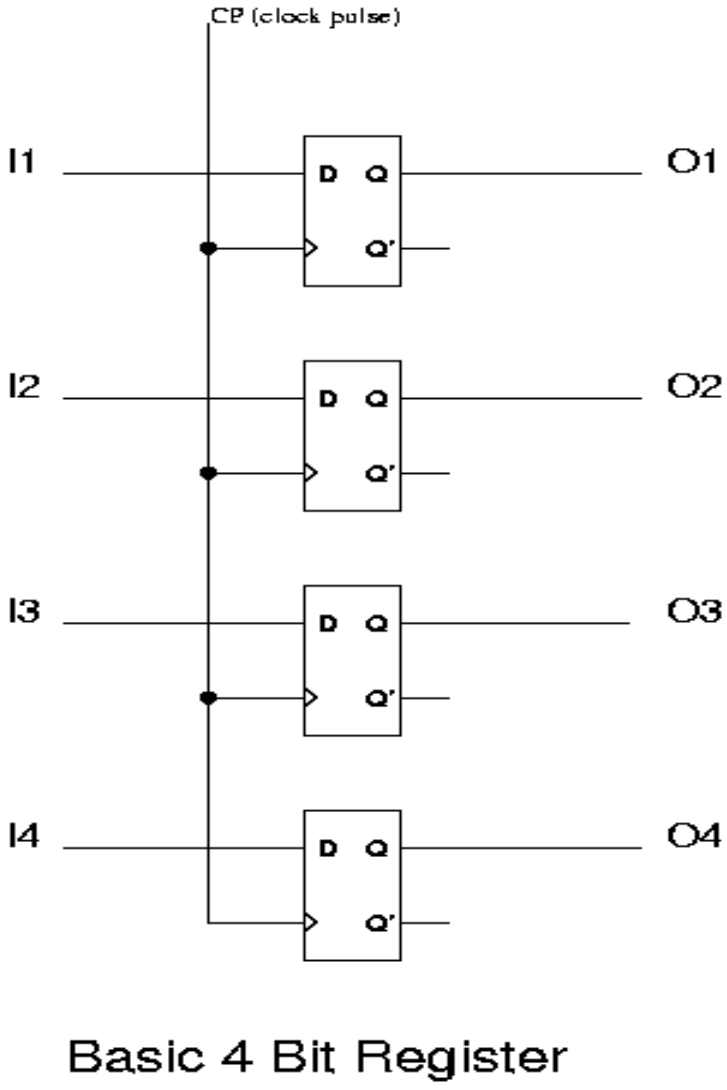
Registers:

הגדרה: **רגיסטר** הוא אוסף של דלגלגים(Flip-Flops) מסוג מסויים(אלו היו לרוב D או T דלגלגים) המשמשות לאגירת נתונים.ניתן לחשוב על כך שריגסטר הוא מעיין מערך של ביטים.בהמשך החומר נשתמש בהם הרבה

השם בעברית:אוגר

הגדרה: הכנסתביטים **לרגיסטר בצורה מקבילית** הינה הכנסת ביטים לרגיסטר בצורה שבו אנו מכניסים לכל כניסה של הרגיסטר איזשהו קו-כניסה(בצורה דוגמא ניתן להגדיר שיציאה בצורה מקבילית זה שהפלט הוא הפלט של כל הFF).

לדוגמא: רגיסטר 4bit(4 כניסות)

