

# Увод у организацију и архитектуру рачунара 1

Александар Картељ

[kartelj@matf.bg.ac.rs](mailto:kartelj@matf.bg.ac.rs)

прилагодила: Јована Ковачевић

Напомена: садржај ових слајдова је преузет од проф. Саше Малкова

# Секвенцијалне мреже

Основни појмови

# Појам секвенцијалне мреже

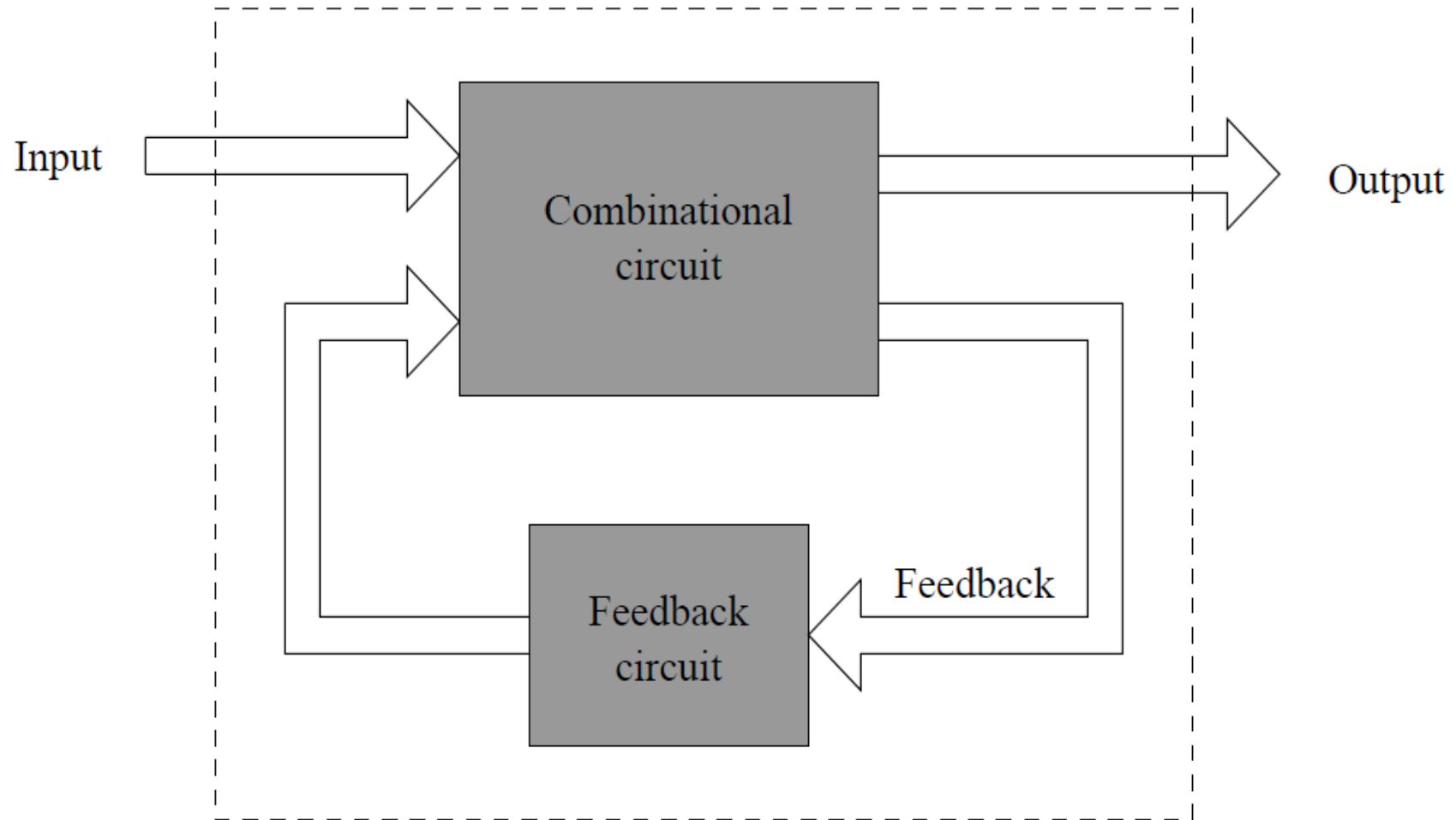
- Секвенцијална мрежа је скуп повезаних логичких елемената чији излаз у неком тренутку зависи од:
  - текућег стања елемената мреже
  - и вредности улаза у “том истом” временском тренутку

# Концепти секвенцијалне мреже

- Текуће стање елемената се читава путем неких излаза мреже
- То практично значи да
  - из секвенцијалне мреже излазе
    - резултати
    - стања елемената апстрахована као “променљиве стања”
  - у секвенцијалну мрежу улазе
    - аргументи
    - стања елемената апстрахована као “променљиве стања”
- Секвенцијална мрежа се може апстраховати као склоп:
  - комбинаторне мреже
  - и мреже која омогућава поновну употребу променљивих стања

# Концептуални дијаграм

Sequential circuit

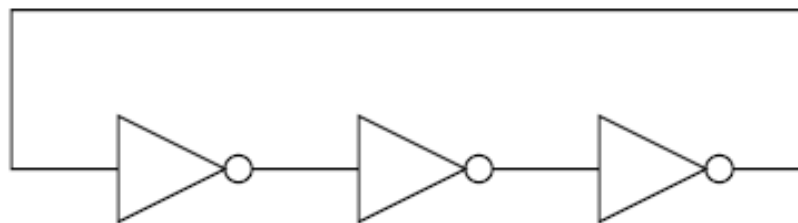
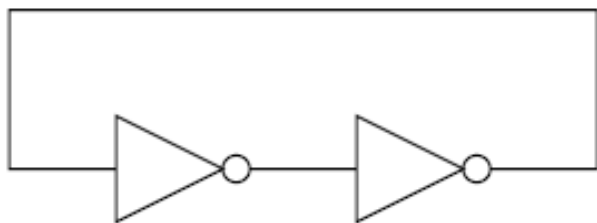


# Стабилност система

- За систем кажемо да је стабилан ако се за непроменљив улаз добијају непроменљиво стање и непроменљив излаз
- Систем није стабилан ако се за непроменљив улаз добијају променљиво стање или променљив излаз

# Стабилност система (2)

- Примери стабилног и нестабилног кола:



# Проблем пројектовања

- У случају комбинаторних мрежа постоји једноставан метод пресликавања логичких функција у кола
  - зато што је систем стабилан (нема повратне спреге као на претх. слици)
  - зато што је детерминистички (у односу на улаз)
- У случају секвенцијалних мрежа проблем је далеко сложенији
  - Пројектовање секвенцијалних мрежа нећемо изучавати!



