Дигитални склопови (Булова алгебра и логички кола)

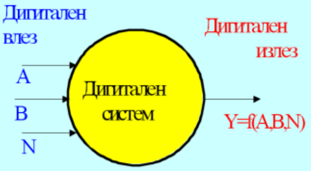
Проф. д-р Јосиф Ќосев
Доц. д-р Томислав Карталов

(во соработка со проф. д-р Методија Камиловски)

Теми

- Дигитални системи и дигитализација
- Бројни системи
- Основни логички кола (и математичката логика)
  - Инвертор (НЕ-коло), И-коло, ИЛИ-коло
- Други логички кола
  - НИ-коло, НИЛИ-коло, ИСКЛУЧИВО ИЛИ-коло, ...
- Булова алгебра
  - Закони на Буловата алгебра
- Комбинациони склопови
  - Кодер, Декодер, ROM, Мултиплексер, (компаратор)
  - Потполн собирач
  - Општа форма на комбинациони функции
- Секвенцијални склопови
  - Бистабилно коло, Леч, Флип-флоп
  - Регистар, Бројач
  - Меморија (RAM)
  - Конечни автомати

Дигитални системи



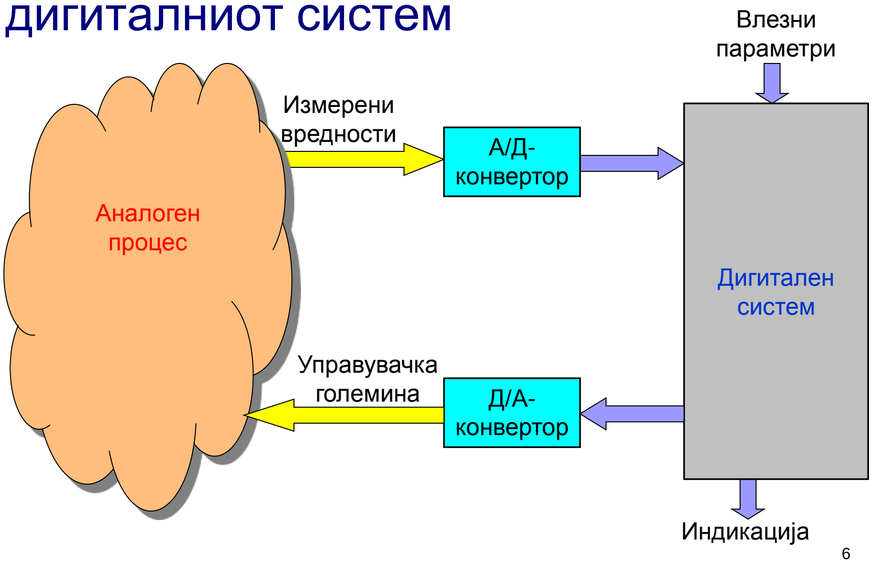
- Дефиниција:
  - **Систем кој манупулира со броеви/цифри (дигити)**
    - сите компоненти работат во **дигитален режим** и сите големини се дигитални

- Споредба на АНАЛОГНИТЕ и **ДИГИТАЛНИТЕ** големини:
  - **Континуирана** во време – **Дискретна** во време
  - **Континуирана** по вредност - Дискретна по вредност **(квантизирана)**
  - Претставена со **физичка величина** – Претставена со дигитален код **(кодирана)**

Дигитални системи

- **Предности**:
  - **Флексибилност**: промената на функцијата на системот се постигнува со замена на една низа од дигитални вредности со друга (т.н. софтвер) [наместо замена на компоненти или цел систем]
  - **Едноставност**: саканата функција се изведува со пресметки според дефиниран алгоритам (во софтвер) [наместо со физички елементи – електрични кола, механички, ...]
  - **Отпорност на шум/пречки**: вградена способност за потиснување пречки [наместо нивно акумулирање]
  - **Заштита од грешки**: зависно од кодирањето, релативно лесна детекција на грешки со соодветни алгоритми (парност, CRC код и др.) [наместо нивно акумулирање]

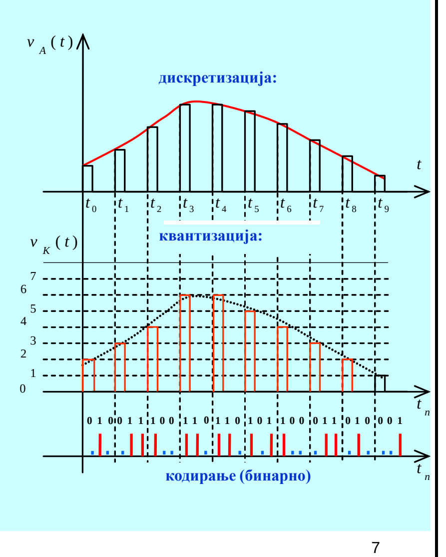
Врска меѓу аналогниот свет и дигиталниот систем