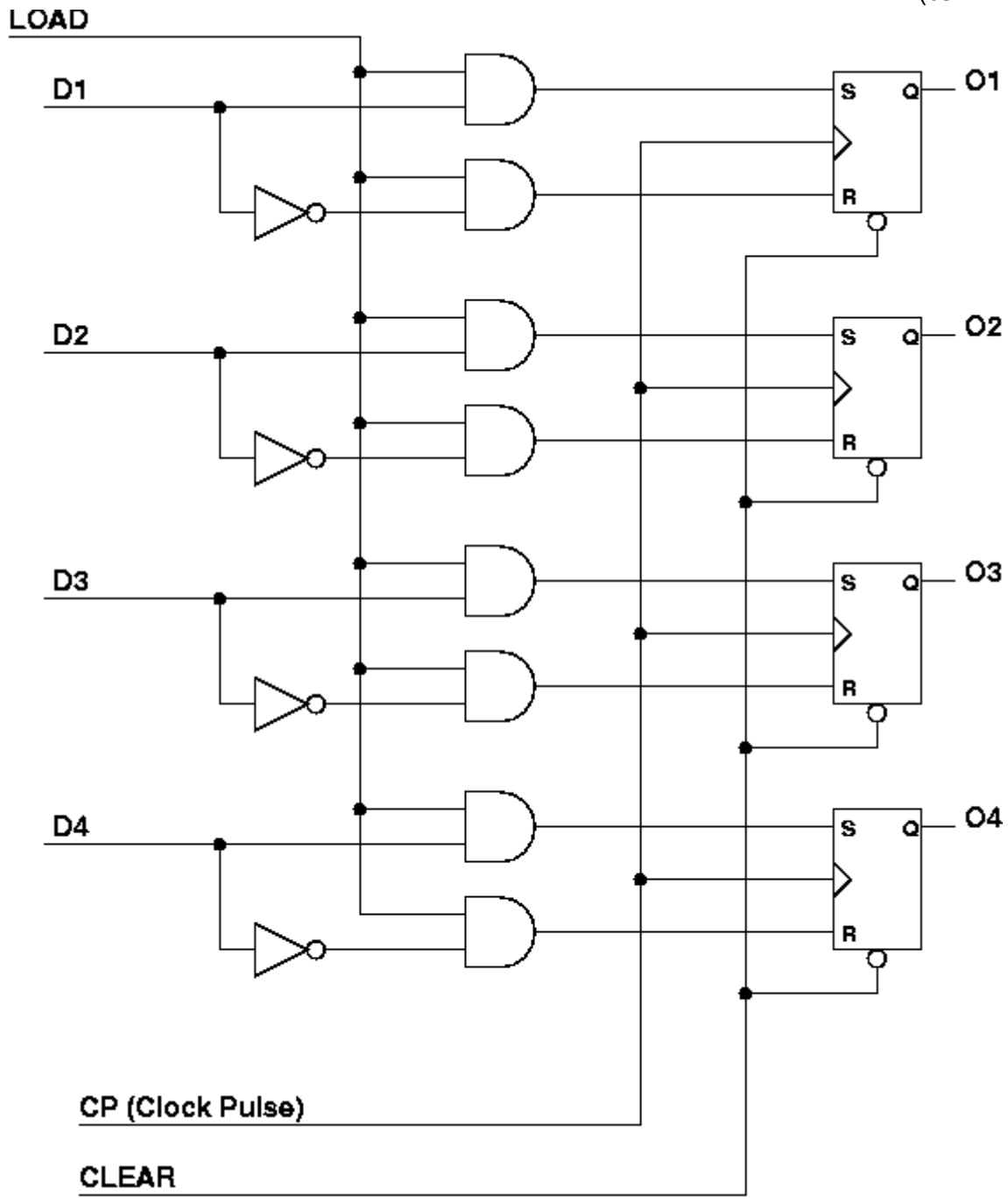


Basic 4 Bit Register

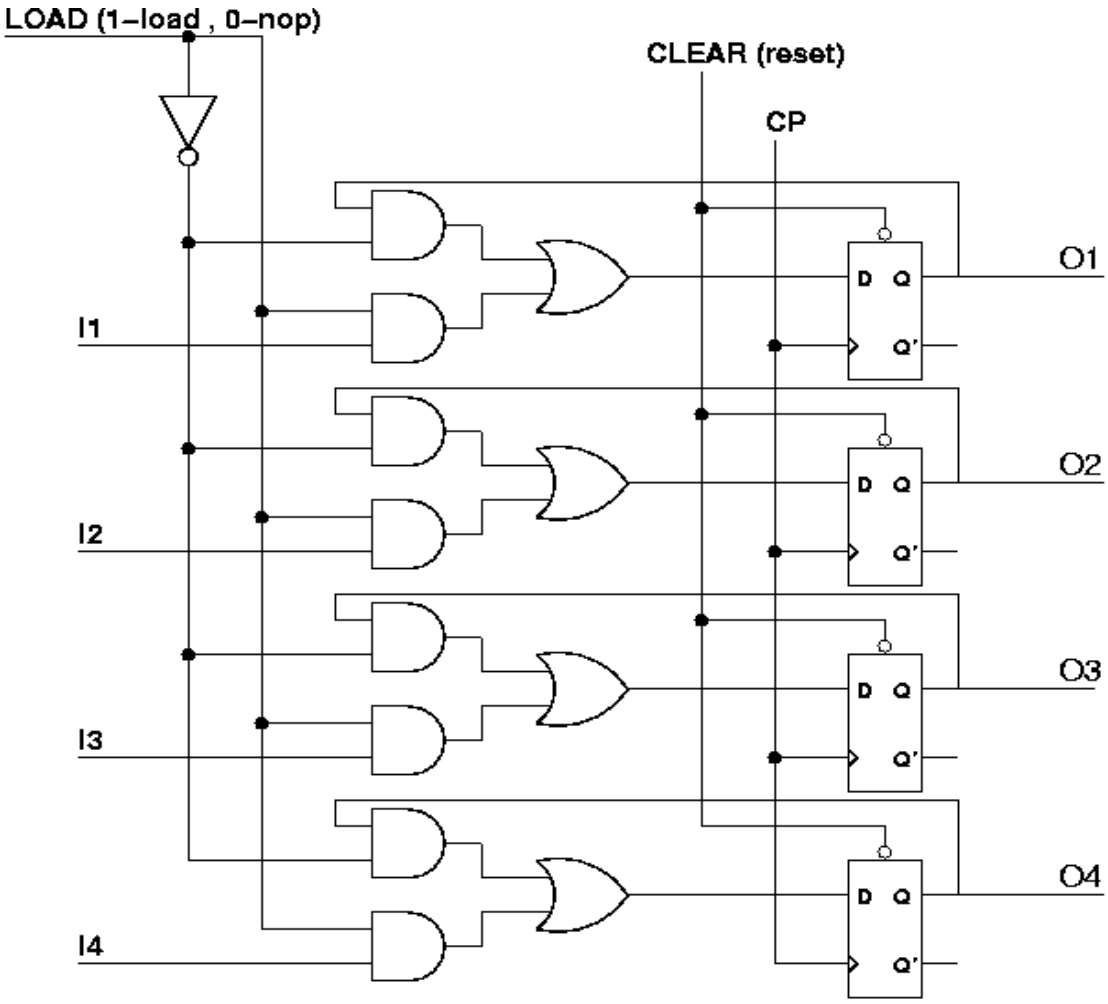
הערה: הריגסטר יכול לקבל/להוציא את כניסותיו כאשר השעון בעליה או בירידה(ניתן להניח בלי הנחת הכלליות שהוא מקבל בעליה).