# Асинхрона и синхрона кола

- Дигитална кола могу да функционишу у два режима:
  - Асинхрони режим
  - Синхрони режим

# Асинхрони режим рада

- Асинхрони режим
  - дигитална кола функционишу независно једна од других
  - тренутак одвијања промена у једном колу не зависи од тренутка одвијања промена у другом
  - нису сви излази и улази активирани у истом тренутку
  - асинхрони рад је проблематичан ако излаз једног кола мора да представља улаз за неко друго коло

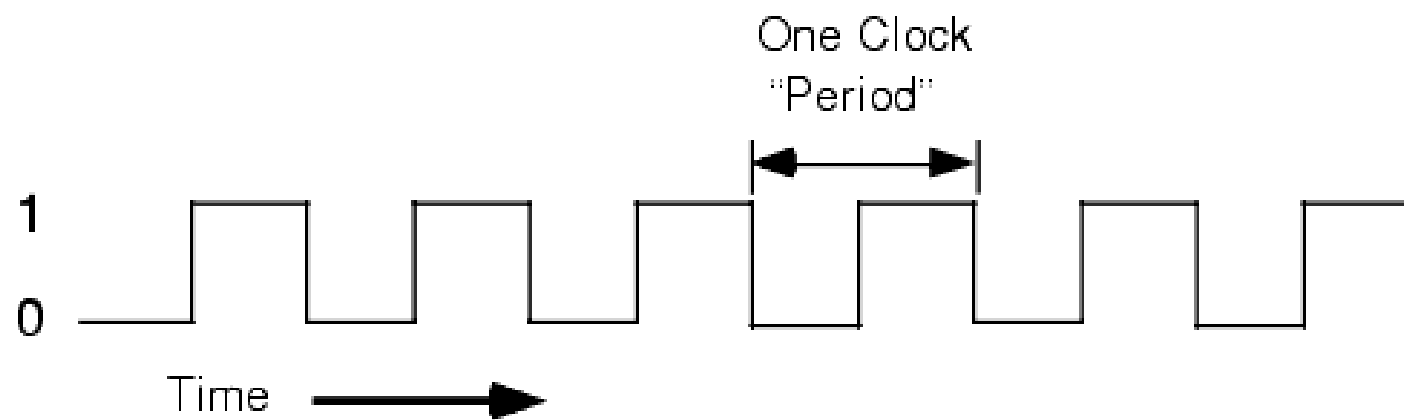
# Синхрони режим рада

- Синхрони режим
  - сва кола у систему мењају своја стања у прецизно дефинисаним тренуцима
  - тренуци промена су одређени сигналом часовника
  - последица је да брзина рада зависи од часовника
    - чак и када би неке операције могле да се заврше раније, мора да се чека на часовник

# Часовник

- Часовник је сигнал који представља секвенцу наизменичних вредности 0 и 1
- Уобичајено се сигнал представља као да се прелазак стања часовника са 0 на 1 (и обратно)
- Овај прелаз се не одвија тренутно, има неко кратко трајање
  - Период (или тренутак) мењања стања се назива “руб”
  - Прелазак са 0 на 1 се назива “узлазни руб” (или “руб успона”, “позитиван руб”) (енгл. *rising edge*)
  - Прелазак са 1 на 0 се назива “силазни руб” (или “руб силаска”, “негативан руб”) (енгл. *falling edge*)

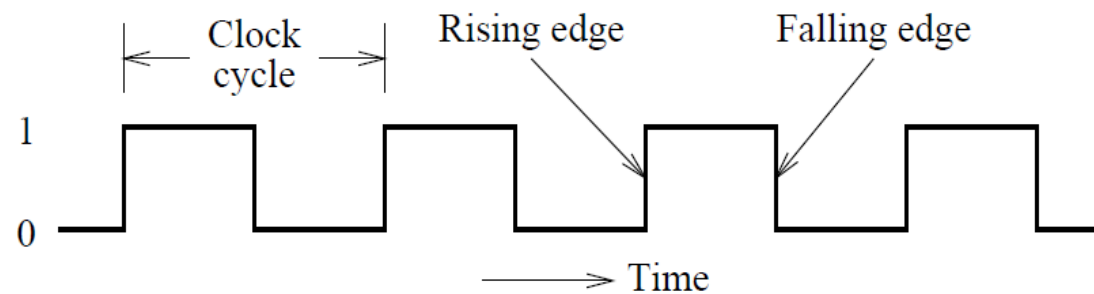
# Часовник (2)



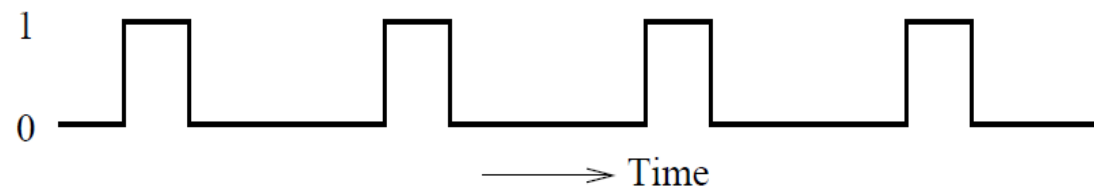
# Трајање стања сигнала

- Уобичајено је да трајање сваког непроменљивог стања сигнала часовника буде једнако (тзв. *симетричан часовник*)
- Међутим, може се употребљавати и часовник код кога трајање стања 0 и 1 није једнако (тзв. *асиметричан часовник*)
- “Циклус часовника” је период између два узастопна узлазна (или силазна) руба
- “Брзина часовника” је број циклуса у секунди и изражава се у Херцима (Hz)
  - на пример, ако је брзина часовника 100MHz, тада је трајање једног циклуса 10ns

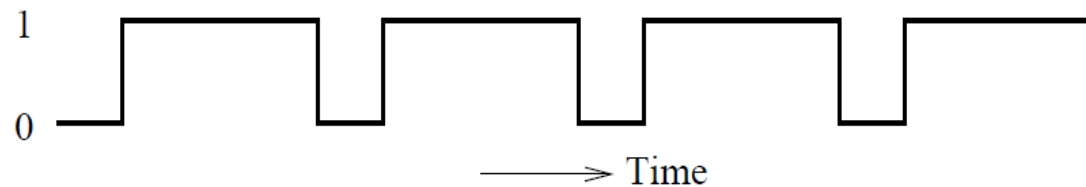
# Типови часовника



(a) Symmetric



(b) Smaller ON period



(c) Smaller OFF period

# Улоге часовника

- Основна улога часовника је глобална синхронизација сигнала у систему
- Друга улога часовника је мерење времена у облику броја циклуса



# Секвенцијалне мреже

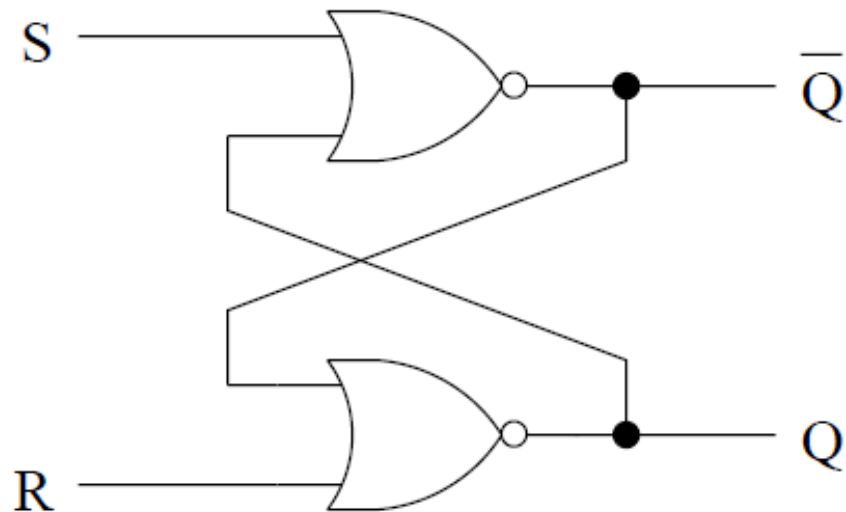
Основни елементи за памћење стања: резе и флип-флопови

# Елементарне секвенцијалне мреже

- Постоје две врсте елементарних секвенцијалних мрежа:
  - реза (енгл. *latch*)
    - коло које реагује на ниво сигнала, без обзира на тип промене
    - чува 1 бит стања
  - флип-флоп (енгл. *flip-flop*)
    - коло које реагује само на промене само на узлазном или само на силазном рубу циклуса
    - чува 1 бит стања
- Подела је релативно нова, па се негде употребљава само један од ових назива (обично *флип-флоп*) за обе врсте елемената

