Дигитална логика и системи

1

Вовед.
Бројни системи. Кодови (1*)

доц. д-р Никола Рендевски
nikola.rendevski@fikt.edu.mk

Летен семестар 2017/2018
ФИКТ, УКЛО, Битола

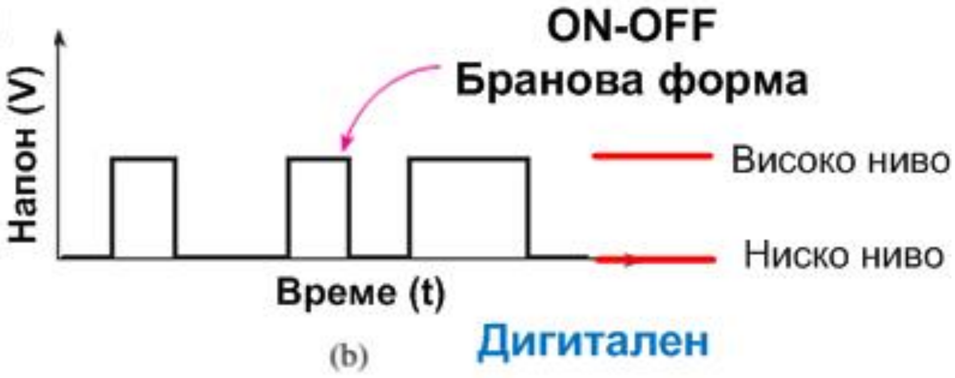
Цели на предметот

- Дигиталната логика е фундамент (основа) на компјутерските науки
- Најниско архитектурно ниво во дизајнот на хардвер, и основа во дизајнот на софтвер
- Познавањето на дигитална електроника и Булова алгебра (бинарни аритметичко логички операции) овозможува дизајн на дигитална логика за различна намена
- Дигитална логика:
  - Комбинациона логика (излезот е чиста логичка функција од тековната состојба/вредност на влезот)
  - Секвенцијална (меморија!) – излезот не зависи само од тековната состојба на влезот туку и од претходната состојба (од историјата на промена на влезните променливи)
- Анализа и синтеза на дигитални кола
- Хардверски описни јазици - HDL (VHDL и Verilog). Практичен дизајн (синтеза) на дигитални кола во FPGA (на вежби)
- Оценување
  - Два колоквиуми
  - Можност за изработка на семинарска работа (практична реализација на одредена дигитална архитектура, FPGA, Proteus, Fritzing, Arduino, изведба логика на протоборд со интегрални кола, итн.)
- Учете редовно! Предметната програма е обемна и бара сериозна ангажираност!
- **!!!!!! Напомена: Внимавајте на стари верзии од предавања по Дигитална техника. За учење користете исклучиво документи на кои семестарот и годината одговара на вашиот случај !!!!!!!!**

Вовед

- Дигиталната технологија <u>е основата во физичкиот дизајн</u> на модерните компјутерско-комуникациски системи
  - Компјутери, паметни телефони, периферни уреди
  - Системи за автоматско управување/контрола
  - Мрежни уреди, DSP системи...
- Дискусија*:
- Што значи терминот <u>дигитално</u>?
- Што е сигнал?
- Какви типови на сигнали постојат?
- Со какви сигнали/величини се карактеризира физичкиот свет околу нас