דוגמא2: רגיסטר 4bit עם טעינה מקבלית ועם יחידת כניסה load,כלומר הרגיסטר יטען את הביטים שבכניסה אם"ם load=1, ויחידת כניסה clear,יחידת כניסה א-סיכרונית,שקיימת גם בכל דלגלג כך

שאם הוא מקבל 1 אזי הרגיסטר מתאפס(בדוגמא בכיתה יוצא שאם clear=1 אזי הרגיסטר מתאפס):



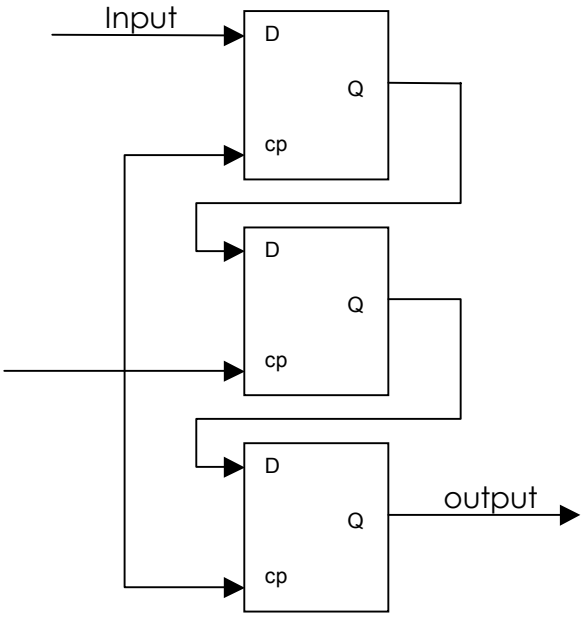
או לחילופין,במקום SR FF ניתן לעשות רגיסטר זה בעזרת D FF:



דוגמאות לשימוש ברגיסטרים שבדוגמא 2 ו3 ניתן למצוא בעמודים 7 ו8 בהרצאה.כמוכן ברור שהיציאה פה היא מקבלית.

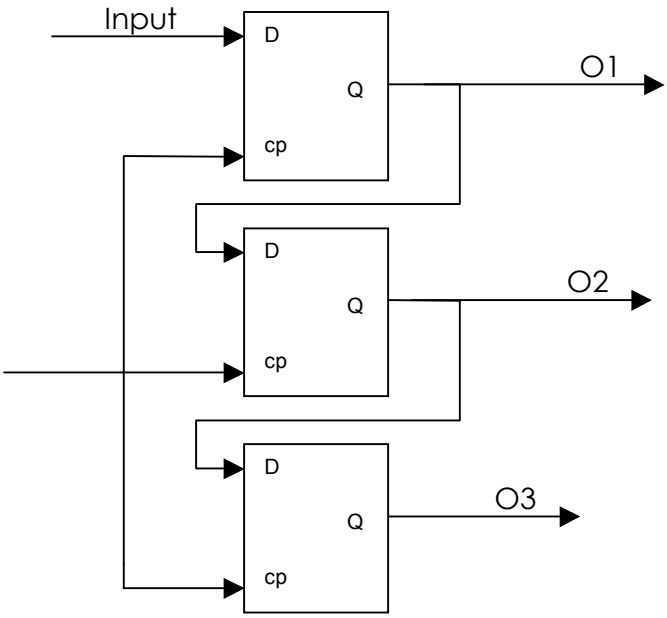
<u>הגדרה:</u> הכנסת ביטים לרגיסטר **בצורה טורית** הינה הכנסת ביטים לרגיסטר בצורה שבו אנו מכניסים באמצעות קו-כניסה בודד(בצורה דומא ניתן להגדיר יציאה טורית).ואנו מוציאים ביט-ביט אחרי זמן קבוע.

