

## Регистри и бројачи

Доц. д-р Никола Рендевски

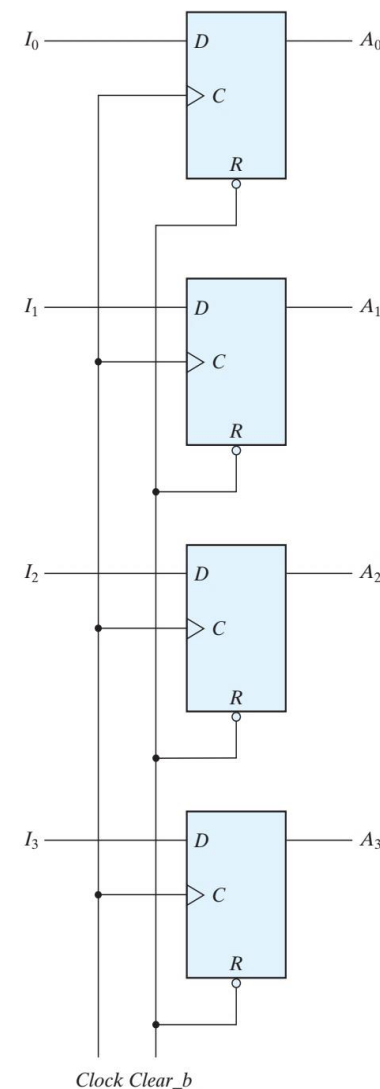
*[nikola.rendevski@fikt.edu.mk](mailto:nikola.rendevski@fikt.edu.mk)*

# Регистри

- Тактираните секвенцијални кола се составени од ФФ и комбинаторна логика
- Дигитално коло составено и само од ФФ се смета за секвенцијално коло
- Дигиталните секвенцијални кола со ФФ се класифицирани според функцијата која ја изведуваат.
- Две класи на секвенцијални кола се
  - РЕГИСТРИ (Registers), и
  - БРОЈАЧИ (Counters)
- регистарот претставува група на ФФ кои делат ист такт сигнал
- n-битен register се состои од n ФФ и има можност да складира информација од n битови
- регистарот се состои од ФФ кои ја чуваат бинарната информација и логички порти кои определуваат како бинарната информација се носи до и излегува од регистарот
- Бројачот во основа е регистар кој поминува низ предефинирана секвенца од бинарни состојби
- Портите во бројачот се поврзани на начин да ја обезбедат пропишаната низа на состојби
- Бројачите се специјален тип на регистри

# Регистри

- Во дигиталната технологија постојат различни изведби на регистри
- „Најпростиот“ (најосновниот) тип на регистар е составен само од ФФ без други логички порти
- Пример на регистар од D-FF
- Заедничкиот такт сигнал ги активира сите ФФ на растечки раб (растечка ивица) и бинарниот податок кој се наоѓа на четирите влезови се пренесува во регистарот
- Четирите излези може да се прочитаат во било кој момент за да се добие складираниот бинарен податок
- Влезот Clear\_b влегува во Reset влезовите за празнење, активни на ниско (во случајот на сл.)
- Clear\_b се користи за празнење на регистарот на 0 при што сите ФФ асинхроно се празнат
- За време на нормалната тактирана работа на рег. R влезовите мора да се држат на логичка 1 во случајот