Дигитализација

- 1. **земање примероци (sampling)** (дискретизација во време) според **теоремата за земање примероци**: *сигналот може потполно да се репродуцира од примероците земени со фреквенција f_s што е барем два пати повисока од највисоката фреквенција во неговиот спектар* $f_s > 2f_{max}$,
- 2. **квантизација** (заокружување на конечен број вредности – на пр. за 8 напонски нивоа),
- 3. **кодирање** (претставување со секвенци од импулси, на пр. според бинарниот броен систем)
- **Бит**: единечен импулс со вредност 0 или 1
- **Бинарен броен систем**: броен систем со основа 2 (битовите имаат месна вредност = степенот на основата 2):

$$\overline{b_{n-1}b_{n-2}\cdots b_2 b_1 b_0} = \sum_{i=0}^{n-1} b_i 2^i$$



Бројни системи

- Систем на претставување на броевите дефиниран со множество на симболи (цифри), база и знак
  - Множество на симболи (цифри) ги содржи симболите за основните бројни вредности во системот, вообичаено 0 и првите природни броеви до базата (пр. 0,1,2,3,4,5,6,7,8 и 9)
  - База на системот (radix): број на симболи во множество на симболи
  - Знак: симбол/начин на претставување на негативните вредности

Бројни системи (2)

- Нотација базирана на позиција: нотација во која положбата на цифрите во бројот ја определува вредноста која ја претставуваат (месна вредност)

$$a_{n-1} \cdots a_1 a_0 = a_{n-1} b^{n-1} + \cdots + a_1 b^1 + a_0 b^0$$

b – база

n – должина на зборот

a_0 – најмалку значаен бит (LSB)

a_{n-1} – најзначаен бит (MSB)

Бројни системи (4)

- Во почеста употреба се следните:
  - Декаден
  - Бинарен
    - Природен
    - Бинарно кодиран декаден (BCD)
  - Хексадекаден
  - Октален

Декаден броен систем

- База: 10
- Цифри: 0,1,2,3,4,5,6,7,8 и 9

Пример:

$$5324 = 5*10^3+3*10^2+2*10^1+4*10^0$$

Бинарен броен систем (природен)

- База: 2
- Цифри: 0 и 1

Пример:

$10110_2 = 1*2^4+0*2^3+1*2^2+1*2^1 +0*2^0 = 22_{10}$

Бинарно кодиран декаден броен систем (BCD код)

- База: 2
- Цифри: 0 и 1
- Низи од четири бинарни цифри кодираат декадна цифра
- Групирани во број според декадниот броен систем

Пример:

5 9 3

$0101\ 1001\ 0011_{BCD} = 593_{10}$

Хексадекаден броен систем

- База: 16
- Цифри: 0,1,2,3,4,5,6,7,8, 9, A, B, C, D, E и F

Пример:

$2A9_H = 2*16^2 + 10*16^1 + 9*16^0 = 681_{10}$

$BABE_H = 11*16^3 + 10*16^2 + 11*16^1 + 14*16^0 = 47806_{10}$

$(B \quad A \quad B \quad E)_H$

$(1011\ 1010\ 1011\ 1110)_2$

- Погоден за луѓето за презентирање на непрегледните низи од битови кај бинарните броеви

Претворање од една во друга база

- Целобројниот дел се дели со новата база, остатокот при секое делење е цифрата во новиот броен систем, почнувајќи од најмалата

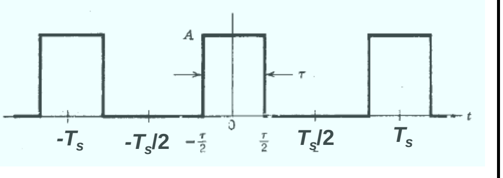
Пример: декаден во бинарен

134 / 2 = 67	ост	0	LSB
67 / 2 = 33	ост	1	
33 / 2 = 16	ост	1	
16 / 2 = 8	ост	0	
8 / 2 = 4	ост	0	
4 / 2 = 2	ост	0	
2 / 2 = 1	ост	0	
1 / 2 = 0	ост	1	MSB

$= 10000110_2$

За земањето примероци (1) (информативно)

- Земање примероци = множење на аналогниот сигнал со правоаголна периодична функција $p(t)$ со $\tau \ll T_s$ и A=1:



- Според Фурие може да напишеме:

$$p(t) = C_0 + 2C_1 \cos \omega_s t + 2C_2 \cos 2\omega_s t + \ldots + 2C_n \cos n\omega_s t + \ldots$$

$$C_0 = \frac{\tau}{T_s}, \qquad C_n = \frac{\tau}{T_s} \frac{\sin(n\pi\tau/T_s)}{n\pi\tau/T_s}, \qquad \omega_s = \frac{2\pi}{T_s}$$

За земањето примероци (2) (информативно)

- Ако сигналот што го семплираме е тест-тон...:

$$x(t) = X \cos \omega_m t$$

- ...тогаш резултантниот сигнал е:

$$x_s(t) = C_0 \cdot X \cos \omega_m t + 2C_1 \cos \omega_s t \cdot X \cos \omega_m t +$$
$$+ 2C_2 \cos 2\omega_s t \cdot X \cos \omega_m t + \ldots + 2C_n \cos n\omega_s t \cdot X \cos \omega_m t + \ldots$$

- А со малку тригонометрија:

$$x_s(t) = C_0 \cdot X \cos \omega_m t + C_1 X \cos(\omega_s - \omega_m)t + C_1 X \cos(\omega_s + \omega_m)t +$$
$$+ C_2 X \cos(2\omega_s - \omega_m)t + C_2 X \cos(2\omega_s + \omega_m)t + \ldots$$

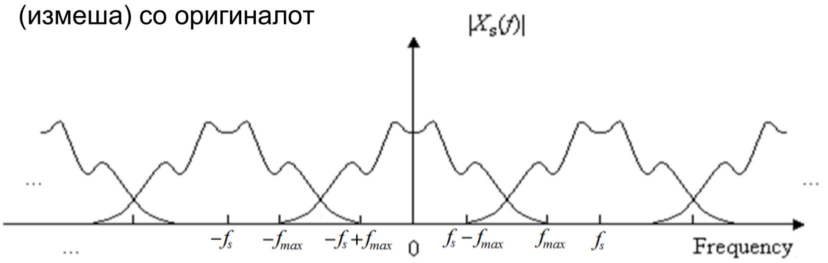
За земањето примероци (3) (информативно)

- Последниот израз покажува дека земањето примероци произведува ефект сличен на амплитудната модулација:
  - покрај основниот сигнал во спектарот се јавуваат и слики околу фреквенциите f_s, $2f_s$, …, nf_s, …

$$x_s(t) = C_0 \cdot X\cos\omega_m t + C_1 X\cos(\omega_s - \omega_m)t + C_1 X\cos(\omega_s + \omega_m)t + \textit{останато}$$

За земањето примероци (3) (информативно)

- Ако f_s не е доволно висока, првата слика ќе се преклопи (измеша) со оригиналот

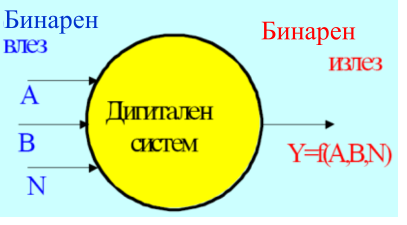


преклопување

- Преклопувањето е познато под името aliasing.
- За да нема преклопување, треба да биде задоволен условот

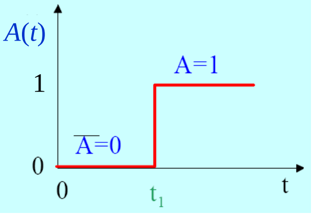
$$f_s - f_{\max} > f_{\max} \quad \text{или} \quad f_s > 2f_{\max}$$

Бинарен дигитален систем



Функцијата на системот може да се дефинира преку:

Бинарна променлива



Табела на вистинитост

Влез			Излез
A	B	N	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Од каде „Табела на вистинитост“?

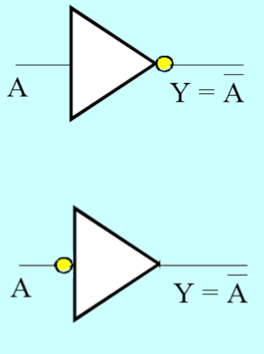
- Произлегува од еквивалентноста на основните операции кај бинарните променливи со операциите со вистинитосни вредности во алгебрата на искази (од математичката логика)
- Ако се апстрахира вредноста на симболите 1 и 0 тие можат да се означат и поинаку: H и L, или T и ⊥, или True и False …
- Основните операции според алгебрата на искази со променливите што примаат вредности од множеството {1,0}, односно {T, ⊥} се:
  - Логичко множење (·) ⇔ конјункција (∧) = И (=AND)
  - Логичко собирање (+) ⇔ дисјункција (∨) = ИЛИ (=OR)
  - Комплементирање (¯) ⇔ негација (¬) = НЕ (=NOT)