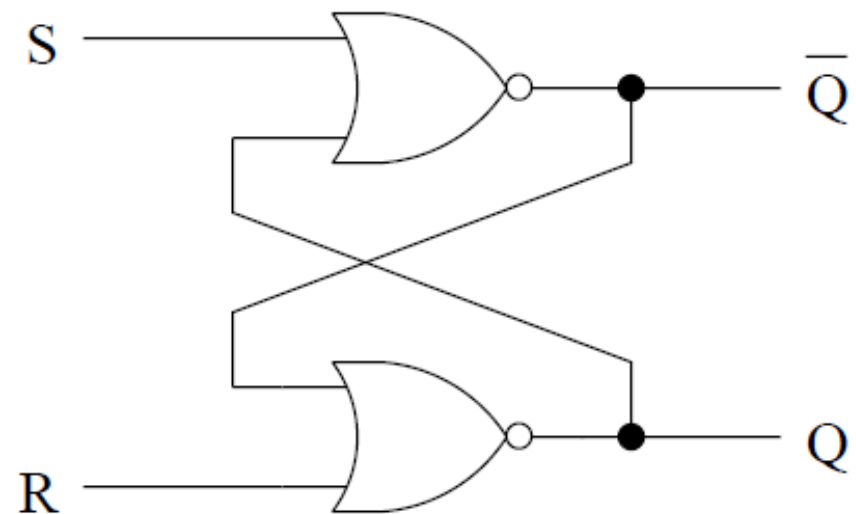
# SR реза

- Назива се и *SR* елемент (енгл. *SR latch*, *SR* потиче од *set-reset*)
  - користи се и назив *RS* елемент
- Има два улаза *S* и *R* и два излаза *Q* и  $\bar{Q}$
- Имплементира се помоћу два НИЛИ елемента:



# SR реза – анализа понашања

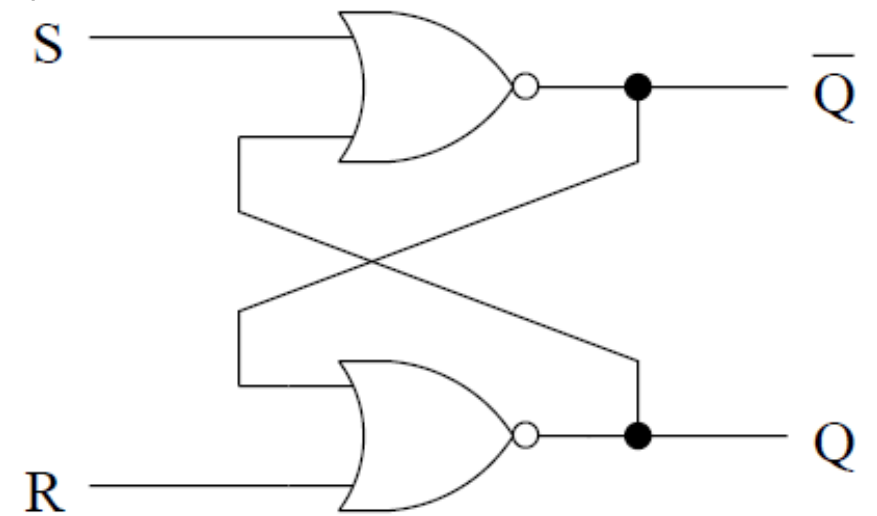
- $S=0, R=1$ 
  - због  $R=1$  мора да буде  $Q=0$
  - због  $S=Q=0$  се добија  $Q'=1$
- $S=1, R=0$ 
  - због  $S=1$  мора да буде  $Q'=0$
  - због  $R=Q'=0$  се добија  $Q=1$
- $S=0, R=0$ 
  - ако је претходно било  $Q=0, Q'=1$ 
    - онда ће сада бити  $Q'=1, Q=0$
  - ако је претходно било  $Q=1, Q'=0$ 
    - онда ће сада бити  $Q'=0, Q=1$



НИЛИ  
0 0 1  
0 1 0  
1 0 0  
1 1 0

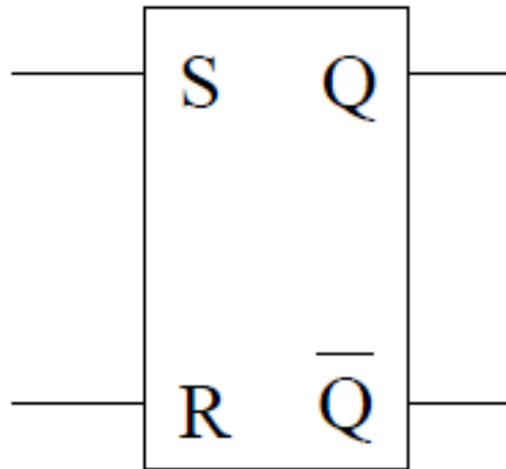
# SR реза – анализа понашања (2)

- Све док су оба улаза 1, оба излаза ће бити 0
- Проблем је у случају промене улаза са (1,1) на (0,0) :
  - промена улаза се никада у пракси не дешава дословно истовремено
    - или ће бити  $(1, 1) \rightarrow (0, 1) \rightarrow (0, 0)$
    - или ће бити  $(1, 1) \rightarrow (1, 0) \rightarrow (0, 0)$
  - ако се промена одвија као:  $(1, 1) \rightarrow (0, 1) \rightarrow (0, 0)$ 
    - прва промена поставља  $(Q, Q')$  на  $(0, 1)$
    - друга не мења стање
  - ако се промена одвија као:  $(1, 1) \rightarrow (1, 0) \rightarrow (0, 0)$ 
    - прва промена поставља  $(Q, Q')$  на  $(1, 0)$
    - друга не мења стање
- Због тога што се тако добија недетерминистичко понашање, улаз (1, 1) се сматра за неисправан (!!!)



# SR реза

- Логички симбол и истинитосна таблица:

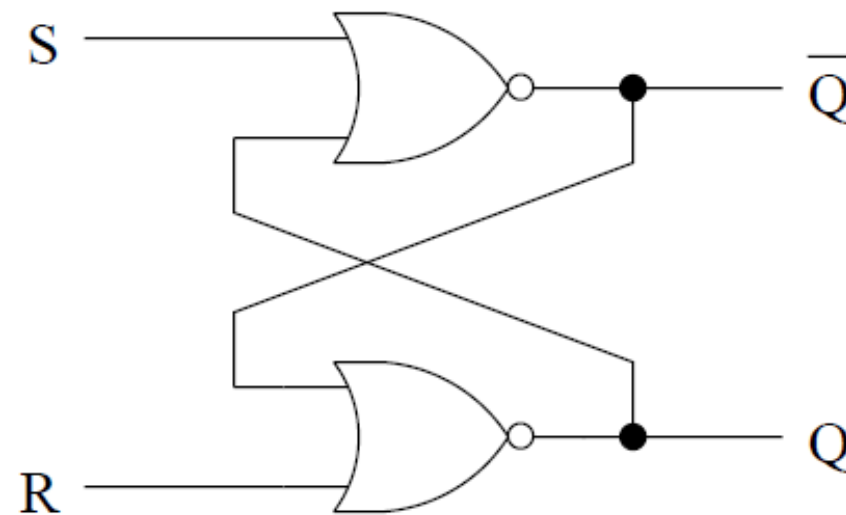
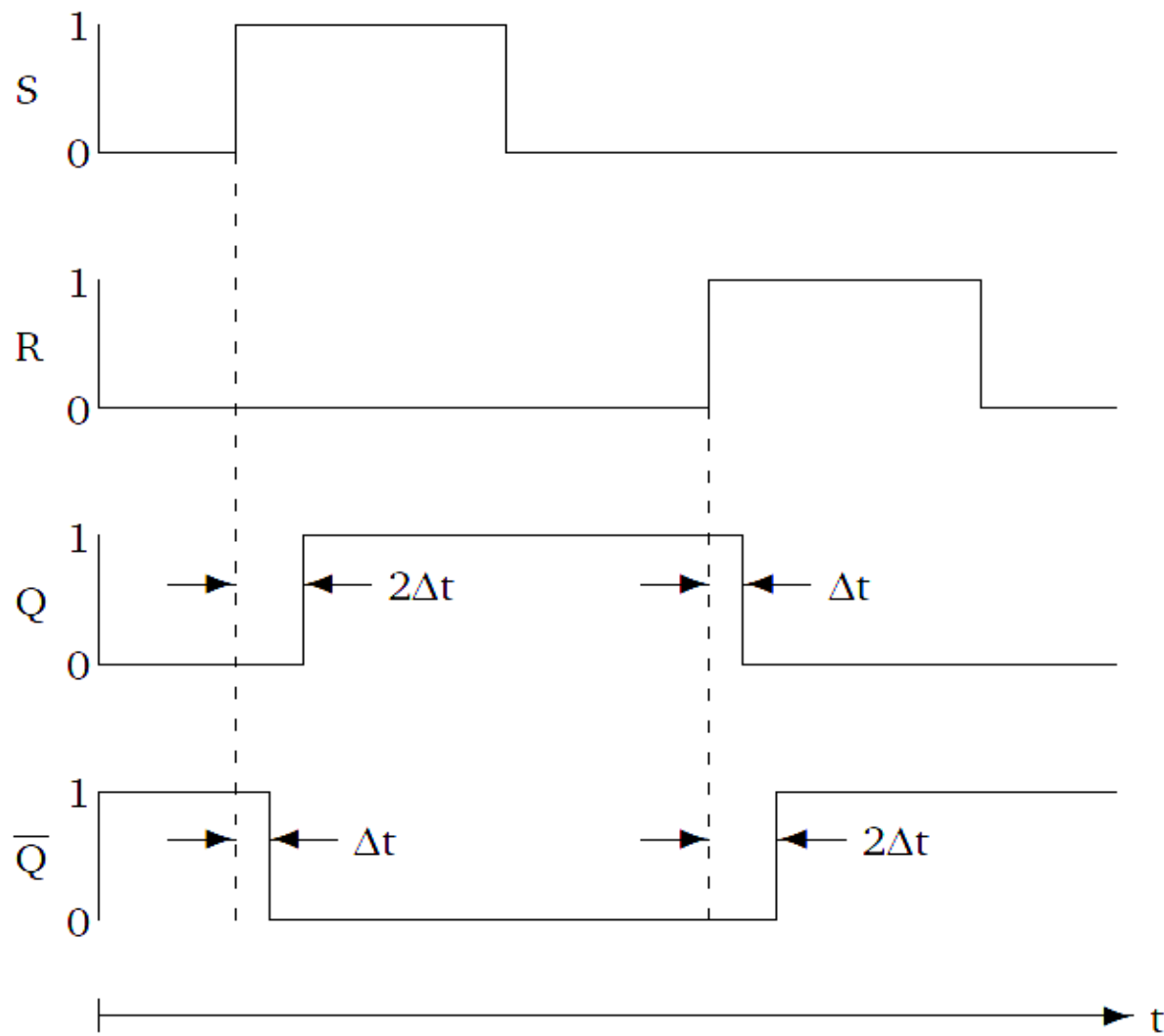


S	R	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	0

# Понашање $SR$ резе

- Ако су оба улаза неактивна, чува стање
- Ако је само улаз  $R$  активан, поставља стање на 0
- Ако је само улаз  $S$  активан, поставља стање на 1
  
- Излаз  $SR$  резе се мења асинхроно у односу на улаз, у зависности од брзине употребљених НИЛИ елемената
  
- Анимација: <https://www.youtube.com/watch?v=--pv3MZMoo0>

# Временски дијаграм $SR$ резе

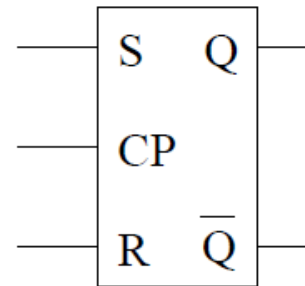
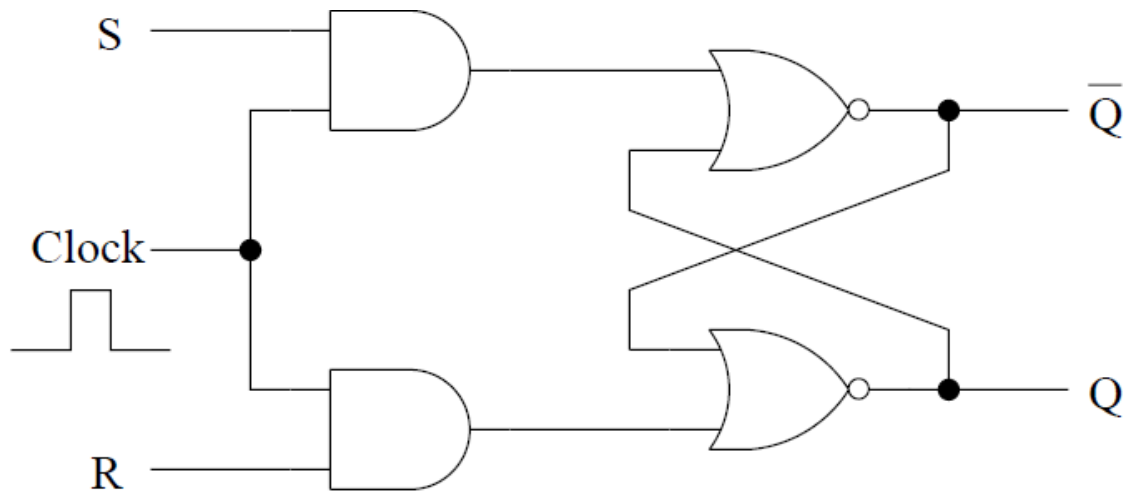


- $t_0: (S,R)=(0,0), (Q,Q')=(0,1)$
- $t_1: (S,R)=(\mathbf{1},0), (Q,Q')=(0,1)$
- $t_2: (S,R)=(1,0), (Q,Q')=(0,\mathbf{0})$
- $t_3: (S,R)=(1,0), (Q,Q')=(\mathbf{1},0)$
- $t_4: (S,R)=(\mathbf{0},0), (Q,Q')=(1,0)$  (nema promene na  $Q,Q'$  jer  $0,0$  cuva stanje)
- $t_5: (S,R)=(0,\mathbf{1}), (Q,Q')=(1,0)$
- $t_6: (S,R)=(0,1), (Q,Q')=(\mathbf{0},0)$
- $t_7: (S,R)=(0,1), (Q,Q')=(0,\mathbf{1})$



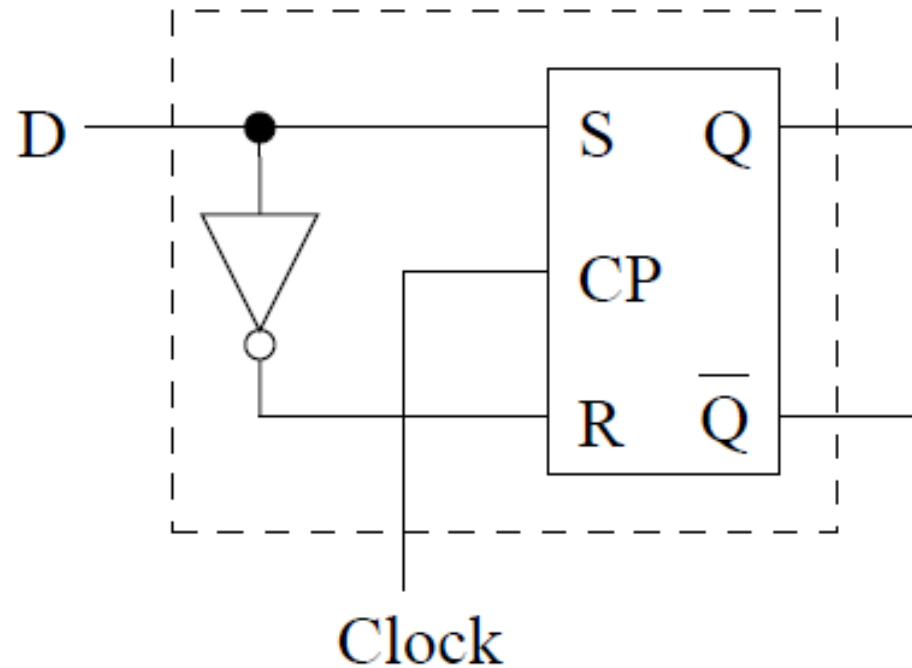
# SR реза са часовником

- Синхронизација се остварује додавањем улазног сигнала часовника у коло
  - тако улазни сигнали не утичу на евентуалну промену све док сигнал часовника не достигне висок ниво



