כאשר ה- Gate=0 היציאות לא משתנות.

Gated D-Latch



כיוון שהשימוש שתיארנו ב SR Latch עושה שימוש רק במצבים בהם R ו S שונים, די לנו בכניסה אחת והיפוכה (ע"י הוספת שער NOT).
טבלת עירור:

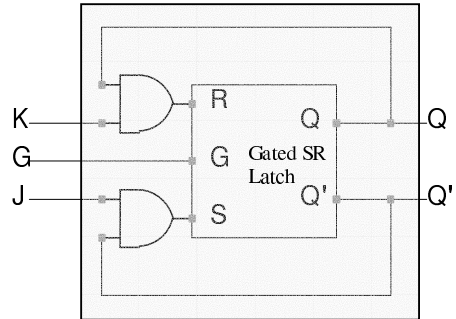
| D | Q(t+1) |
|---|--------|
| 0 | 0 |
| 1 | 1 |

טבלת מעברים:

| From Q(t) | To Q(t+1) | D |
|-----------|-----------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

לרכיב זה קוראים D(ata) Latch כי (כאשר ה- Gate=1) הוא פשוט מעביר את ה data.

(Gated) JK Latch



זהו רכיב שמתנהג כמו Gated SR Latch בתחום החוקי שלו ומתנהג נותן יציאות עם סימן הפוך גם כשהכניסות הן $J=K=1$

טבלת עירור (אופיינית):

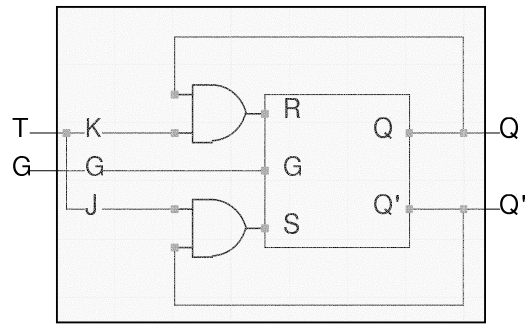
| J=S | K=R | Q(t+1) |
|-----|-----|--------|
| 0 | 0 | Q(t) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Q(t)' |

נשים לב שכשהכניסות הן 1,1 (זה-1-Gate) היציאות הופכות סימן שוב ושוב (באופן אסינכרוני).

טבלת מעברים:

| From Q(t) | To Q(t+1) | J | K |
|-----------|-----------|--------|--------|
| 0 | 0 | 0 | ϕ |
| 0 | 1 | 1 | ϕ |
| 1 | 0 | ϕ | 1 |
| 1 | 1 | ϕ | 0 |

T Latch



זה JK-Latch שחיברו לא את הכניסות לכניסה אחת.
טבלת עירור:

| | |
|---|--------|
| T | Q(t+1) |
| 0 | Q(t) |
| 1 | Q(t)' |

ניתן לראות שכש T=0 היציאה נשמרת וכש T=1 מתהפכת שוב ושוב (כל עוד ה-Gate=1)

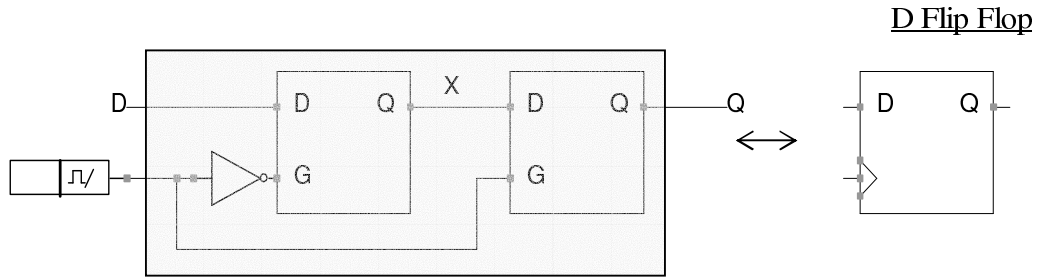
טבלת מעברים:

| From Q(t) | To Q(t+1) | T |
|-----------|-----------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

מכל הרכיבים שתוארו ניתן לקבל רכיבים תלויי שעות ע"י כך שנהבר את Gate של Gated Latches לשעות. באופן זה נקבל רכיבים שמשנים את ערכם רק כאשר השעות נותן 1. אם ניתן לדאוג לכך שהשעות יתן 1 לזמן מספיק קצר כך שהרכיב יתעדכן בדיוק פעם אחת – הרי שקיבלנו מעגל סינכרוני. מגבלה זו היא בעייתית. בהמשך נראה רכיבים ללא מגבלה זו.