Логички кола

- Кола што ги извршуваат **логичките операции**

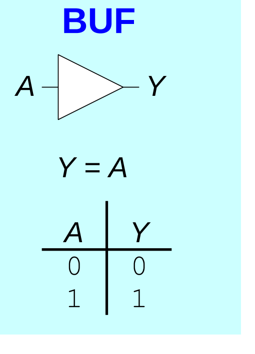
Основно логичко коло: Инвертор (НЕ-коло)

- Врши **логичка негација**: состојбата на излезот е **комплемент** на состојбата на влезот
  - **Кругчето** означува негација



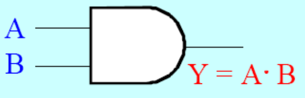
Влез	Излез
A	**Y**
0	**1**
1	**0**

Помошно коло - бафер (следител):



Основни логички кола:

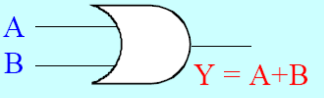
И-коло



Влез Излез

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

ИЛИ-коло

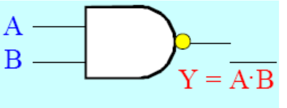


Влез Излез

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Изведени логички кола:

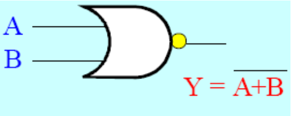
НИ-коло



Влез Излез

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

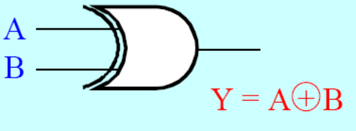
НИЛИ-коло



Влез Излез

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Изведено логичко коло: ИСКЛУЧИВО-ИЛИ-коло



Влез		Излез
A	**B**	**Y**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

Булова алгебра

- Алгебра над множеството {0,1} (т.е {FALSE,TRUE} или {⊥,T})
- Дефинициите на основните логички функции се нарекуваат аксиоми на Буловата алгебра

Булова алгебра (закони)

1. Идентитет со логичка нула:
   $0 + A = A \qquad 0 \cdot A = 0$
2. Идентитет со логичка единица
   $1 + A = 1 \qquad 1 \cdot A = A$
3. Идентитет со исти вредности
   $A + A = A \qquad A \cdot A = A$
4. Идентитет со комплементарни вредности
   $A + \overline{A} = 1 \qquad A \cdot \overline{A} = 0$

Булова алгебра (закони)

5. Закон за комутација:
   $A + B = B + A \qquad A \cdot B = B \cdot A$
6. Закон за асоцијација:
   $A + (B + C) = (A + B) + C$
   $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
7. Закон за дистрибуција:
   $A \cdot (B + C) = A \cdot B + A \cdot C$
   $A + (B \cdot C) = (A + B) \cdot (A + C)$
8. Закон за апсорпција:
   $A + A \cdot B = A \qquad A \cdot (A + B) = A$
   $A + \overline{A} \cdot B = A + B; A \cdot (\overline{A} + B) = A \cdot B$

Булова алгебра (закони)

- Теореми на Де Морган

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

- Законите од Буловата алгебра можат да се докажат со исцрпување на сите случаи во вистинитосна табела

Булова алгебра
пример за докажување

A+(B·C) = (A + B)·(A + C)

A	B	C	BC	L=A+BC	A+B	A+C	D=(A+B)(A+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Логички мрежи

- Системи изградени од логички кола
- Комбинациони мрежи
  - Излезите зависат само од влезовите
  - Немаат меморија
- Секвенцијални мрежи
  - Излезите зависат од влезовите и од некои излези
  - Имаат повратни врски
  - Имаат меморија

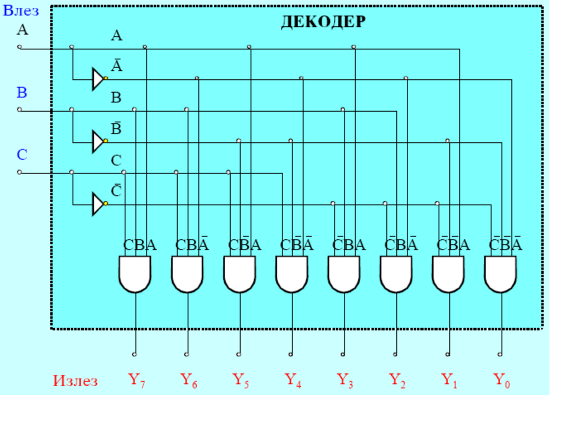
- Комбинациони логички мрежи

Декодер

- Логичка мрежа со повеќе влезови и излези
- За секоја комбинација на влезните битови се активира само еден од излезите и тој е на логичка 1. Сите други излези се на логичка 0