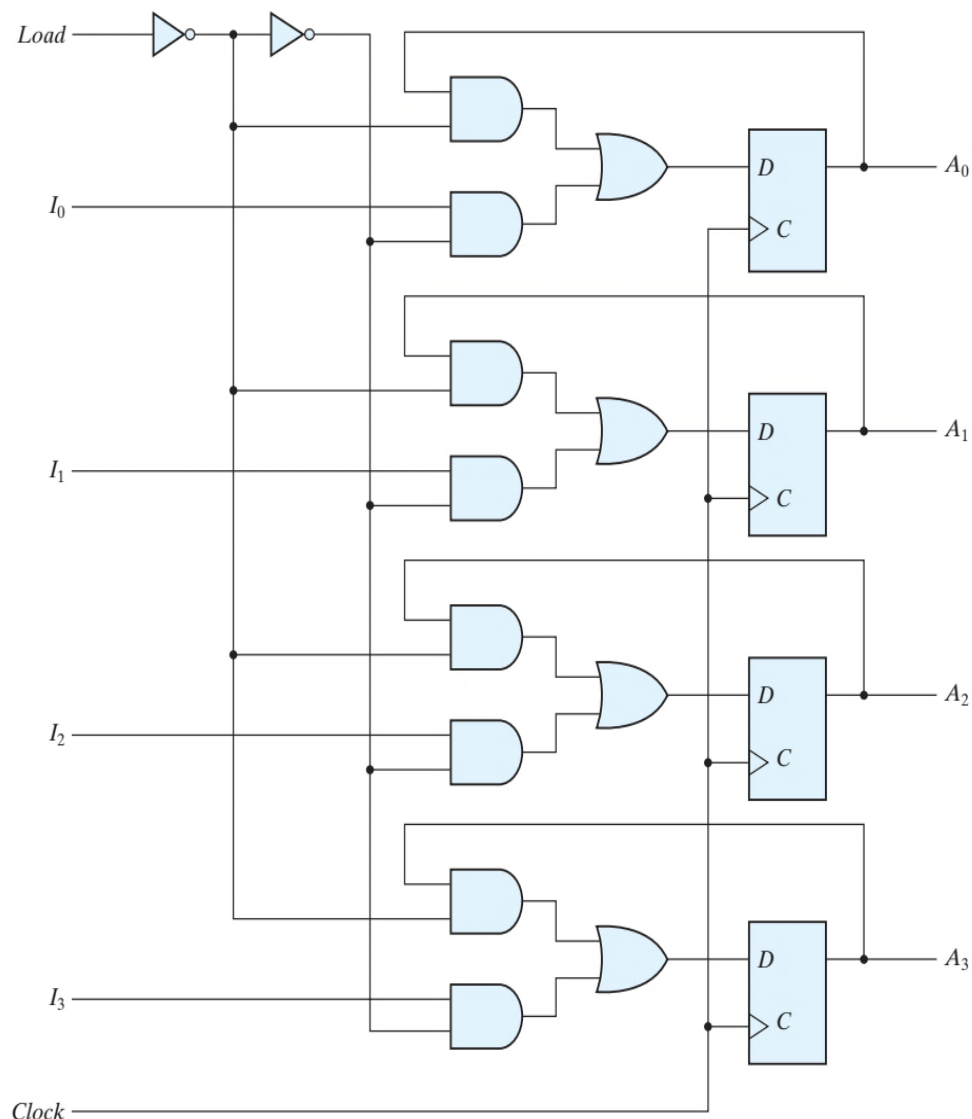


# Регистар со паралелно полнење

- Register with Parallel Load
- Синхроните дигитални системи имаат еден главен такт кој овозможува континуирана низа на такт импулси
- Дали операциите на регистарот ќе се извршуваат на секој такт се дефинира со користење на посебен контролен сигнал
- Пренесувањето на нова информација во регистарот се нарекува полнење (Load) т.е. Ажурирање
- Ако сите битови во регистарот се ажурираат истовремено, со заеднички такт, велиме дека вчитувањето се врши паралелно
- Во ваква конфигурација, доколку содржината на регистарот треба да остане непроменета, влезовите мора да се задржат константни или пак да се отстрани влијанието на такт колото
  - Во првиот случај би требало податочната магистрала која е поврзана со регистарот да биде исклучена/недостапна за пренос кон регистарот
  - Во вториот случај влијанието на такт сигналот да се менува со додавање на дополнителна порта за овозможување и оневозможување
    - Важно! Додавање на дополнителни порти на патеката на такт сигналот НЕ СЕ ПРЕПОРАЧУВА, бидејќи воведува нерамномерно пропагациско доцнење во преносот на импулсите од главниот такт генератор до ФФ

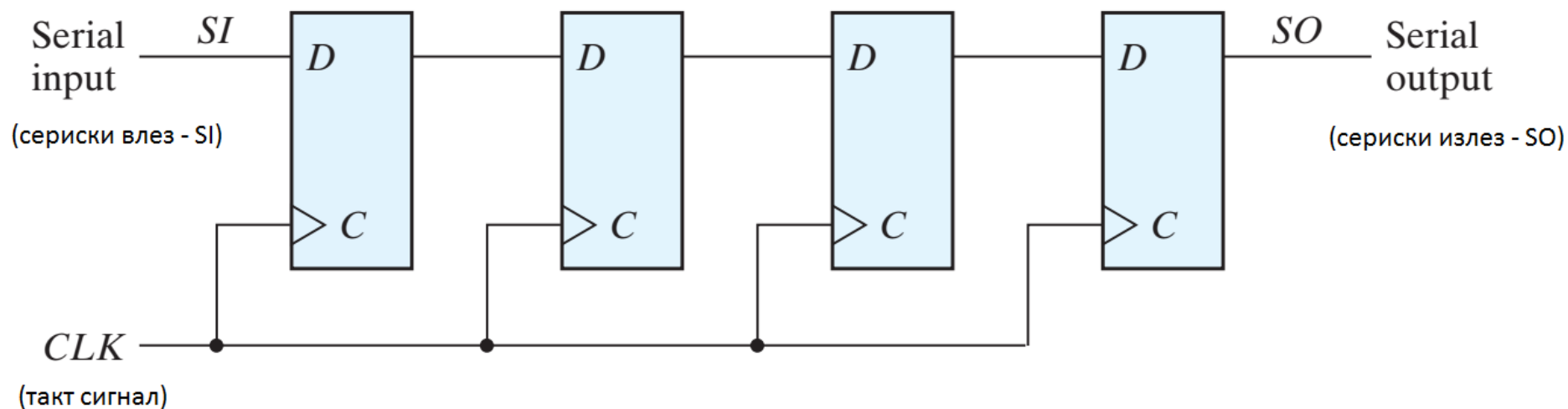
# Регистар со паралелно полнење

- Поради овој факт, се препорачува контрола на операциите на регистарот преку влезовите (D влезовите во случајот), наместо контрола преку тактот
- На сликата е прикажан 4-битен регистар со влез за контрола на полнење
- Дополнителните порти играат улога на дво-канален мултиплексер чиј излез одредува дали влезот во регистарот ќе се вчита од податочната магистрала или пак од излезот на регистарот.
- Ако влезот Load = 1, податоците се пренесуваат од податочната магистрала  
Ако Load = 0, излезот од ФФ се поврзува со соодветниот влез
- Ваквата повратна врска е неопходна бидејќи D-ФФ немаат можност да останат во истата состојба сами
- Со секоја ивица на тактот, влезот D ја одредува следната состојба
- Влезот Load, одредува дали со следниот такт ќе биде прифатена новата или тековната состојба на регистарот
- Преносот на информации од податочните влезови или од излезите се прави симултано односно сите четири бита се пренесуваат во исто време како резултат на такт импулсот