לדוגמא: **shift register 3bit**(נקרא בעברית "אוגר הזזה") אשר מקבל 3 ביטים.וננית שהregister מקבל ביט במחזור מספר t . אזי הרגיסטר יוציא אותו במחזור שעון מס t+3 .



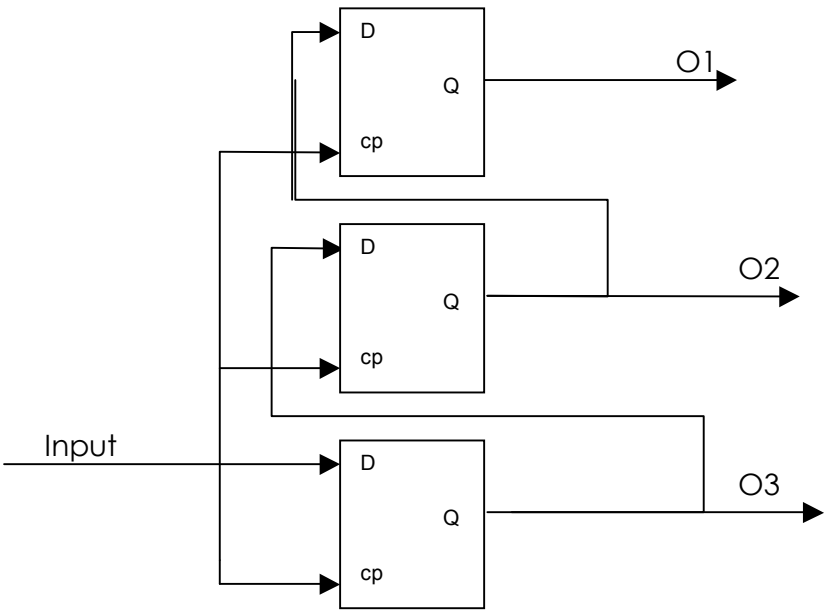
נשים לב כי גם הכניסה וגם היציאה היא טורית

נוכל לעשות שהיציאה שלו תהיה מקבילית כך:



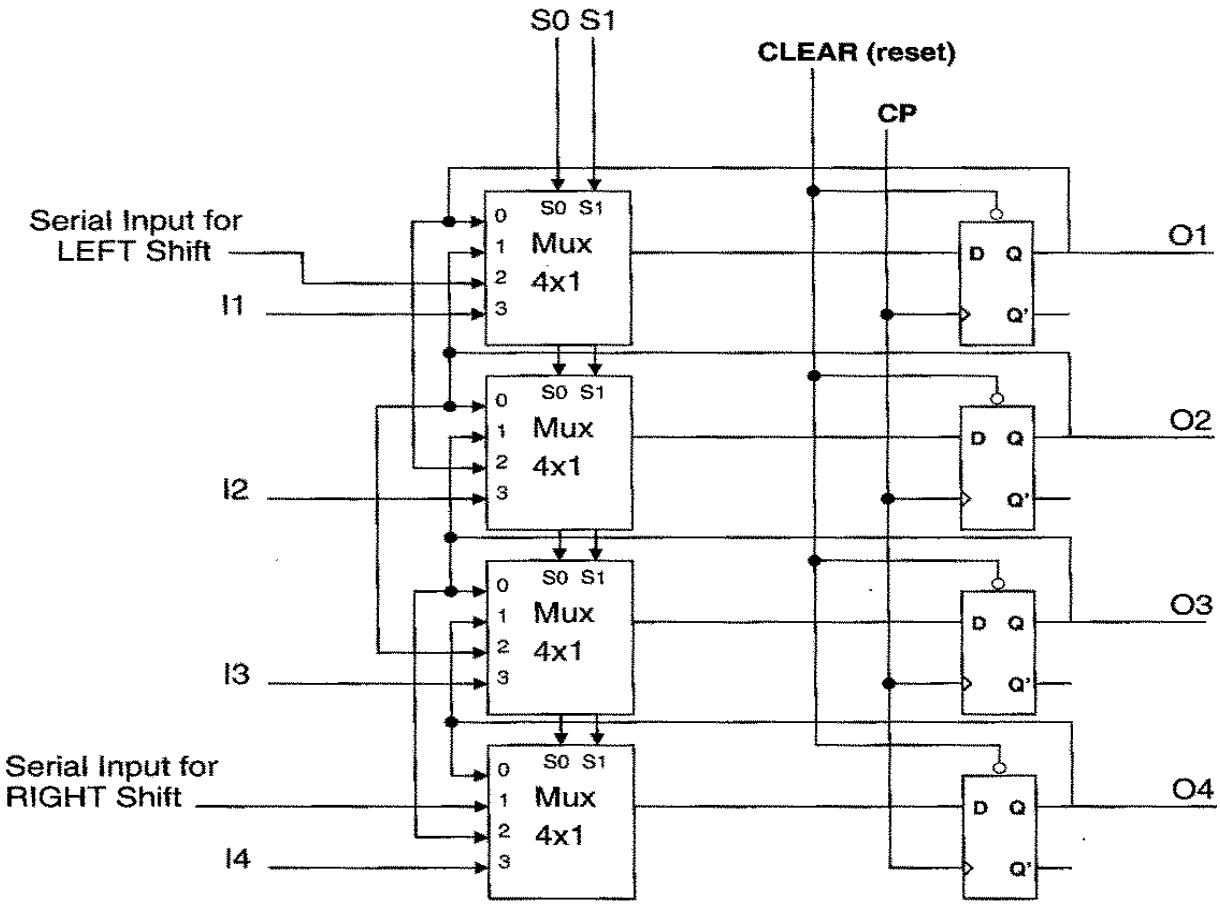
<u>הגדרה</u>: בהינתן שיש לנו רגיסטר ששמור בו מילה של ביטים $O_n O_{n-1} \ldots O_1$ (כלומר ברגיסטר הi היה הערך O_i) פעולת Shift right יהפוך את המילה ל $I O_n O_{n-1} \ldots O_2$ (כלומר ברגיסטר הi היה הערך O_{i-1} וברגיסטר הראשון היה הערך I). בצורה דומה נגדיר Shift left(קוראים לזה גם Serial Shift left כי מבצעים את הפעולה בצורה טורית).