רכיבים מתוזמני שעות (synchronized) Flip Flops –

עד עכשיו ראינו רכיבים שפעילים ברמה - 0 או 1 של השעות (level triggered) Flip Flops. הם רכיבים ממותגי שפה (Edge Triggered). כלומר- הרכיב רגיש לשינויים רק בזמן קצר סביב המעבר 0-1 או 1-0 של השעות. ניתן לממש רכיב שכזה כמה דרכים. דרך מקובלת היא מבנה ה-master-slave.



כל עוד השעות ב-0 שינויים בכניסה משנים את ערכו של X אבל ה-Latch השני נעול (כשמו כן הוא) ולכן היציאה Q לא משתנה. כאשר השעות עולה ל-1 קורים 2 דברים:
 1. ה-Latch הראשון מפסיק לעדכן את X. מרגע זה X אינו מושפע משינויים בכניסה.
 2. ה-Latch השני משחרר את הבעילה ו-X משפיע על היציאה.
 כאשר השעות חוזר ל-0 ה-Latch השני נעול והשינויים בכניסה יחולים להשפיע שוב על X.

רכיב זה יקרא **Positive Edge Triggered** כי השינויים ביציאה ובערך הפנימי נקבעים עם עליית השעות. רכיב שעובד הפוך יקרא **Negative Edge Triggered**. הסימון המקובל ל-Positive Edge Triggered זה משולש בכניסת השעות (עבור Negative – מוסיפים עיגול).

טבלאות העירור והמעבר עבור ה-Latches אינן משתנות – אך כעת יש משמעות למושג $t, t+1$ (עליית/ירידת השעות הבאה).

באופן דומה אפשר לשרשר רכיבים מסוג SR, JK, T לקבלת ה-Flip Flops המתאימים.

תזמונים ב-Flip Flops

בדומה לפרמטרי ההשהיות ברכיבים צרופיים, מוגדרים פרמטרים דומים גם עבור רכיבים ממותגי שפה (Edge-Triggered).

תזמון ערכי יציאות

t_{pC-Q} זמן התפשטות מכסימלי מרגע עליית השעות ועד שמוצא ה-Flip-Flop מתייצב בערך החוקי.

t_{cC-Q} הזמן המינימלי מרגע השפה הפעילה של השעות בו עדיין מובטח שמוצא הרכיב ישאר יציב ברמתו הלוגית הקודמת.

תזמון ערכי כניסות

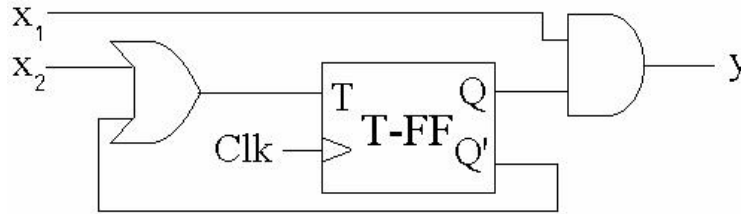
t_s משך הזמן לפני עליית השעות בו הכניסה ל-Flip Flop צריכה להיות יציבה.

t_h משך הזמן לאחר עליית השעות בו הכניסה לפליפ-פלופ צריכה להישאר יציבה.

בדרך כלל t_h ו- t_s קטנים בהרבה מן ההשהיות האחרות במערכת, ומניחים שהם 0. גם t_{cC-Q} הוא בדרך כלל קטן, אולם מניחים שאיננו 0, אלא מקיים $t_{cC-Q} > t_h$. דרישה זו חיונית בכדי שנוכל לחבר יציאת פליפ-פלופ אחד ישירות לכניסת פליפ-פלופ אחר:

דוגמא :

נתון המעגל הבא :



עדכון הכניסות x_1, x_2 נעשה ברגע ירידת השעון. ה-FF נועל בעליית השעון.