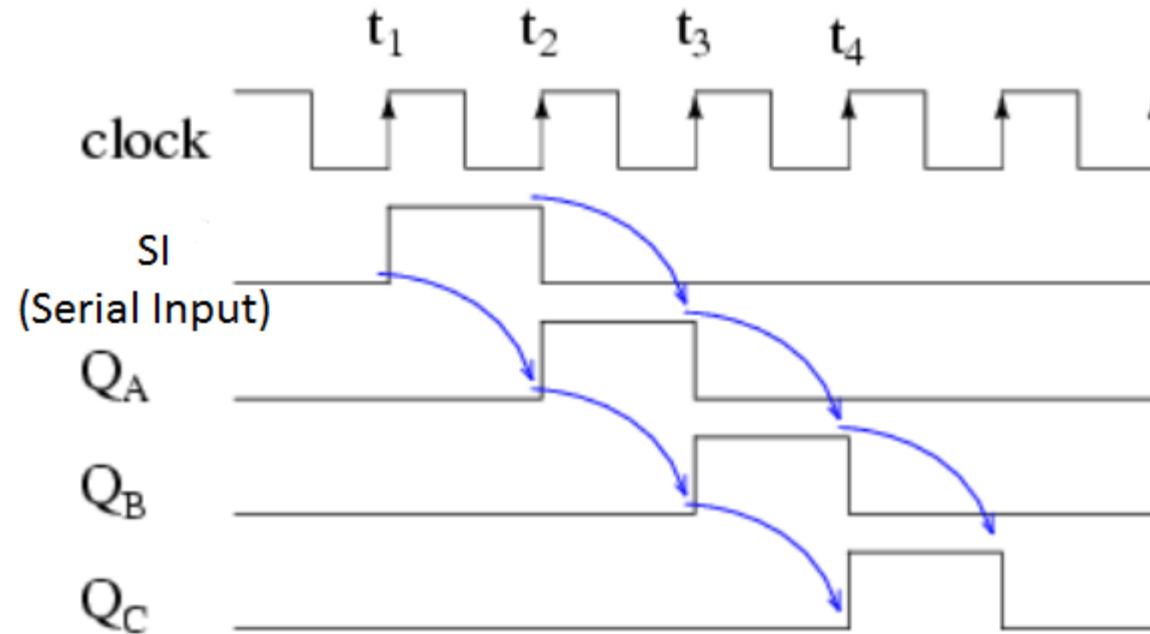
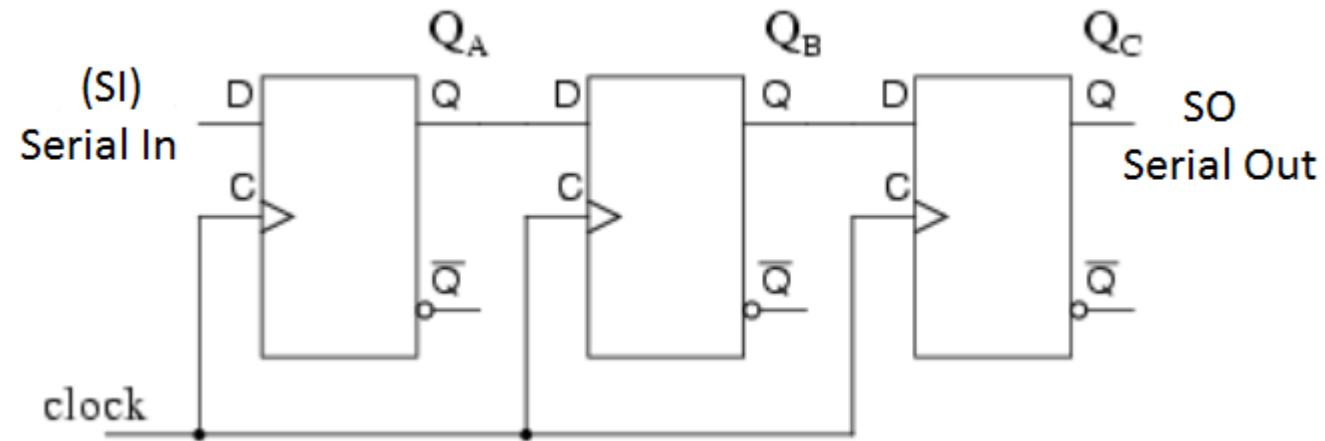
# Регистри со поместување (Shift Registers)

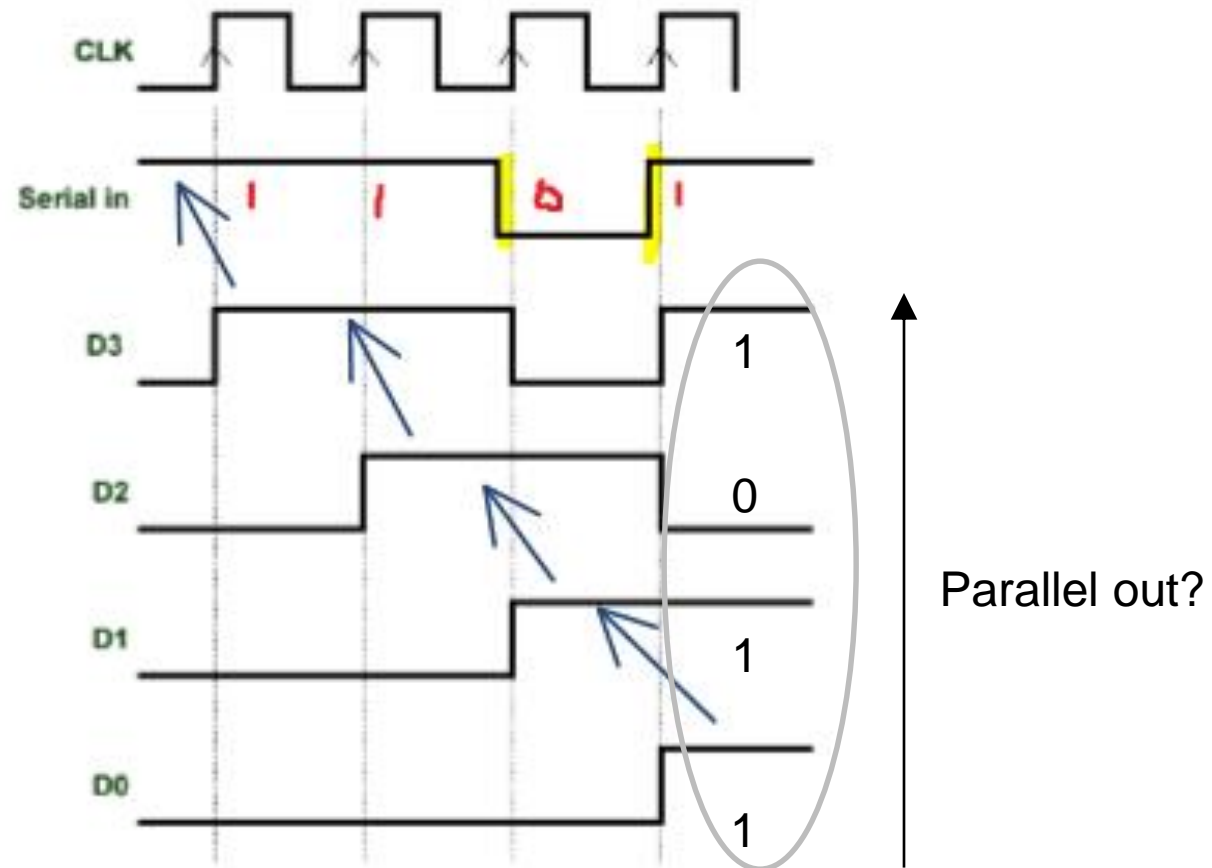
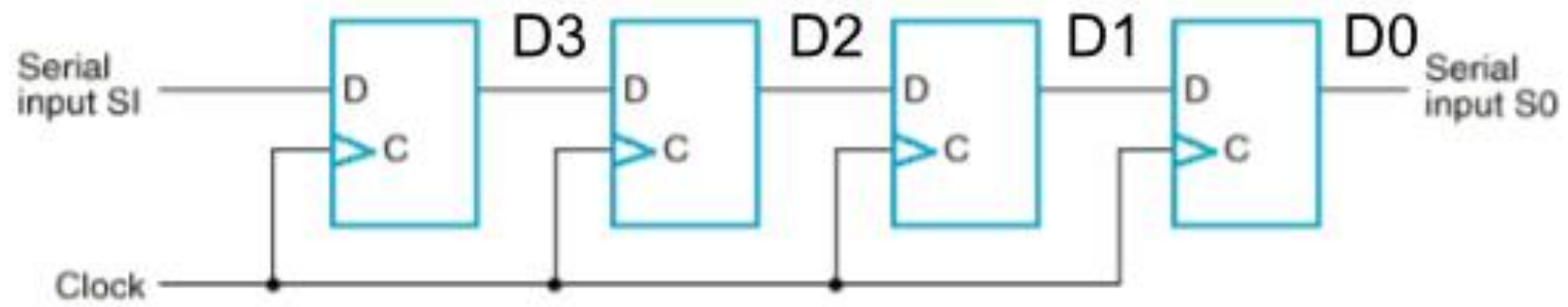
- Регистарот кој ги пренесува бинарните информации од секоја ќелија во соседната ќелија во одбрана насока се нарекува Shift Register
- Логичката архитектура на овој тип регистар е составена од низа (синџир) на ФФ каскадно поврзани, при што излезот од еден ФФ е поврзан со влезот од следниот ФФ. Сеуште сите ФФ имаат заеднички такт сигнал кој го активира поместувањето на податоци поместувањето на податоци (битови)