# D реза

- Проблем са свим врстама *SR* реза је у томе што мора да се избегава пар вредности (1,1) на улазу
- То се може решавати применом *D* резе:

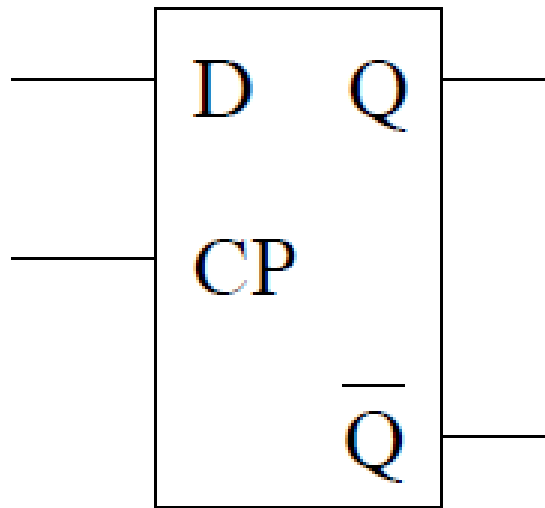


## D реза – анализа понашања

- Све док је сигнал часовника неактиван, промене на улазу немају утицаја на резултат, тј. резултат је исти као и раније
- У тренутку активирања сигнала часовника (или контролног сигнала), стање улаза се пропагира на излаз

# D реза

- Логички симбол и таблица истинитосних вредности



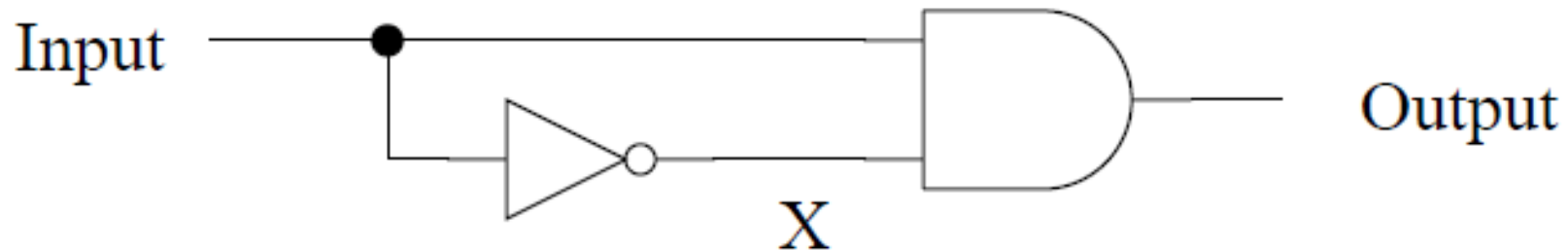
D	$Q_{n+1}$
0	0
1	1

# Флип-флоп кола

- Секвенцијална мрежа се назива *флип-флоп* ако се вредности улаза употребљавају само на једном рубу циклуса часовника
  - обично на узлазном рубу
- Тиме се омогућава да се у осталим фазама циклуса промене улаза практично игноришу и не ремете рад кола

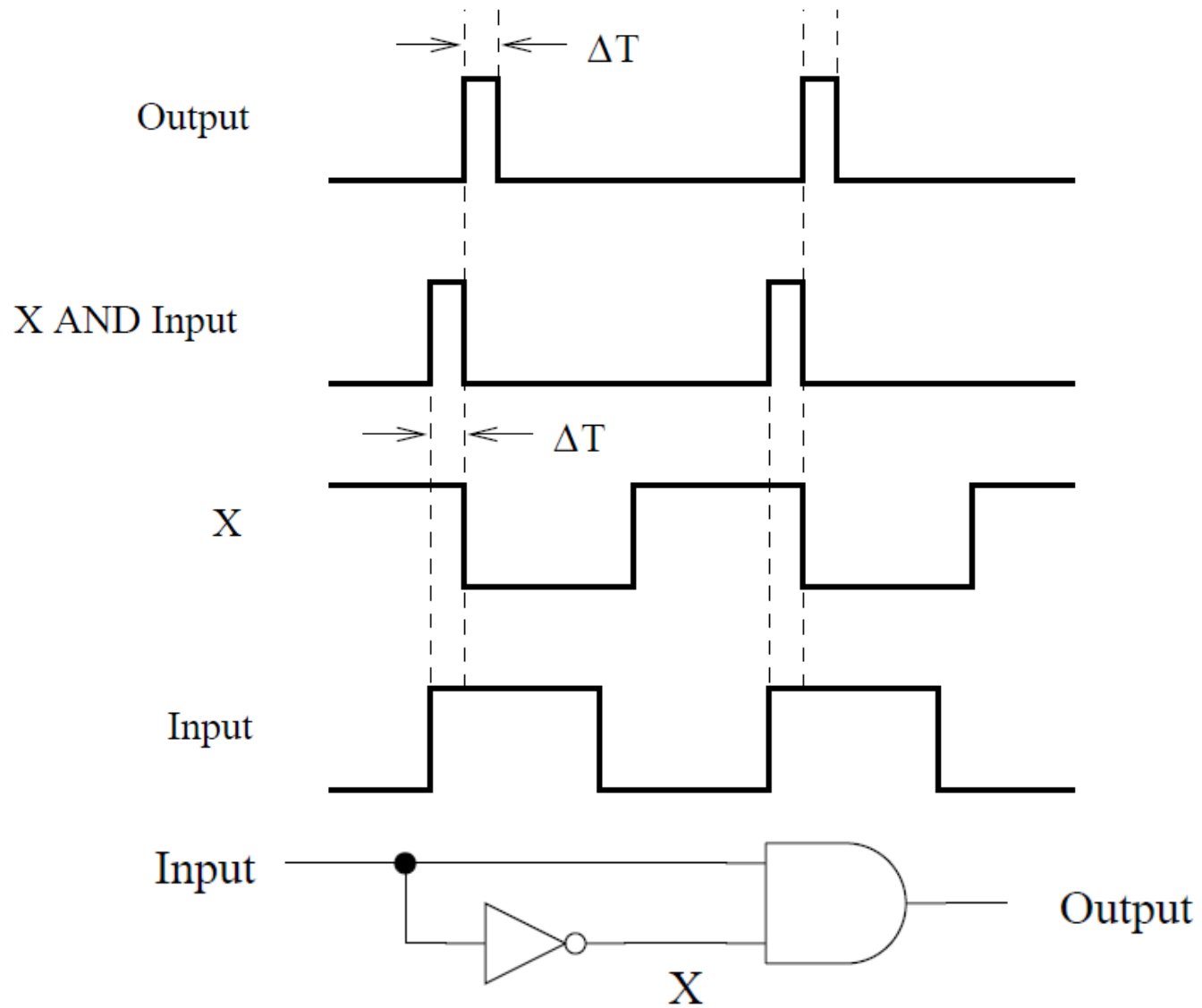
# Ограничавање на улазни руб

- Улазни сигнал се може ограничити на употребу само на улазном рубу часовника применом једноставног кола:



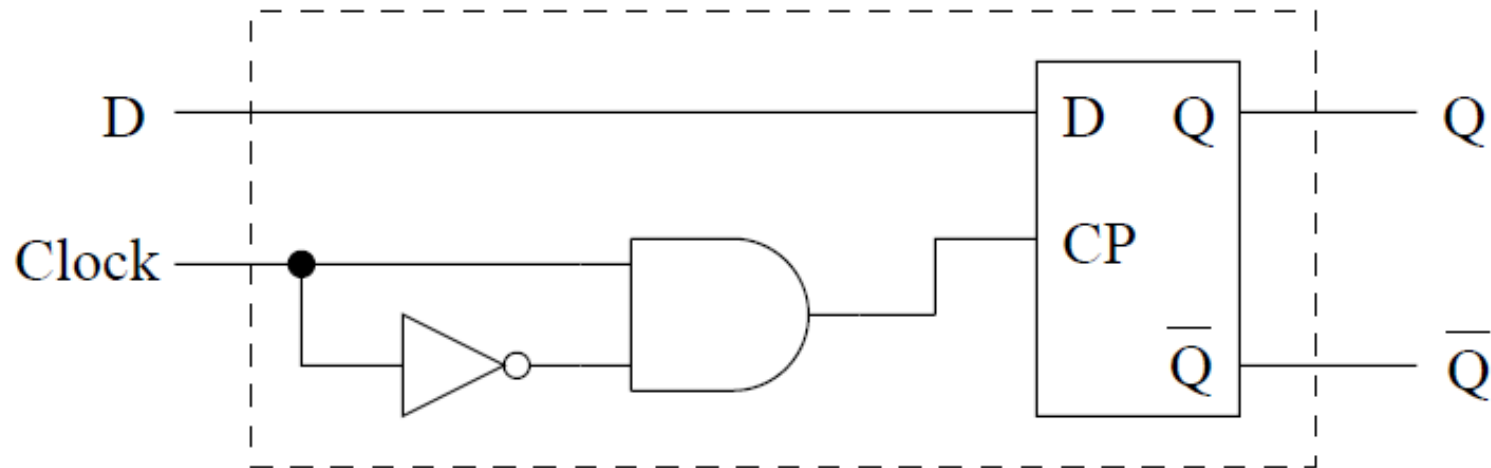
Излаз из овог кола је активан само на почетку циклуса (на улазном рубу) онолико дуго колико је НЕ елементу потребно да пропагира промену

# Ограничавање на улазни руб (2)



# D флип-флоп

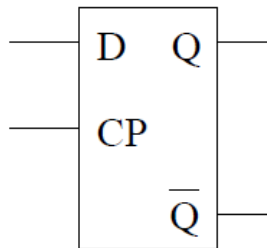
- D флип-флоп се прави помоћу D резе, ограничавањем контролног сигнала на узлазни руб циклуса:



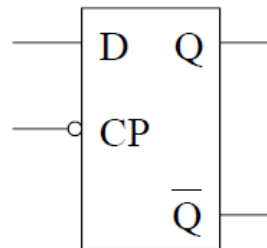


# Симболи за резе и флип-флопове

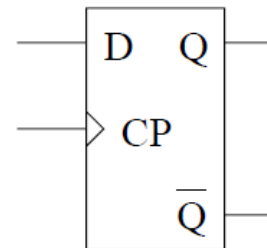
- Логички симболи за флип-флопове се разликују по нацртаном врху стрелице контролног улаза (часовника):



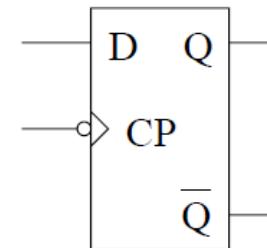
(a)



(b)



(c)

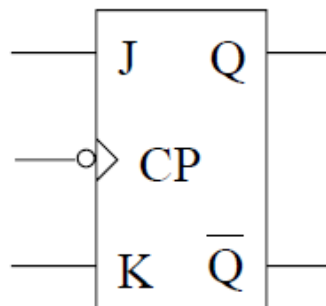


(d)

- (a)  $D$  реза осетљива на висок ниво контролног сигнала
- (b)  $D$  реза осетљива на низак ниво контролног сигнала
- (c)  $D$  флип-флоп осетљив на узлазни руб циклуса
- (d)  $D$  флип-флоп осетљив на силазни руб циклуса

# JK флип-флоп

- JK флип-флоп се понаша као SR флип-флоп, само што стање 11 користи за инверзију стања (toggle)
- Прави се помоћу две SR резе (*главне и подређене*)
  - главна SR реза се активира током активног дела циклуса
  - излаз главне SR резе се преноси на излаз током неактивног дела циклуса
- Због тога што се излаз прави на силазном рубу логички симбол је:



# Секвенцијалне мреже

Примери неких мрежа: регистри и бројачи

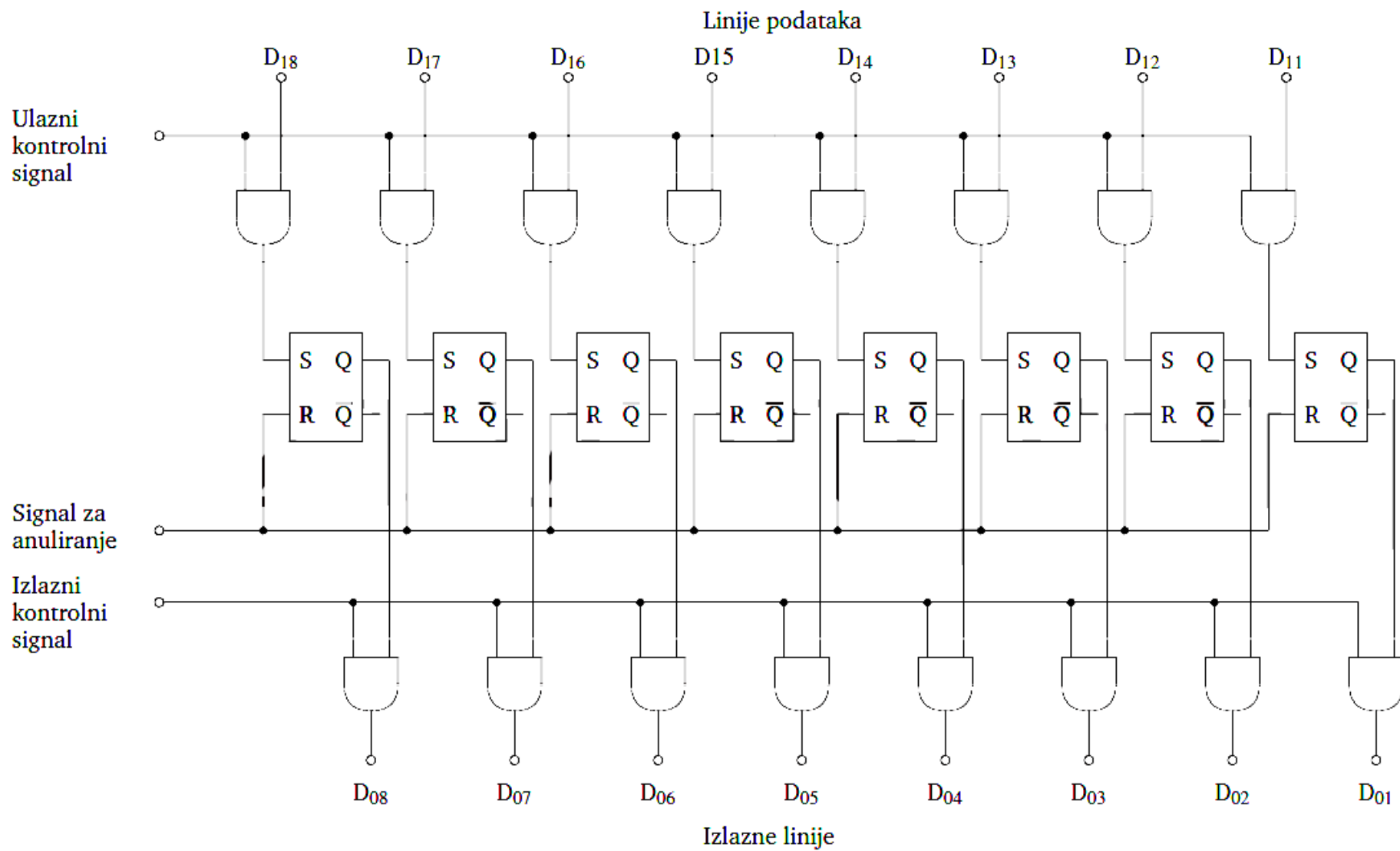
# Примери секвенцијалних мрежа

- Неки од примера секвенцијалних мрежа су:
  - регистри
    - паралелни регистри
    - померачки регистри
  - бројачи