בדוגמא של למעלה,אנו עושים פעולת Shift left ולכן קוראים לו גם **shift left register 3bit**.נוכל לעשות בצורה דומה את **shift right register 3bit** כך:



ניתן גם לעשות פעולת Shift right ציקלית שיהפוך מילה מ $O_1O_nO_{n-1}...O_2$ למילה $O_nO_{n-1}...O_1$

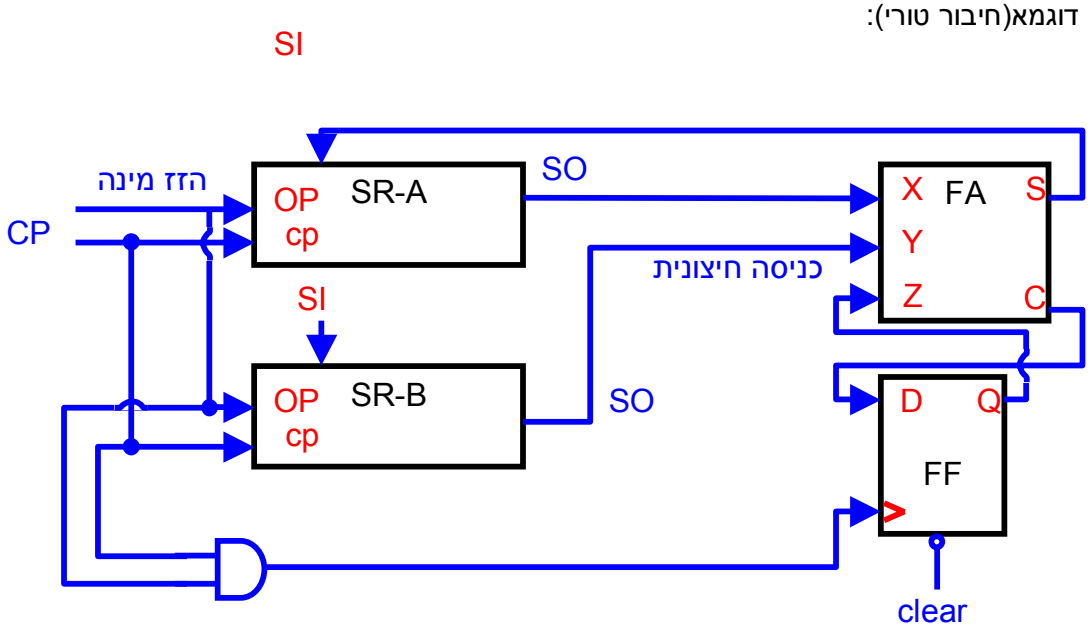
דוגמא(Serial Shift Register with Parallel Load כלומר רגיסטר שמבצע פעולות shift עם אפשרות לכניסה ויציאה מקבילית):



(כל mux שם מקבל את שני הכניסות הבקרה של הmux הקודם)

כאשר המספר של ערך קו הבקרה הוא 0(ז"א s1s0=00) ערך הרגיסטר(ערכי הFF בה) לא משתנה.כאשר המספר של ערך קו הבקרה הוא 1 אנו עושים פעולת Shift right.אם המספר הוא 2 אנו עושים פעולת Shift left .ואם המספר הוא 3 אנו טוענים את 4 הביטים I4I3I2I1.