- Овој регистар е еднонасочен. Секој такт импулс ја поместува содржината на регистарот за една бит позиција во десно
- Серискиот влез определува што се вчитува во ФФ најлево, серискиот излез е излезот на ФФ најдесно
- Исто како и во претходниот случај, може да се додаде механизам за контрола

# Serial In Serial Out (SISO)





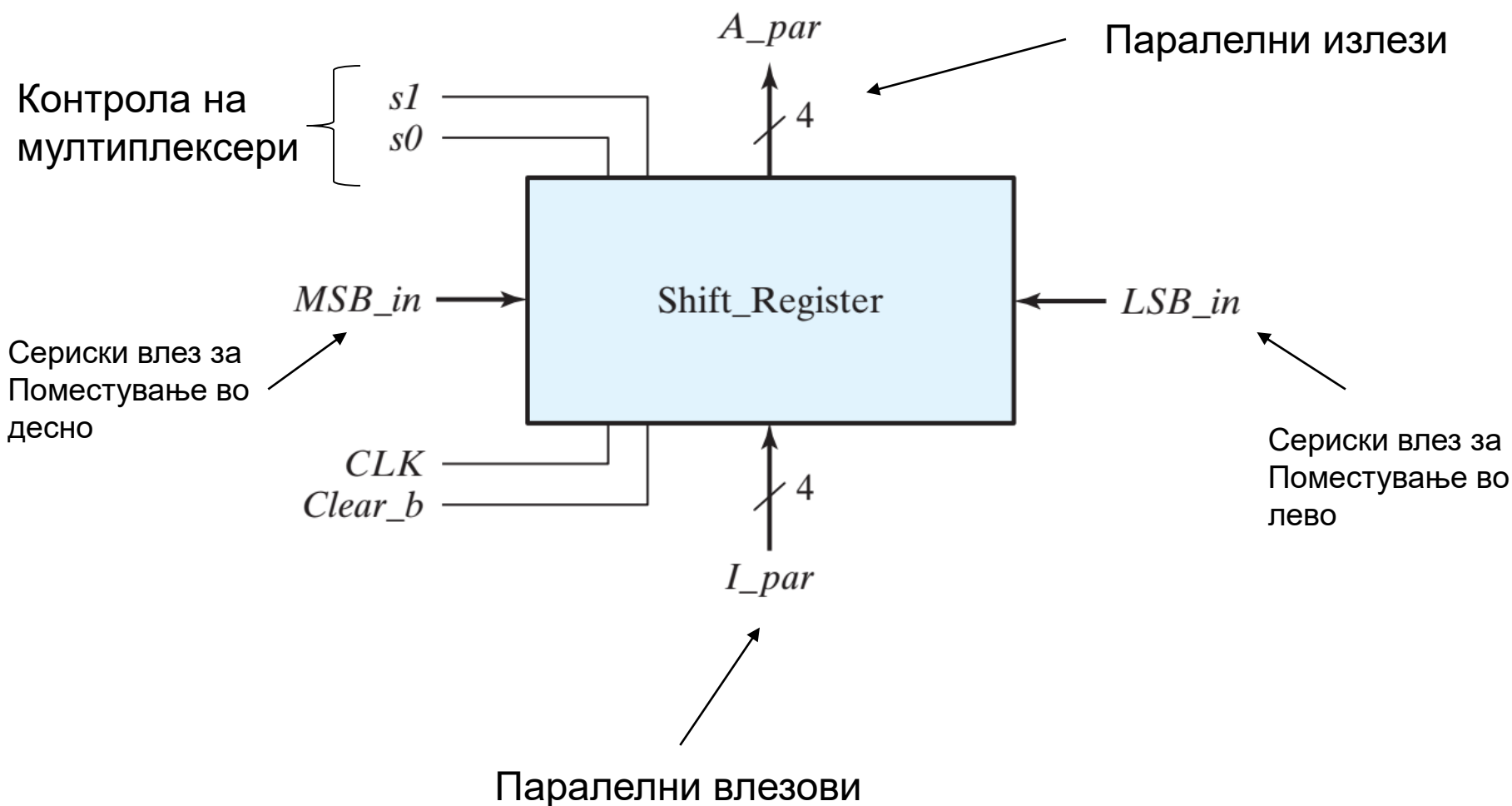


# Универзални регистри со поместување

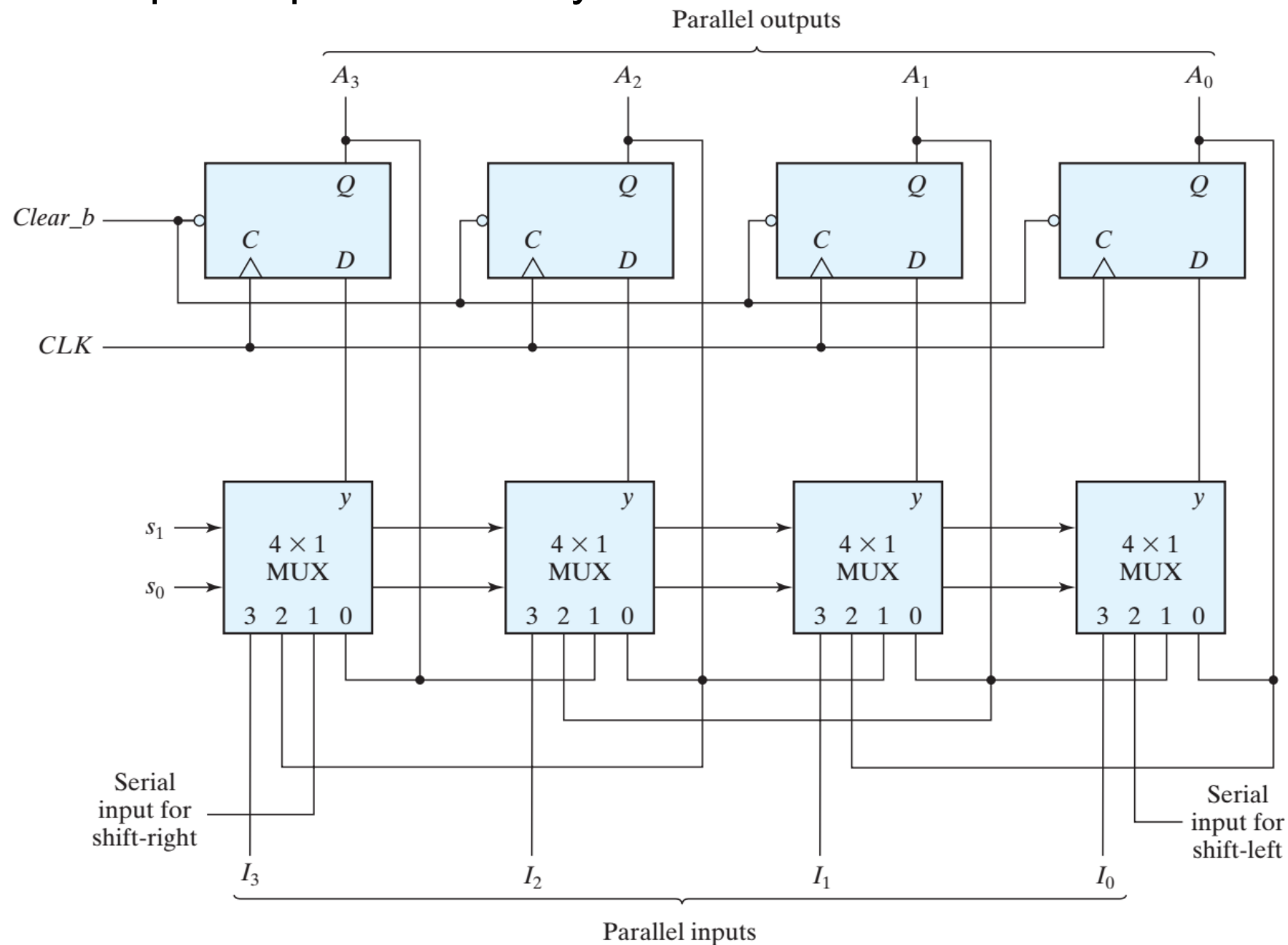
- Доколку се достапни излезите на ФФ на даден регистар со поместување (Shift Register), информациите кои влегуваат сериски со поместување можат да се користат паралелно преку излезите на ФФ од регистарот
- Ако на регистарот со поместување му се даде можност за паралелно полнење, тогаш податоците кои влегуваат паралелно можат да се користат сериски со поместување
- Во пракса, регистрите можат да имаат способност за поместување (shift) во две насоки (налево и надесно)
- Најчесто shift регистрите ги имаат следните можности
  - Контрола за празнење (Clear)
  - Такт влез (Clock)
  - Влез за контрола на насока (лево, десно) – shift-left, shift-right (влезови)
  - Контрола за паралелно полнење (parallel-load)
  - Паралелни излезни линии
  - Контрола за заштита на информациите од промени и покрај активноста на такт сигналот
- Некои регистри во пракса може да ги имаат само дел од претходно споменатите можности
- Доколку регистарот може да се поместува во две насоки и има можност за паралелно полнење, тогаш се нарекува универзален регистар

# Универзални регистри со поместување

- Доколку се достапни излезите на ФФ на даден регистар со поместување (Shift Register), информациите кои влегуваат сериски со поместување можат да се користат паралелно преку излезите на ФФ од регистарот