Влез			Излез							
C	B	A	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Шема на декодер „3 на 8“

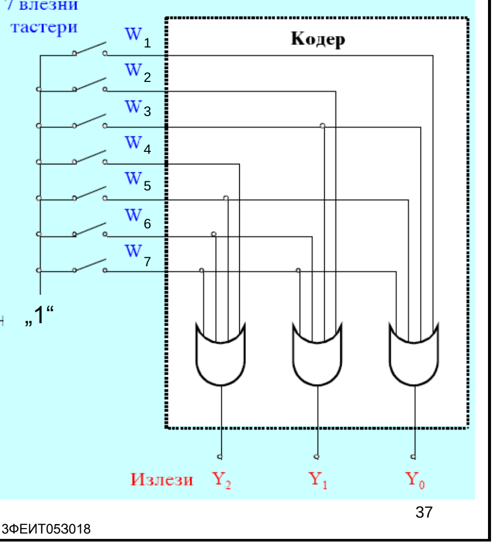


Кодер (Encoder)

- Логичка мрежа со повеќе влезови и излези
- Само еден од влезовите е на логичка 1, сите други се на логичка 0.
  - На излезот се добива бинарен код кој опишува кој од влезовите е на логичка 1
- Ако има логичка 1 на повеќе од еден влез, тоа е недозволена состојба и кодерот греши.
- Варијација: приоритетен кодер
  - Излезот е кодот на највисокиот влез

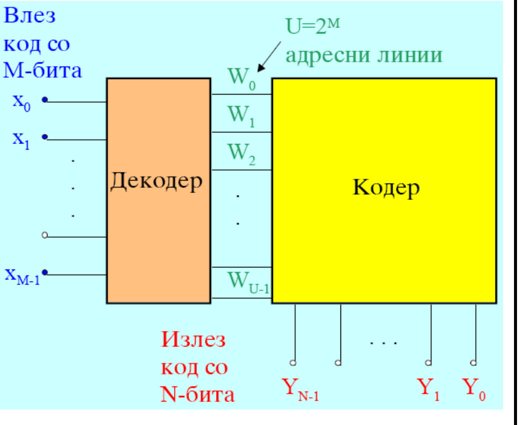
Кодер „7 (8) на 3“

Влез							Излез		
W_7	W_6	W_5	W_4	W_3	W_2	W_1	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	**1**	0	0	1
0	0	0	0	0	**1**	0	0	1	0
0	0	0	0	**1**	0	0	0	1	1
0	0	0	**1**	0	0	0	1	0	0
0	0	**1**	0	0	0	0	1	0	1
0	**1**	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	1	1	1

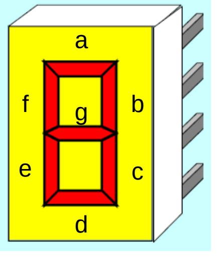


ROM (Read Only Memory)

- Потекнува од времето кога содржината се фиксирала при производството
- Функција:
  - Lookup table
- PROM (programmable)
  - Ги содржи сите врски, а корисникот програмира (гори) дел од нив и втиснува своја содржина



ROM за претворање на бинарен (BCD) во 7-сегментен код [или т.н. BCD-to-7-segment-Decoder]



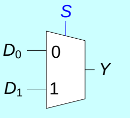
Влез					Излез						
Бр.	D	C	B	A	g	f	e	d	c	b	a
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	0
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	0	1	1	1

Мултиплексер

- Логичка мрежа со повеќе влезови и еден излез
- Има две групи влезови:
  - Селектирачки
  - Селектирани
- Селектирачките влезови одредуваат кој од селектираните влезови ќе се појави на излезот.

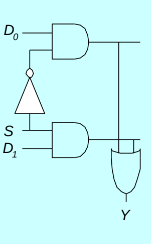
Пример:

2:1 мултиплексер



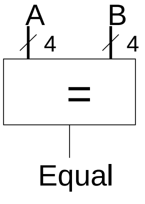
S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Y
0	D_0
1	D_1

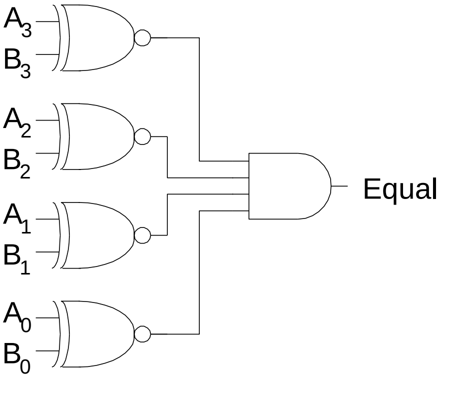


Дигитален компаратор: Еднакво

Симбол



Изведба



Потполн собирач (1)