# Паралелни регистри

- Регистри су кола која се користе за чување једног или више битова података
- Паралелни регистар се састоји од скупа 1-битних меморијских јединица чији се садржај може истовремено читати или мењати
- Може се имплементирати помоћу реза или флип-флопова

# Паралелни регистар са SR резом (2)



# Паралелни регистар (3)

- Претходна имплементација омогућава три основне операције
  - анулирање
    - ако је активан контролни сигнал за анулирање
  - уписивање
    - ако је активан улазни контролни сигнал за уписивање
    - уз претпоставку да је претходно извршено анулирање
  - читање
    - ако је активан излазни контролни сигнал

# Померачки регистри

- Померачки регистри померају низ битова улево или удесно са сваким циклусом часовника
- Могу да се употребљавају за конвертовање из паралелног у серијски вид комуникације и обратно



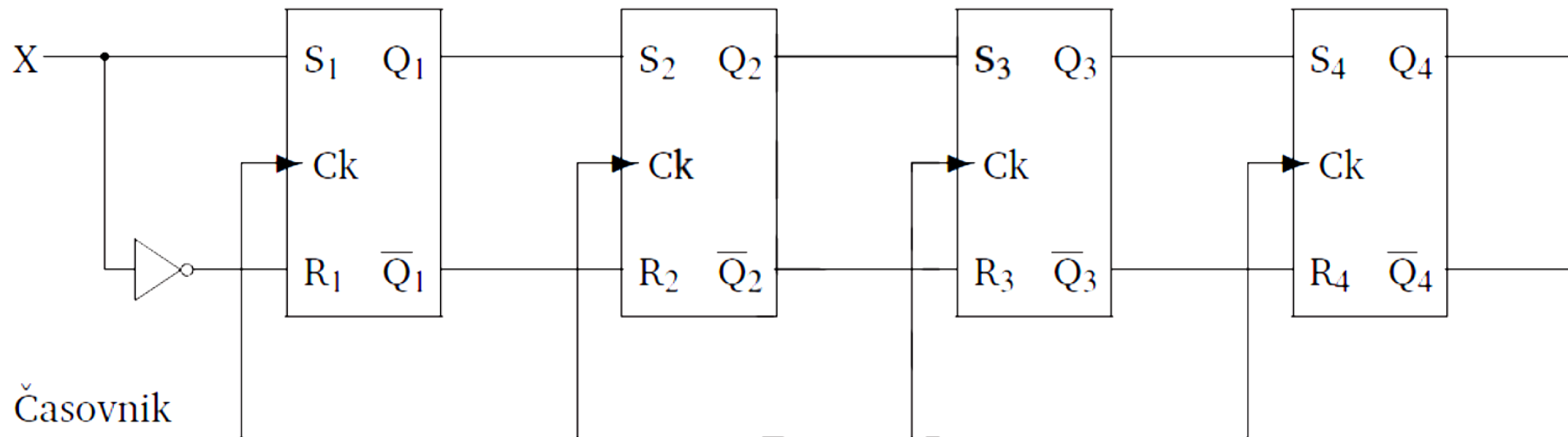
# Логичко померање

- Не постоји улазни податак
- Уместо улаза дописује се
  - нула (тзв. *линијско померање*) или
  - вредност “избаченог” бита (тзв. *циклично померање*)

# Аритметичко померање

- Не постоји улазни податак
- Уместо улаза дописује се
  - нула, ако је померање улево
  - највиши бит, ако је померање удесно

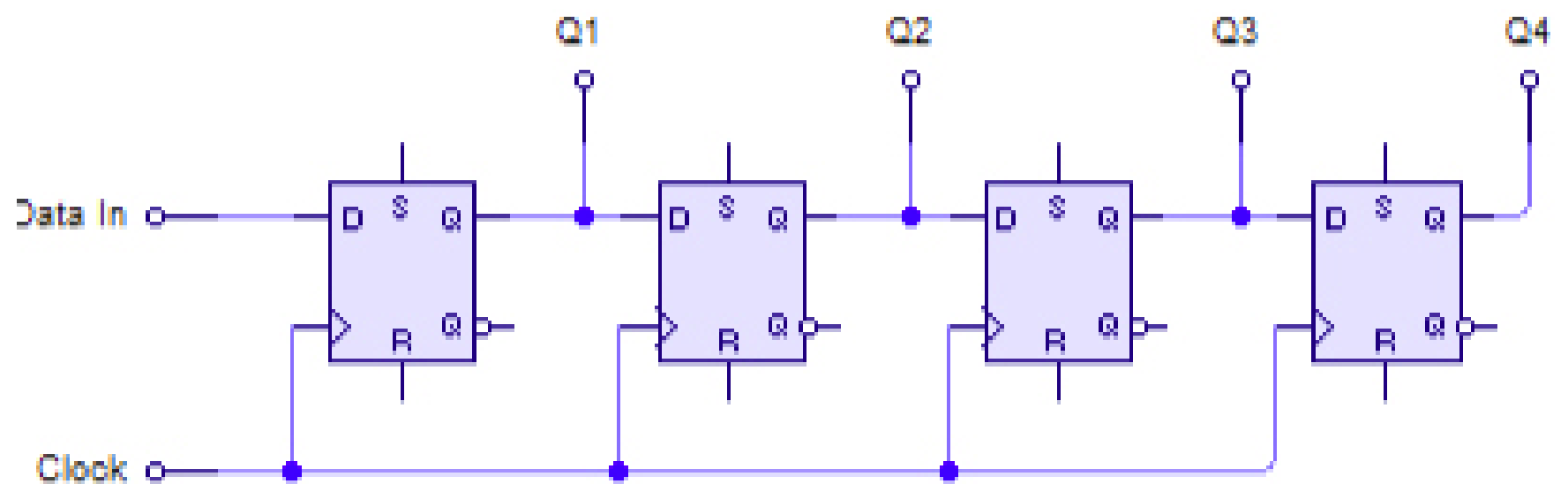
# 4-битни померачки регистар са серијским улазом и излазом (SR фф)



У сваком циклусу:

садржај се помера за једно место удесно  
слева се дописује један нови бит са улаза  
на излазу се чита један “избачен” бит

# 4-битни померачки регистар са паралелним излазом (D фф)



У сваком циклусу:

садржај се помера за једно место удесно

слева се дописује један нови бит са улаза

чита се на паралелном излазу комплетан нови садржај

# Бројачи

- Бројачи су секвенцијалне мреже које са сваким циклусом повећавају вредност регистра за 1
- Веома често се употребљавају
- Бинарни бројач користи за бројање регистар са  $B$  битоа и омогућава бројање од 0 до  $2^B-1$ 
  - Након “прекорачења” се почиње поново од нуле
  - Назива се и бројач “по модулу  $2^B$ ”
- Бројач по модулу 10 се употребљава за рад са декадним цифрама