# Универзални регистри со поместување



# Универзални регистри со поместување

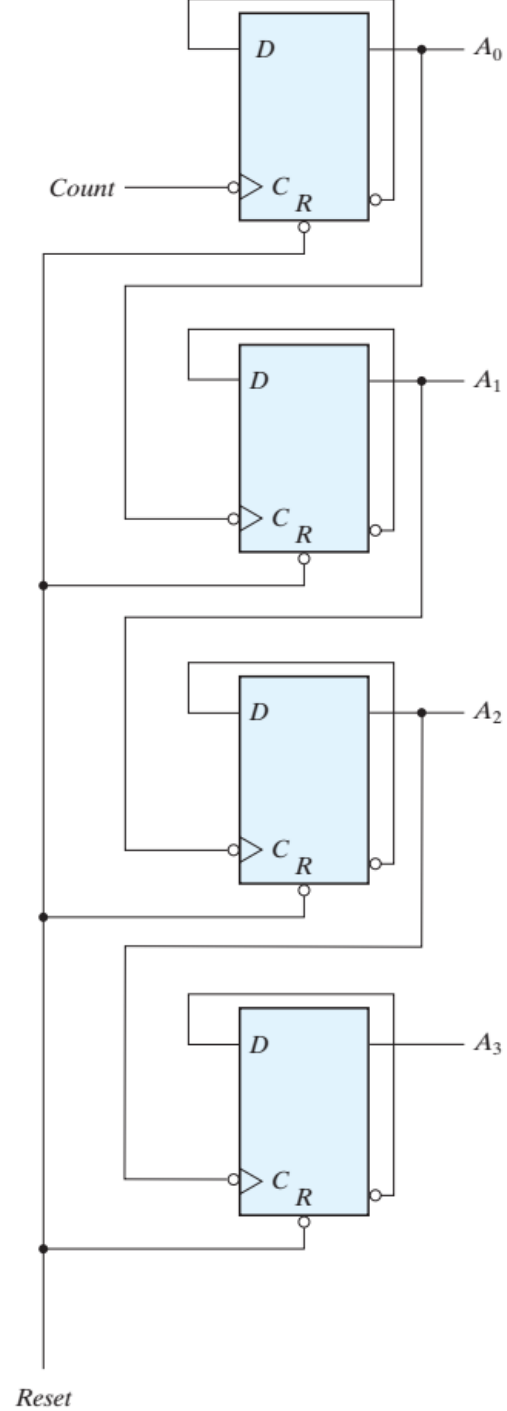
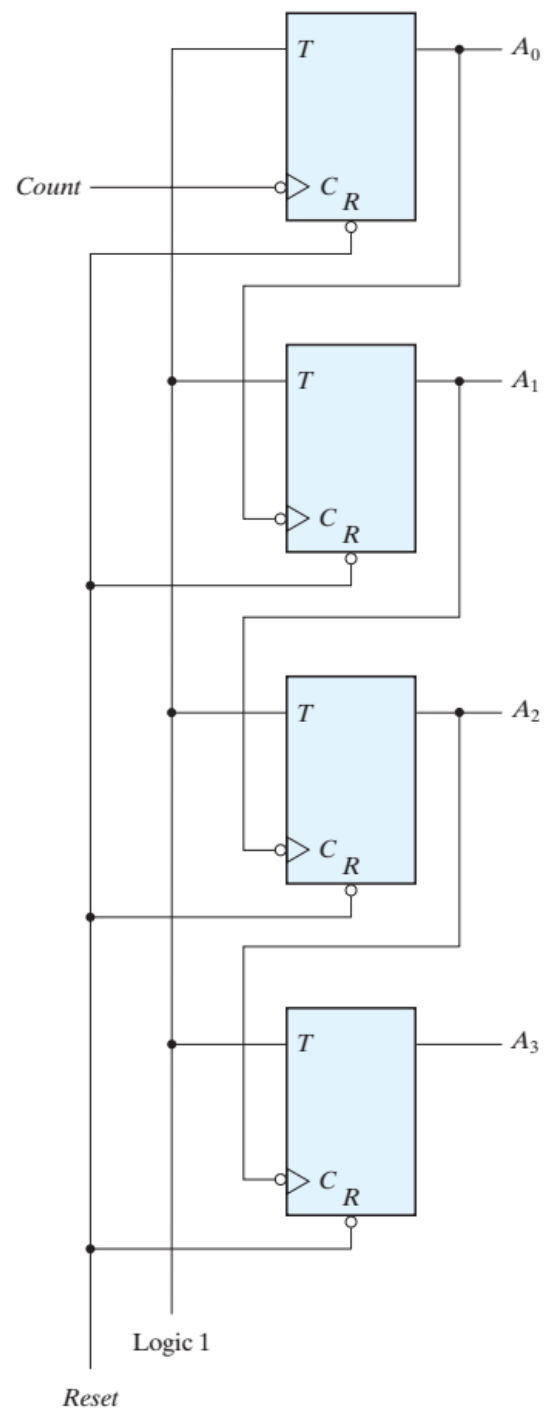
- Колото се состои од 4 ФФ и 4 мултиплексери
- Мултиплексерите имаат два два заеднички влеза за избирање  $s_1$  и  $s_0$
- Влезот 0 кај секој MUX е избран кога  $s_1=0$ ,  $s_0=0$ , а влезот 1 кога  $s_1=0$ ,  $s_0=1$ , итн. за сите останати влезови
- Кога  **$S_1S_0=00$** , тогаш тековната вредност на ФФ-ите се пренесува на D влезовите што значи дека нема промена (store), бидејќи формирана е патека од излезот и влезот на секој ФФ
- Кога  **$S_1S_0=01$** , тогаш влезот 1 од секој MUX има патека до D влезовите на ФФ. Ова предизвикува операција на поместување на десно со пренос на серискиот влез на A3.
- Кога  **$S_1S_0=10$** , се предизвикува операција на поместување во лево, со пренос за другиот сериски влез во ФФ A0.
- Кога  **$S_1S_0=11$** , бинарната информација на паралелните влезови истовремено се пренесува на во регистарот при следната ивица на тактот (Parellel Load)
- Да забележиме дека податоците влегуваат во MSB\_in при операција на поместување на десно, а влегуваат во LSB\_in при операција на поместување во лево.
- Shift регистрите најчесто се користат за комуникација помеѓу одвоени дигитални системи. Испраќачот го прифаќа n-битниот податок паралелно го сместува во shift регистар, додека приемникот сериски го прима (по една заедничка линија за пренос) и сериски го сместува во друг shift регистер.

# Бројачи (Counters)

- Регистар кој поминува во предефинирана секвенца на состојби при појава на такт ивица се нарекува бројач
- Влезните пулсеви можат да бидат такт (clock) пулсеви или пак да потекнуваат од друг надворешен извор
- Секвенцата на состојби може да следи бинарна секвенца или пак било која друга
- Бројачот кој следи (чији состојби следат) бинарна секвенца се нарекува бинарен бројач
- $n$ -битен бинарен бројач се состои од  $n$  ФФ, и може да брои од 0 до  $2^n - 1$
- Бројачите се класифицирани во две категории и тоа
  - **Бројачи со прелевање (Ripple Counters) и**
  - **Синхрони бројачи**
- Кај бројачите со прелевање, транзицијата (преминот) на состојби на излезот на ФФ служи како влез за активирање (triggering) на други ФФ. Со други зборови, влезот за такт на некои FF не е заеднички такт сигнал, туку излезот од некој друг ФФ
- Кај синхроните бројачи, влезот за такт е заеднички за сите ФФ
- За објаснувања на работата на бројачите со прелевање, ќе се задржиме на бинарни и BCD бројачи