- Комбинациона **логичка мрежа** што врши **аритметичко** собирање на две бинарни цифри
- Ги поврзува логичките и аритметичките операции.
- Пример за аритметичко собирање:

- Декаден систем
- пренос1
- 286
- +622
- =908

Бинарен систем:

пренос1

100110

+ 101101

=1010011

Потполн собирач (2)

- Табела на аритметичко собирање во декаден броен систем:
  - (од прво одделение основно ☺)

+	0	1	2	...	8	9
0	0	1	2	...	8	9
1	1	2	3	...	9	10
2	2	3	4	...	10	11
...	...	...	...	...	...	...
8	8	9	10	...	16	17
9	9	10	11	...	17	18

Пренос: ↑ ↑ ↑ ↑

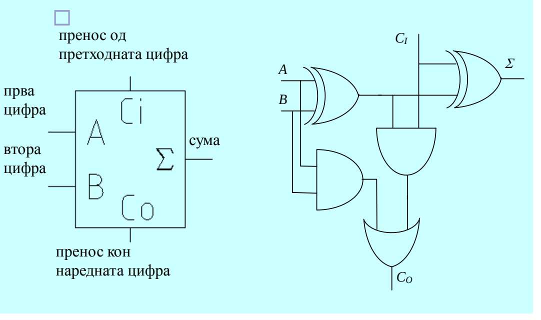
- Табела на аритметичко собирање во бинарен броен систем:

+	0	1
0	0	1
1	1	10

Пренос: ↑

Потполн собирач (3)

- Блок-дијаграм и логичка реализација

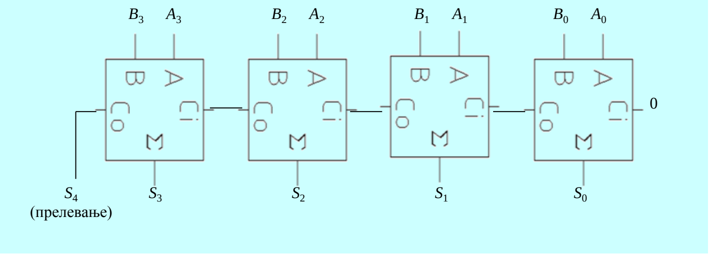


Табела на вистинитост:

Ci	*A*	*B*	*Σ*	*Co*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Повеќебитен потполн собирач

- Како да собереме два четирибитни броја?



Генерална форма на комбинациона функција – SOP (сума од производи) - информативно

- За секоја редица од табелата во која функцијата има вредност 1 се формираат производи (т.н. минтерми) од сите променливи во редицата така што променливата учествува некомплементирана ако табелата содржи 1, а комплементирана ако содржи 0.
- Функцијата се состои од логичка сума на така формираните минтерми.
- Пример за потполниот собирач
  - Очигледно далеку од оптимален резултат (централна тема во ЛКДА)
  - Оптимизација: Карноови мапи, Квин-Мекккласки...

Ci	*A*	*B*	*Σ*	*Co*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

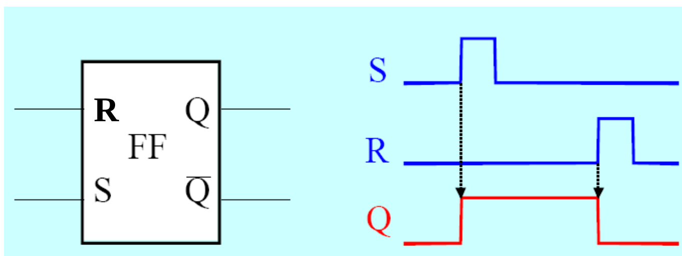
$$\Sigma = \overline{C}_i\overline{A}B + \overline{C}_iA\overline{B} + C_i\overline{A}\overline{B} + C_iAB$$

$$C_o = \overline{C}_iAB + C_i\overline{A}B + C_iA\overline{B} + C_iAB$$

- **Секвенцијални логички мрежи**

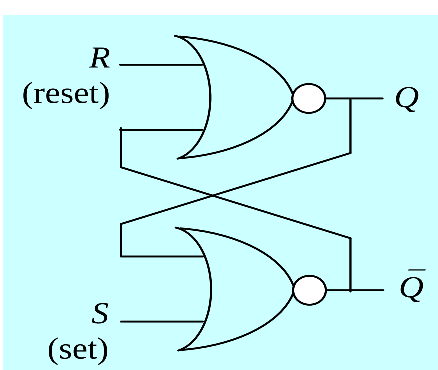
SR-леч (latch)