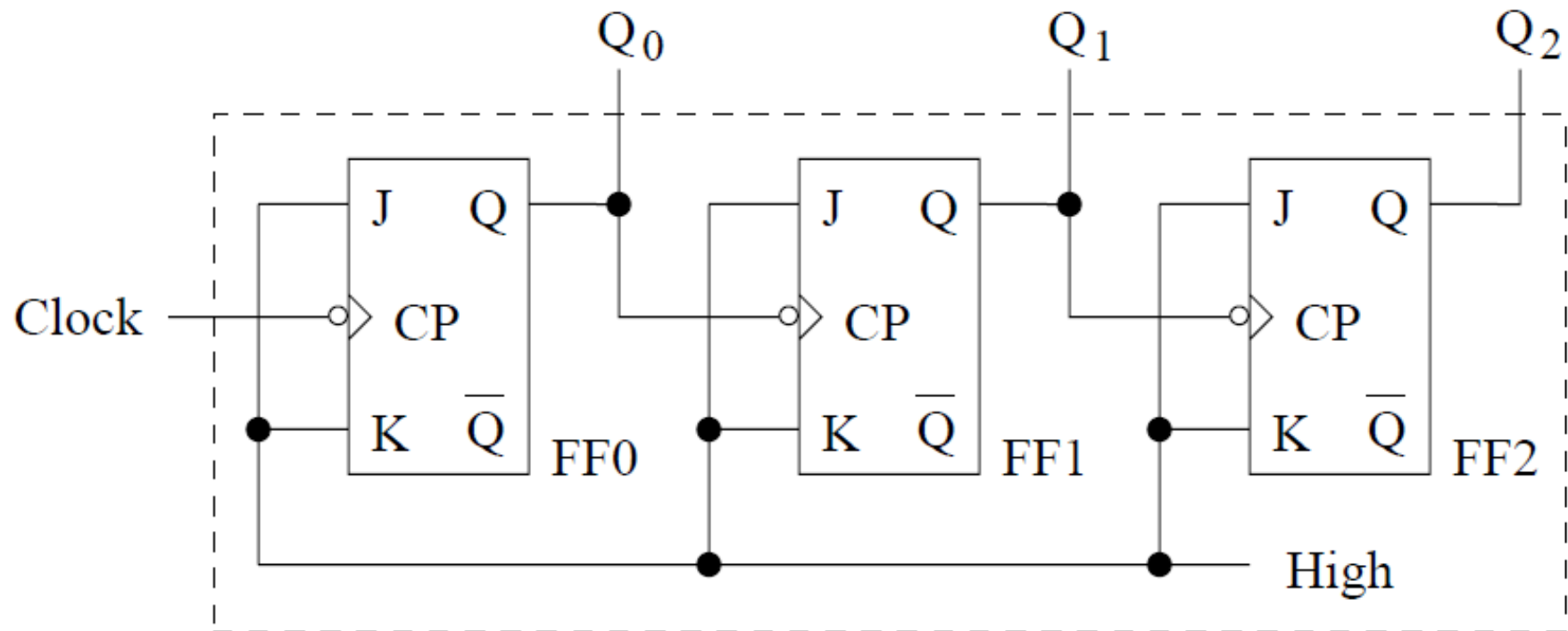
# Имплементација бројача

- Имплементирамо бројач по модулу 8
- Низ битова се посматра као низ бинарних цифара
  - Најнижи бит се мења у сваком циклусу
  - Виши бит је потребно променити сваки пут када се претходни нижи бит промени из 1 у 0

# Имплементација бројача (2)

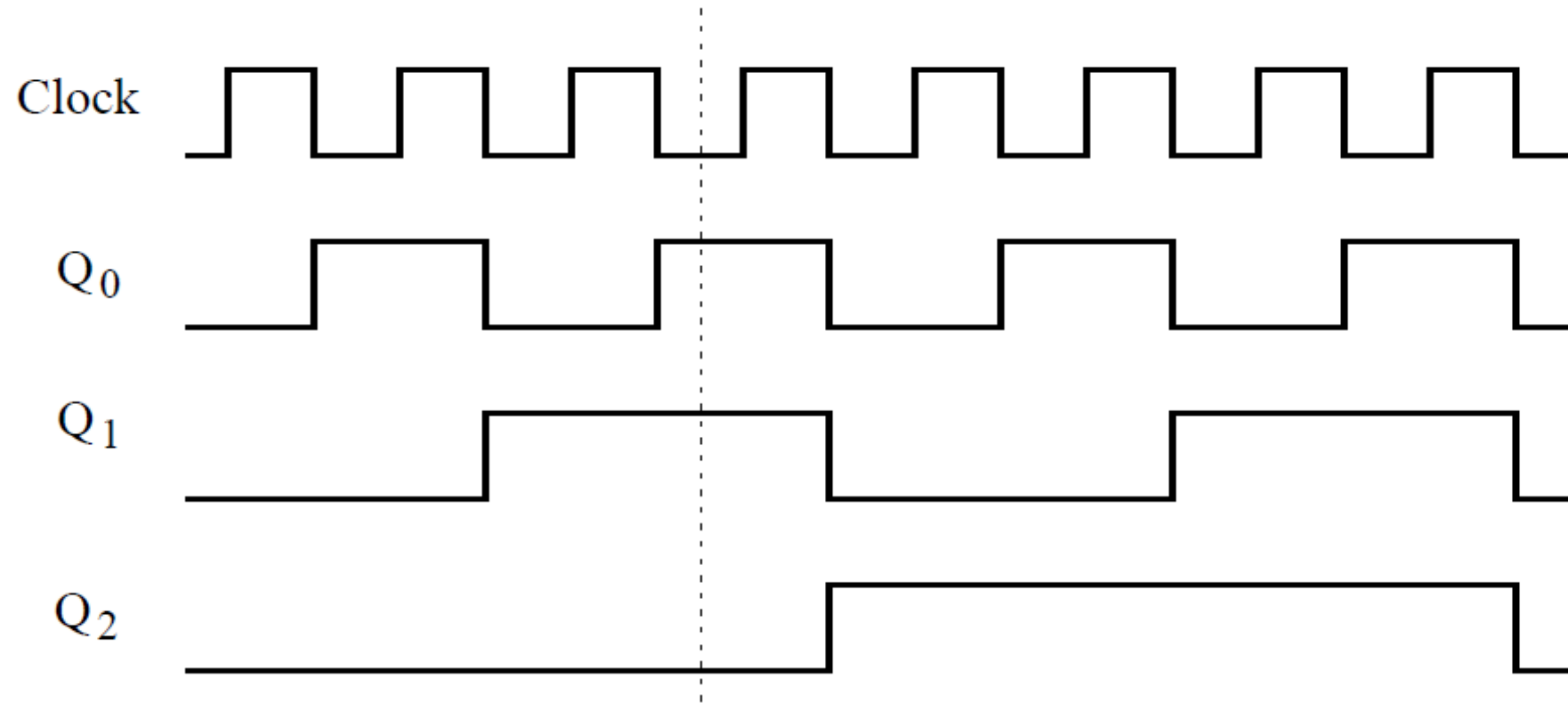
- Подсећање:
  - Ако се улази на *JK* флип-флопу поставе на 1, онда ће се излаз мењати у сваком циклусу
- Идеја имплементације је:
  - на низ *JK* флип-флопова се доведу на улазе активни сигнали
  - на контролни улаз првог (који одговара најнижем биту бројача) се доведе сигнал часовника
  - на контролне улазе осталих се доведу излазни сигнали претходних

# Имплементација бројача (3)





# Понашање бројача



# Неке примене бројача

- Бројачи се могу употребљавати
  - за мерење времена
  - за бројање података на улазу или излазу
  - за генерисање часовника са споријим тактом
  - и друго