# Бројачи со прелевање (Ripple Counters)

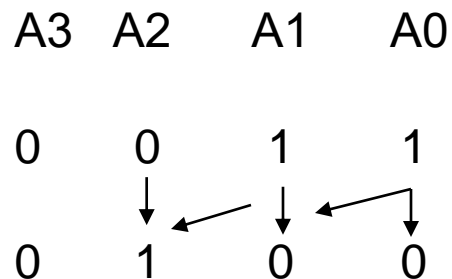
- Бинарниот бројач со прелевање се состои од сериска врска од комплементиращки ФФ при што излезот на секој ФФ оди како влез во такт влезот на ФФ кој ја претставува позицијата од повисок ред (бројач нагоре)
- Такт сигналот (clock) се носи на ФФ кој ја презентира LSB позицијата
- Комплементиращки ФФ може да се добие од JK ФФ при што J и K влезовите ќе се поврзат заедно (ќе се премостат) или со T ФФ
- Важно!!! Научете ги карактеристичните равенки и табелите на сите типови ФФ за да можете да реализирате секвенцијална логика со различни ФФ
- Трета можност за реализација на комплементиращки ФФ е со примена на D ФФ, чиј што комплементиран излез за тековната состојба е поврзан на влезот D
- На овој начин D влезот е секогаш комплемент на тековната состојба, и следниот такт ќе предизвика комплементирање на флип-флопот
- Логичките дијаграми на два 4-битни бројачи со прелевање е прикажан на следната слика
- Изведба со D и T ФФ
- Влезовите T кај секој ФФ се поврзани на логичка 1, што создава комплемент кај секој од ФФ кога такт сигналот има опаѓачка ивица (Слика)
- Кругчето пред симболот за Clock (C) означува дека реагираат на опаѓачка ивица



# Бинарен бројач со прелевање

- Да ги погледнеме првите девет бинарни броеви
- LSB битот  $A_0$ , се комплементира при секој такт
- Секогаш кога  $A_0$  се менува од 1 во 0, се комплементира  $A_1$  (опаѓачка ивица)
- Секогаш кога  $A_1$  се менува од 1 во 0, се комплементира  $A_2$ , со  $A_2$  и  $A_3$ , со  $A_3$  и  $A_4$
- Да го анализираме преминот од 0011 во 0100
- Бидејќи  $A_0$  се менува од 1 во 0, тоа го комплементира  $A_1$ , и тој се комплементира односно се менува од 1 во 0. Тоа пак го активира  $A_2$  и тој се комплементира од 0 во 1. Оваа промена на  $A_2$  е растечка (0 во 1) и нема да го промени  $A_3$  кој ќе остане 0 бидејќи ФФ е активен на опаѓачка.

$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0



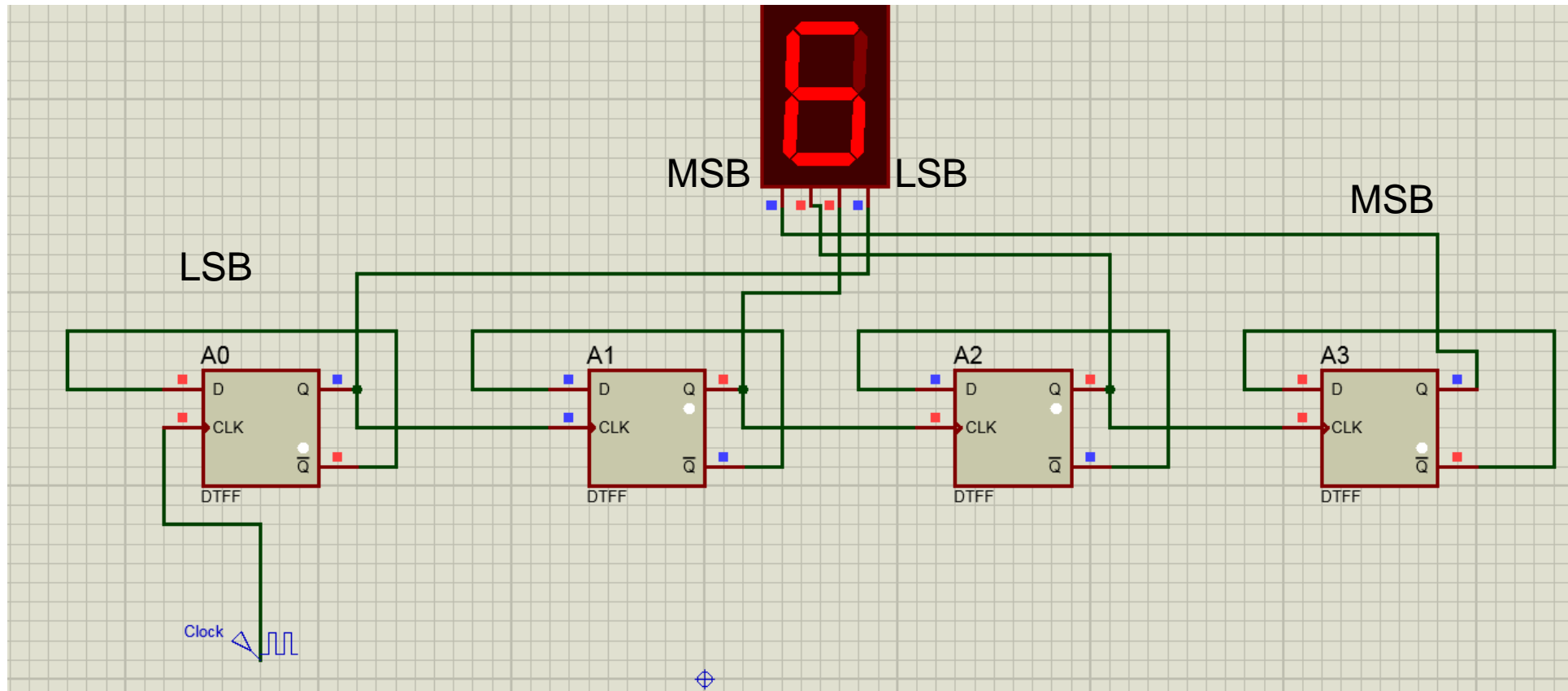


# Бинарен бројач што одбројува

- Доколку бинарниот бројач брои во опаѓачки редослед, истиот се вика (binary countdown counter)
- Кај ваков бројач бинарниот број се намалува за 1 во секој циклус
- Исто така, LSB битот се комплументира во секој такт импулс, а секој друг бит се менува доколку неговиот претходен се менува од 0 во 1. Оттука дујаграмот на ваков бројач ќе изгледа исто како и дијаграмот на бинарен бројач, само што сите ФФ се активираат на растечка ивица на такт сигналот.
- Доколку пак мора да користиме ФФ кои се активни на опаѓачка ивица, тогаш влезот за такт сигналот треба да се поврзи со комплументираниот излез од претходниот ФФ. Тогаш, кога основниот излез Q ќе премине од 0 во 1, комплументираниот ќе премине од 1 во 0 и ќе го активира и комплументира следниот ФФ, што е и потребно.