- Функција:
  - Импулс (1) на S-влезот го поставува Q на 1
  - Импулс (1) на R-влезот го поставува Q на 0
  - При S=0 и R=0 останува претходната состојба



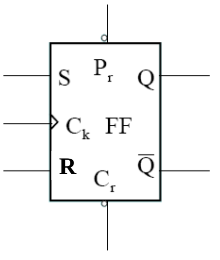
Изведба на SR-леч

- Со НИЛИ кола:
  - Анализата произлегува од поведението на НИ-колото
  - Двата влеза на 1 е недозволено бидејќи не се знае што ќе остане на излез по враќањето на 0

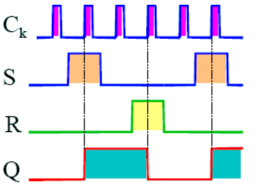


S	*R*	*Q*
1	0	1
0	1	0
0	0	Q_{n-1} (меморира)
1	1	? (неопределен)

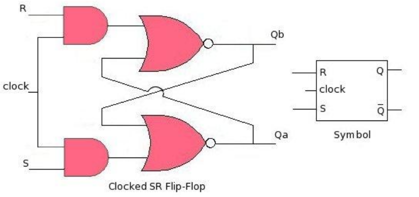
SR- леч со влез за такт
(се нарекува и SR флипфлоп)



S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?



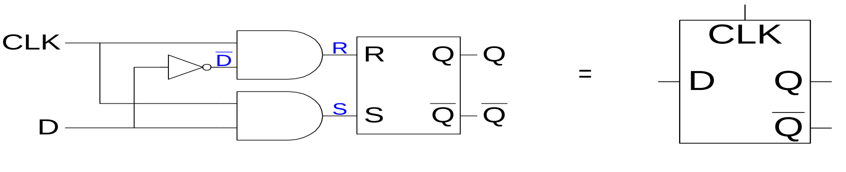
- S и R делуваат само кога тактот е 1 (Ck=1)



Clocked SR Flip-Flop

D – леч (внатрешна структура)

- Обезбедува S и R никогаш да не бидат 1-1.

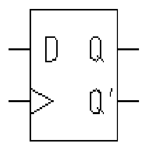


CLK	D	$\overline{D}$	S	R	Q	$\overline{Q}$
0	X	$\overline{X}$	0	0	Q_{prev}	$\overline{Q}_{prev}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

X = билошто (seedно дали 0 или 1)

D-флипфлоп

- Менува состојба само на ивица од тактот



Симбол

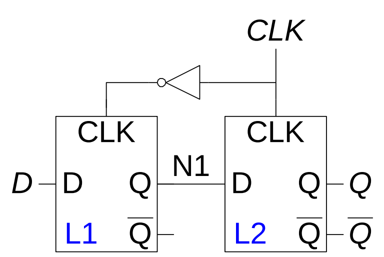
C_k	D	Q
↑	0	0
↑	1	1
0	x (штобило)	Q_{n-1} (меморира)
1	x	Q_{n-1}
↓	x	Q_{n-1}

Логичка табела

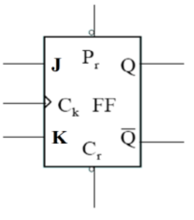
↑ Значи „предна ивица“ т.е. промена на влезот C_k од 0 на 1
(само тогаш влезот D се запишува и се пренесува на излезот Q)

D-флипфлоп интерна структура (информативно)

- Содржи два леча што работат со комплементарен такт:
  - При 0 на CLK отворен е L1 и го префрла D во N1
  - При 1 на CLK се затвора L1, а L2 го префрла N1 во Q
  - Со тоа D се појавува на Q само при премин на CLK од 0 на 1



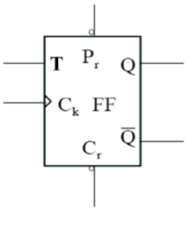
JK- флипфлоп (го решава проблемот со S=R=1)



J	K	Qn+
0	0	Qn
0	1	0
1	0	1
1	1	$\overline{Qn}$

- Влезовите делуваат само кога тактот е 1 (Ck=1)