דוגמא(חיבור טורי):



מחבר בין שני מספרים,שכל מספר הוא ברגיסטר שונה.(OP=1 אזי אנו עושים shift right אם OP=0 אזי אנו לא עושים כלום עם הרגיסטרים,ולכן גם לא מחברים).בשקף 32 מסופר איך לעשות חיבור זה בלי FA

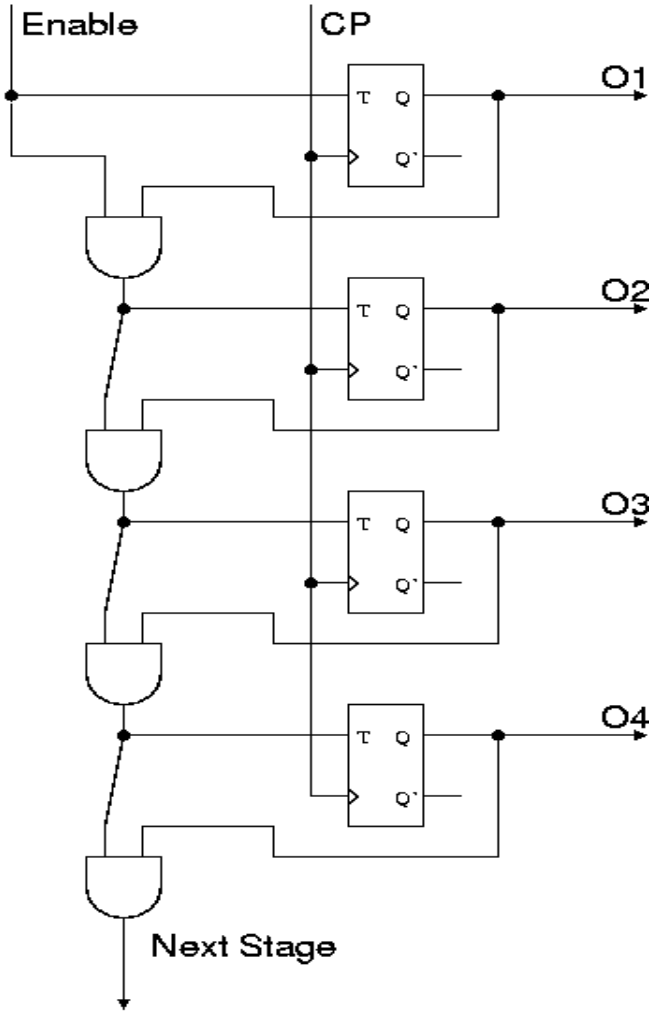
במבחן ב2-3 שנים אחרות היו תמיד שאלות על חיבור אוגרי הזזה!!!

מונים(Counters)

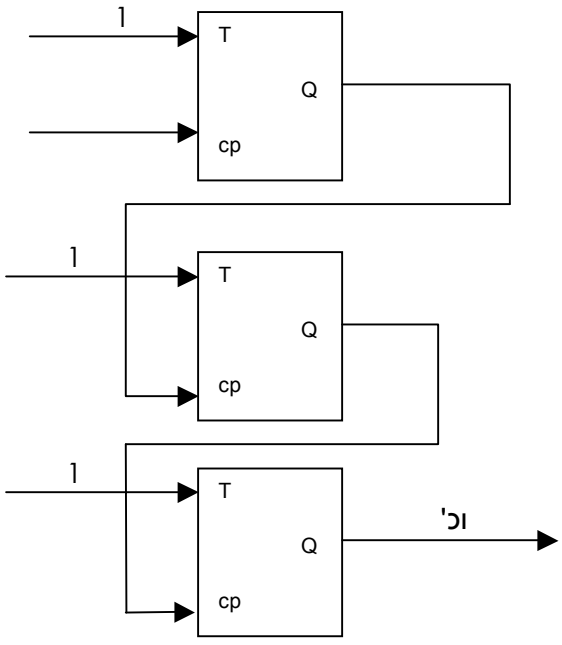
<u>הגדרה:</u> מונה כפי מעלה מודלו-n הוא שיספור לנו(לפי כמות מחזורי השעון) מ0 ועד n-1 וכאשר הוא יגיע לn-1 הוא יספור שוב מ0 ועד n-1 וג'(מונה כלפי מטה ניתן להגדיר בצורה דומה).