$$t_p(\text{OR}) = 10\text{nSec}$$

$$t_p(\text{AND}) = 8\text{nSec}$$

$$t_{\text{setup}}(\text{T-FF}) = 7\text{nSec}$$

$$t_{\text{hold}}(\text{T-FF}) = 3\text{nSec}$$

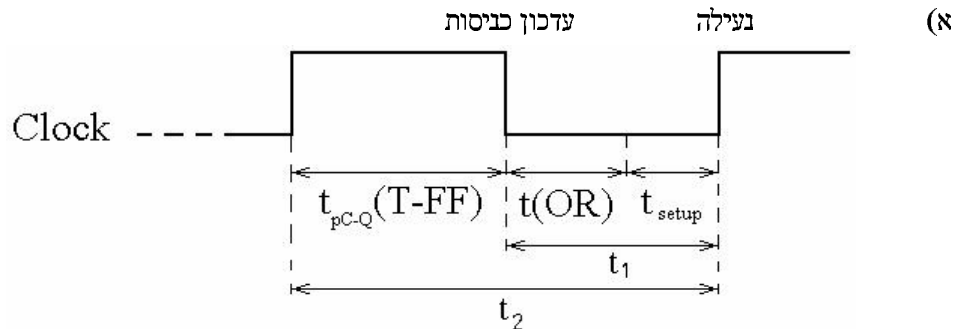
$$t_{pC-Q}(\text{T-FF}) = 10\text{nSec}$$

(א) חשב את מחזור השעון המינימלי הדרוש.

(ב) מהי ההשהיה המכסימלית מעדכון הכניסות עד לעדכון היציאות ?

(ג) מהו התנאי על זמני הזיהום (Contamination times) של הרכיבים, שנדרש לפעולה תקינה של המעגל ?

פתרון :



מרגע עדכון הכניסות אין לבצע נעילה עד שהעדכון יגיע לכניסות ה-FF, ונחכה לפחות t_{setup} , כלומר :

$$t_1 \geq t_p(\text{OR}) + t_{\text{setup}}(\text{FF}) = 10 + 7 = 17\text{nSec}$$

מאחר והיציאה Q' משמשת כמשוב לכניסת שער ה-OR. לא נוכל לבצע ירידת שעון (ועדכון כניסות חודש) לפני העדכון הקודם של Q'. ולכן:

$$t_2 \geq t_1 + t_{pC-Q}(\text{FF}) = 17 + 10 = 27\text{nSec}$$

(ב) הכניסה x_2 מחוברת ליציאה רק דרך רכיב הזכרון, ולא תשפיע על ערך היציאה באותו מחזור השעון. לכן נחשב רק את ההשהיה מ- x_1 ל- y :

$$t_p(x \rightarrow y) = t_p(\text{AND}) = 8\text{nSec}$$

ג) כאן נכנס לפעולה תנאי ה- hold. כמה זמן אחרי הנעילה אסור עדיין לשנות את הכניסות של ה- FF? אם בצענו נעילה, ועדכון Q' נעשה מיידית וכן עדכון יציאות ה- OR, אזי הכניסה ל- FF עלולה להשתנות טרם זמנה ולהפר תנאי ה- hold. לכן חייב להתקיים

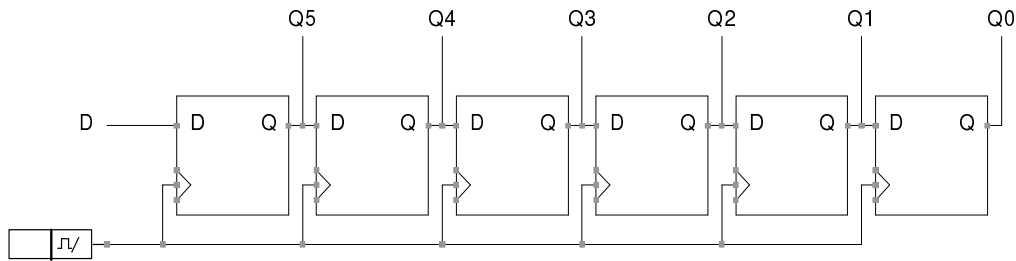
$$t_{hold} < t_{contamination(T-FF)} + t_{contamination(OR)}$$

ניתן לסכם את דרישות התזמון במעגל עם Flip Flops פעילי שפה חיובית באופן הבא:

- השעון חייב להיות 1 לפחות T_{pC-Q} .
- השעון חייב להיות 0 לפחות $T_{setup} + T_{pCOMBINATORIAL}$.
- מעגל הזיהום הקצר ביותר בו נמצאה רכיב זיכרון חייב להיות גדול מה- T_{hold} שלו.

שימושים פשוטים של רכיבי זיכרון - Serial to Parallel converter

נרצה להכניס את ה Data bit by bit – ולקרוא אותו בבת אחת (פעם בכמה זמן).



לו היינו משתמשים ב Latch במקום Flip Flop ה Data היה יכול לעבור מספר Latches בעל רמה (פעילה) של השעון.