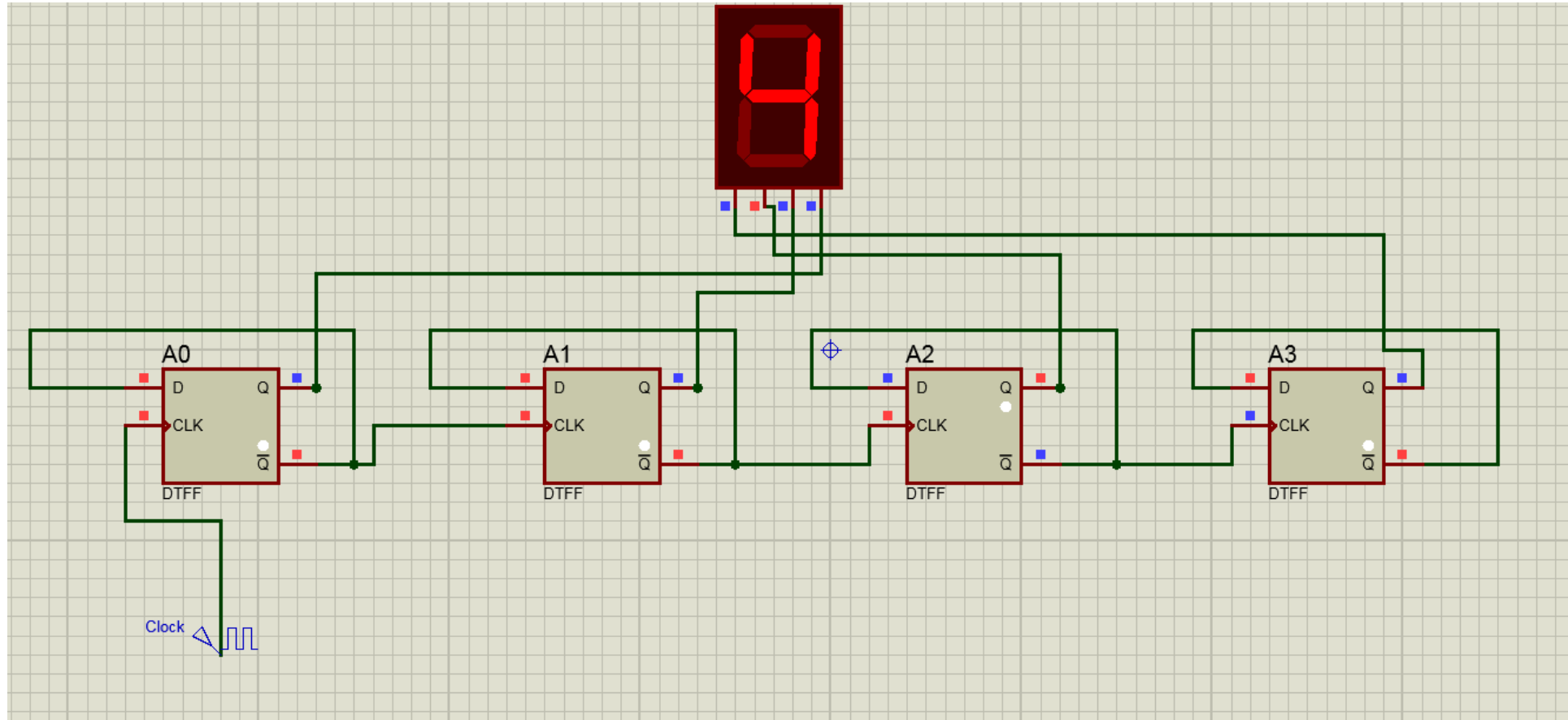
# Бинарен бројач што одбројува

- Да се проба во Proteus



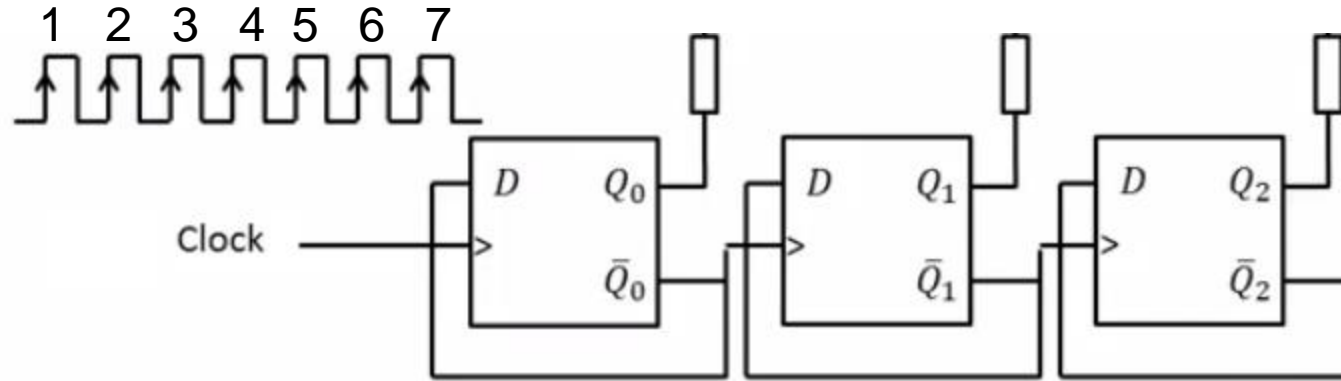
# Бинарен бројач нагоре

- Да се проба во Proteus



# Бинарен бројач нагоре (Да анализираме)

- 0-7 (3 ФФ)



Clock	$Q_0$	$\bar{Q}_0$		$Q_1$	$\bar{Q}_1$		$Q_2$	$\bar{Q}_2$
0	0	1		0	1		0	1
1	1	0		0	1		0	1
2	0	1	┐	1	0		0	1
3	1	0		1	0		0	1
4	0	1	┐	0	1	┐	1	0
5	1	0		0	1		1	0
6	0	1	┐	1	0		1	0
7	1	0		1	0		1	0