שקפים 15-19 מתארים איך להגיע למימוש של מונה up הזה באמצעות הדברים שלמדנו בתירגול " 6,5"(שנעשה בתירגול מספר 9).כמוכן כמות הדלגלגים הדרושה למונה מודלו n הוא logn.ניתן פה רק את המימוש

מונה up:

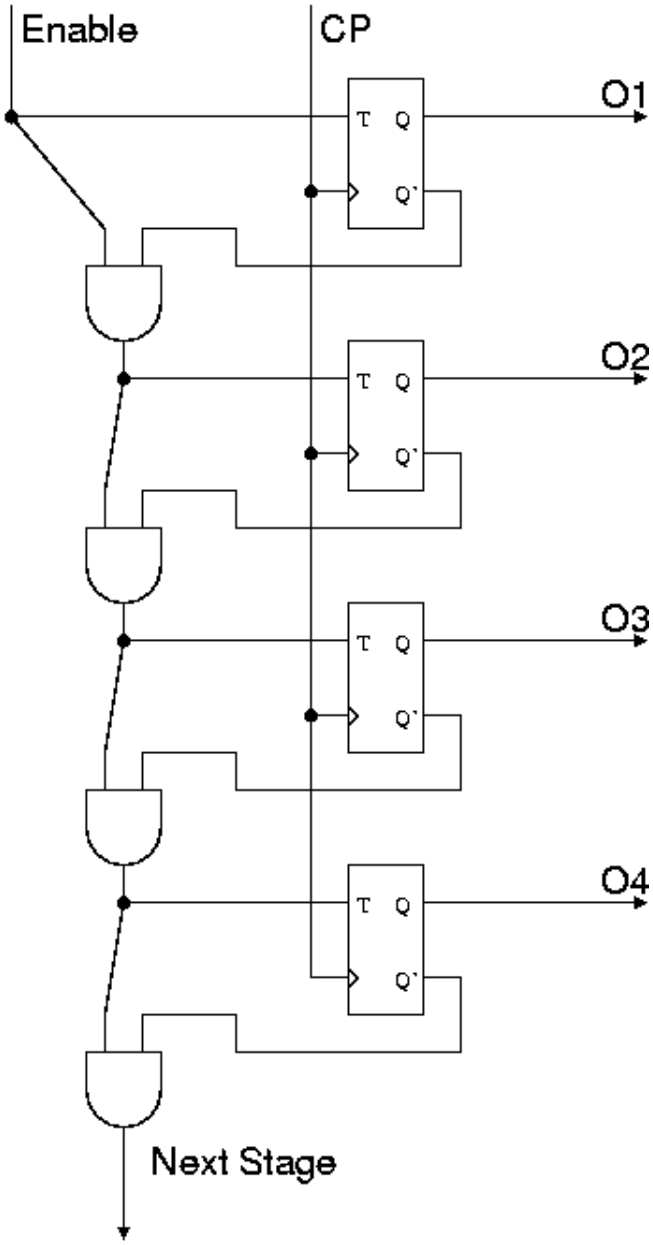


מימוש נאיבי למונה up(כמו שהוסבר בשיעור):