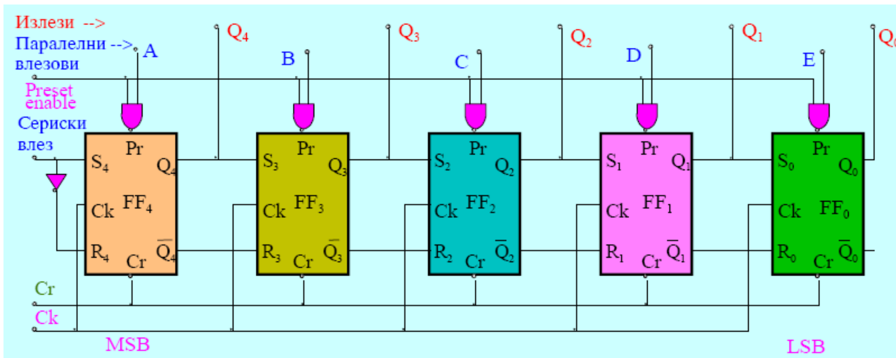
T- флипфлоп (toggle)



T	Qn+
0	Qn
1	$\overline{Qn}$

- Влезот делува само кога тактот е 1 (Ck=1)

Регистар со SR флипфлоп-ови (1)



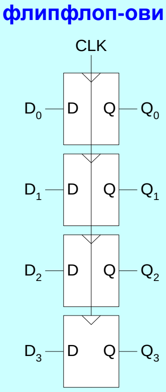
-Паралелен влез (A,B,C,D,E)
-Паралелен излез (Q_0, Q_1,Q_2, Q_3, Q_4)
-Сериски влез (S_4)
-Сериски излез (Q_0)

-Pr=Preset = паралелен асинхрон запис (на1)
-Preset Enable = овозможи паралелен запис
-Cr=Clear = асинхроно бришење (сите на 0)
-Ck = такт

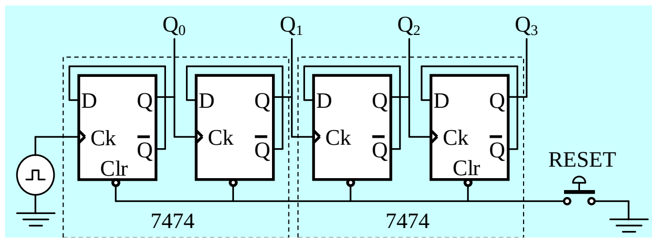
Регистар со SR флипфлоп-ови (2)

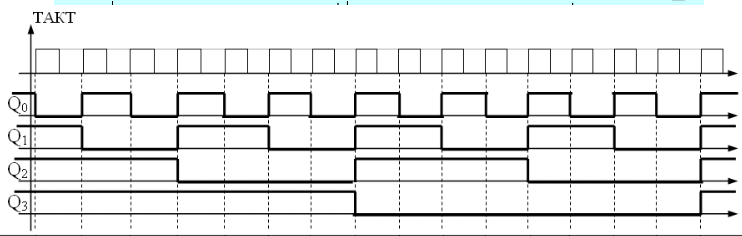
- Функции:
  - Паралелен влез + паралелен излез – за меморирање дигитални зборови
  - Сериски влез + паралелен излез – за прием на битови при сериска комуникација (RS232, USB, ETHERNET)
  - Паралелен влез + сериски излез – за испраќање на битови при сериска комуникација (RS232, USB, ETHERNET)

Паралелните регистри често се прават со D-флипфлоп-ови:



Бројач (четирибитен со D-флипфлоп-ови)

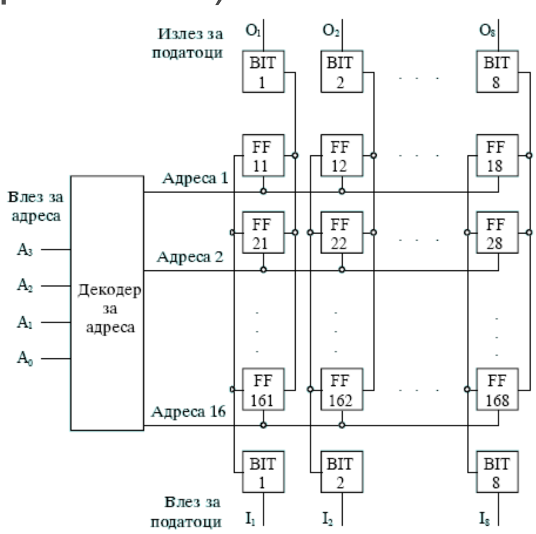




код	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000	1111
вред-ност	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15

Временски дијаграми, бинарни и декадни вредности

RAM-меморија (статичка) (информативно)



Генерална форма на секвенцијална мрежа – конечни автомати (FSM) (информативно)

- Три компоненти:
  - Регистер за меморирање на состојба
  - Комбинациона мрежа за пресметка на следна состојба
  - Комбинациона мрежа за пресметка на излезите
- Два вида автомати:
  - Муров (излезите зависат само од состојбата)
  - Милиев (излезите зависат и од состојбата и од влезовите)