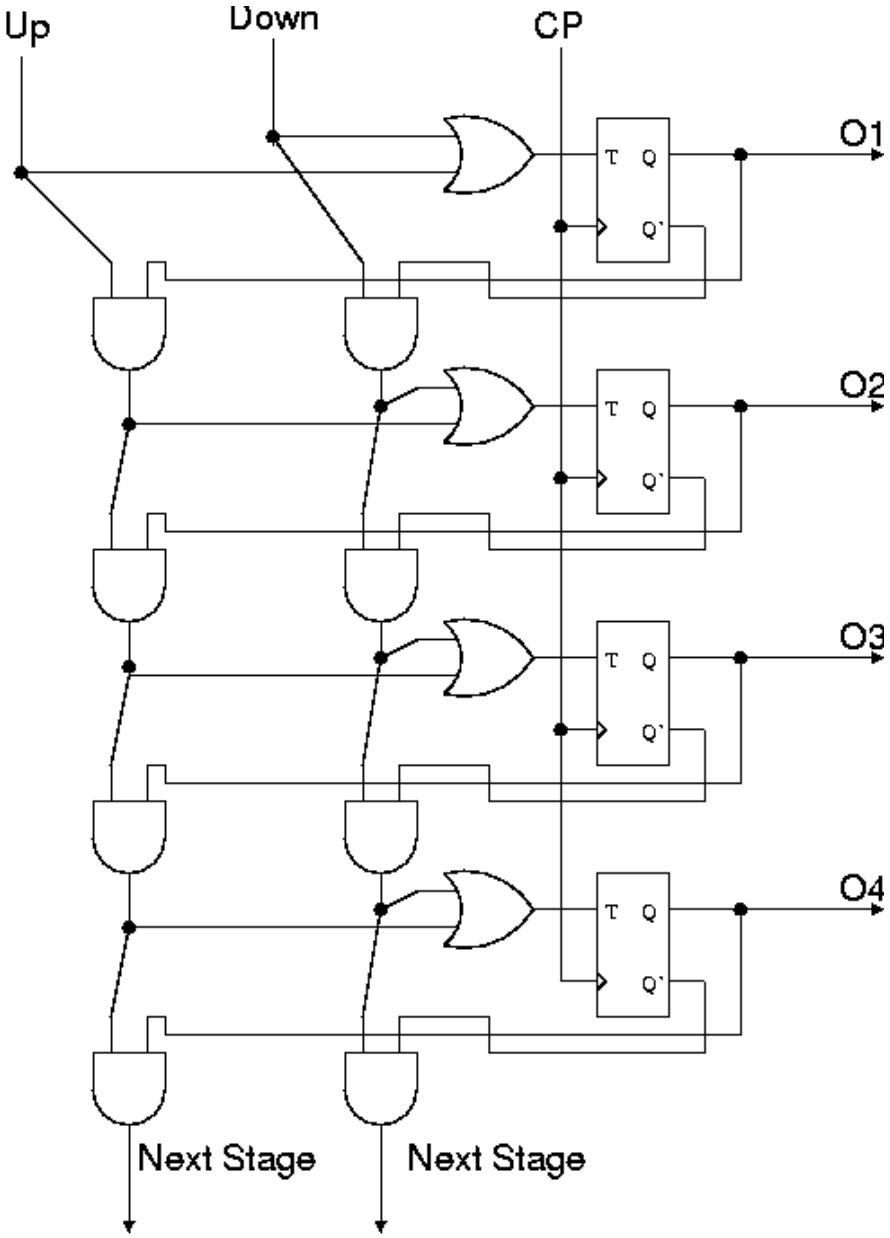


החסרון הבולט פה הוא זמן שלוקח למונה להשתנות,אם לדוגמא ניצור מונה עם הרבה דלגלגים הזמן שיקח לFF להשתנות תהיה מאוד גדולה ואז אם רכיב זה יתחבר לרכיב אחר,ונניח כי זמן השינוי של המונה עולה על יותר מזמן שלוקח מחזור השעון בודד, הרכיב האחר יתכן ויקח ערך זבל.

<u>מונה down:</u>



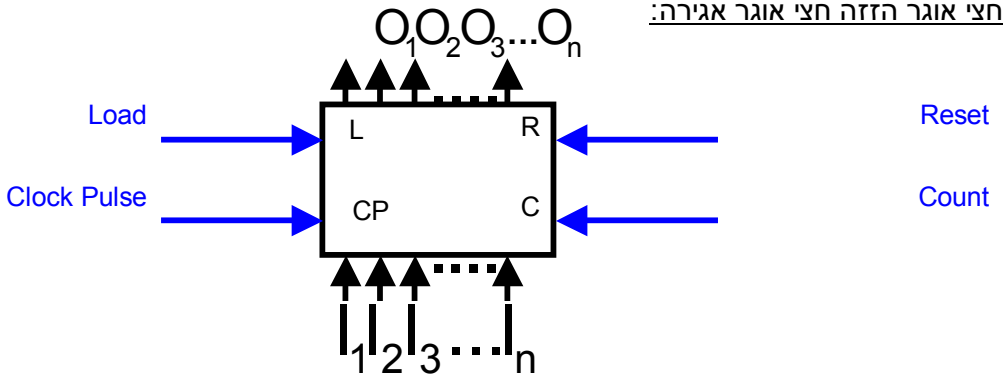
מונה up-down :



כאשר up=0=down אזי המונה נשאר קבוע וכאשר up=1=down נקבל את המילה המשלימה(אם המילה היא 010 אזי תהפוך ל101 וג').

בהרצאה הוא עשה מונה up-down בצורה שונה(יש mux עם קו בקרה s שיגיד לנו האם הכניסה של הFF אמור לקבל את Q או את s=1.Q' אם מעלים את המונה ב1 אם s=1 אנו מורידים את המונה ב1.

חצי אוגר הזזה חצי אוגר אגירה:



והוא תעשה את הפעולות הבאות:

פונקציה/ פעולה	COUNT	LOAD	CP	Reset
מאפס את המונה	∅	∅	∅	0
אין שינוי	0	0	∅	1
טען כניסות לתוך הרגיסטר	∅	1	השעון בעליה	1
מנה – עבור למצב בינארי הבא	1	0	השעון בעליה	1

תרגיל:לעשות את מימוש הזה (רמז:הניחו כי הדלגלגים שלכם הם מסוג FF ותעשו מפות קרנו ל Reset וLoad) .ניתן לממש כל מיני דברים בעזרת רכיב זה.ראה עמוד 28.