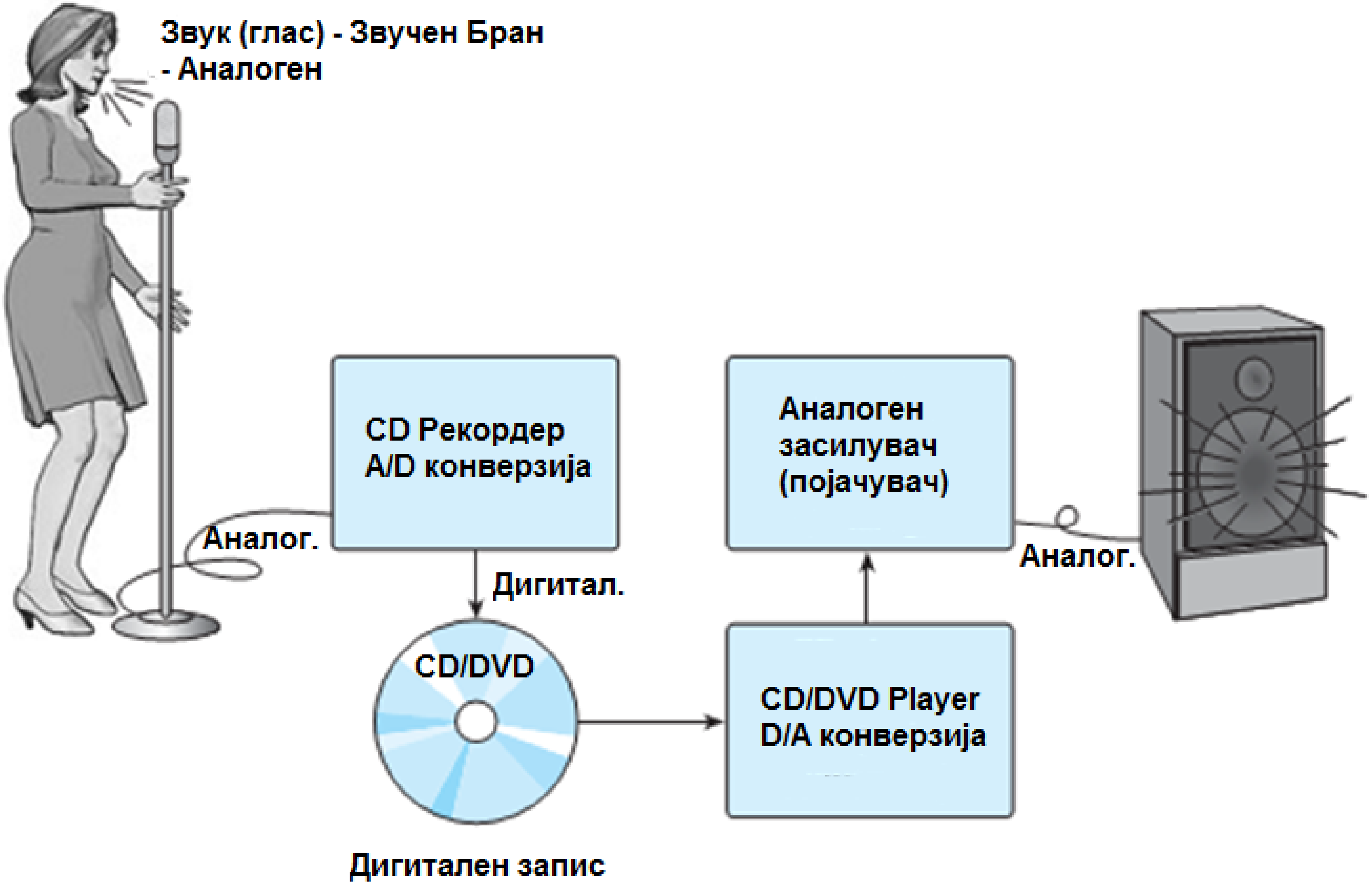
Аналогни и Дигитални сигнали



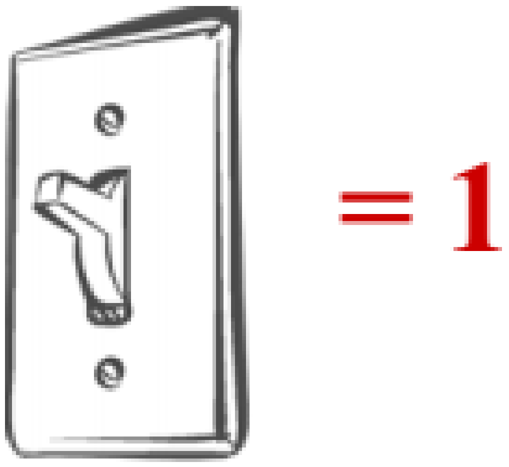
Зошто дигитални системи/сигнали?

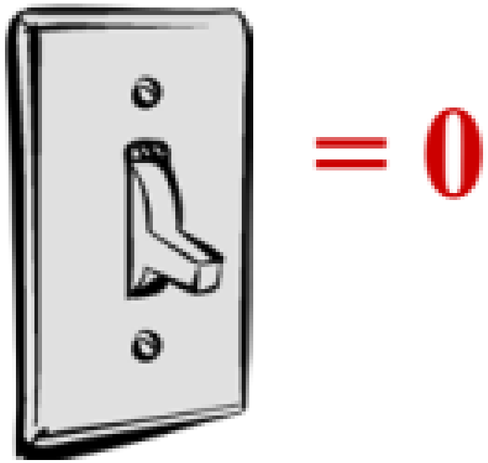
- Зошто ни е потребна дигитална презентација во „свет“ во кој сите сигнали се аналогни?
- Одговор: „Создадовме“ „машини“ кои интепретираат, комуницираат и зачувуваат (снимаат) податоци во „дигитален“ (бинарен) формат!
- Од кои причини?
- Дигиталните сигнали имаат добри **процесирачки** карактеристики, отпорни се на шумови (изобличувања)
- Полесно е да се обезбеди детекција на две нивоа (високо (1) и ниско (0) и покрај несовршеноста на сигналот.

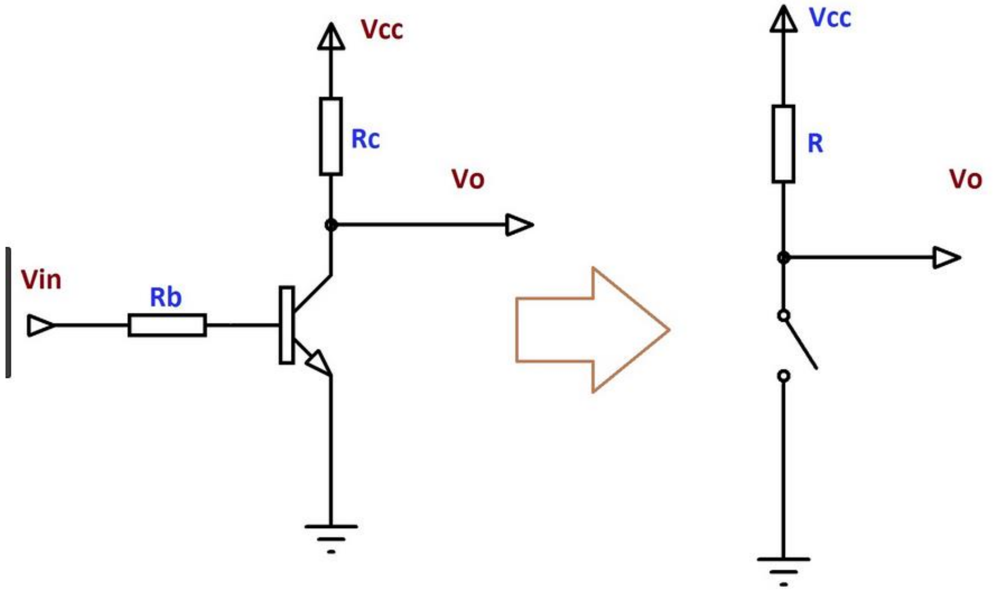
Аналогни и Дигитални сигнали



Зошто дигитални сигнали?

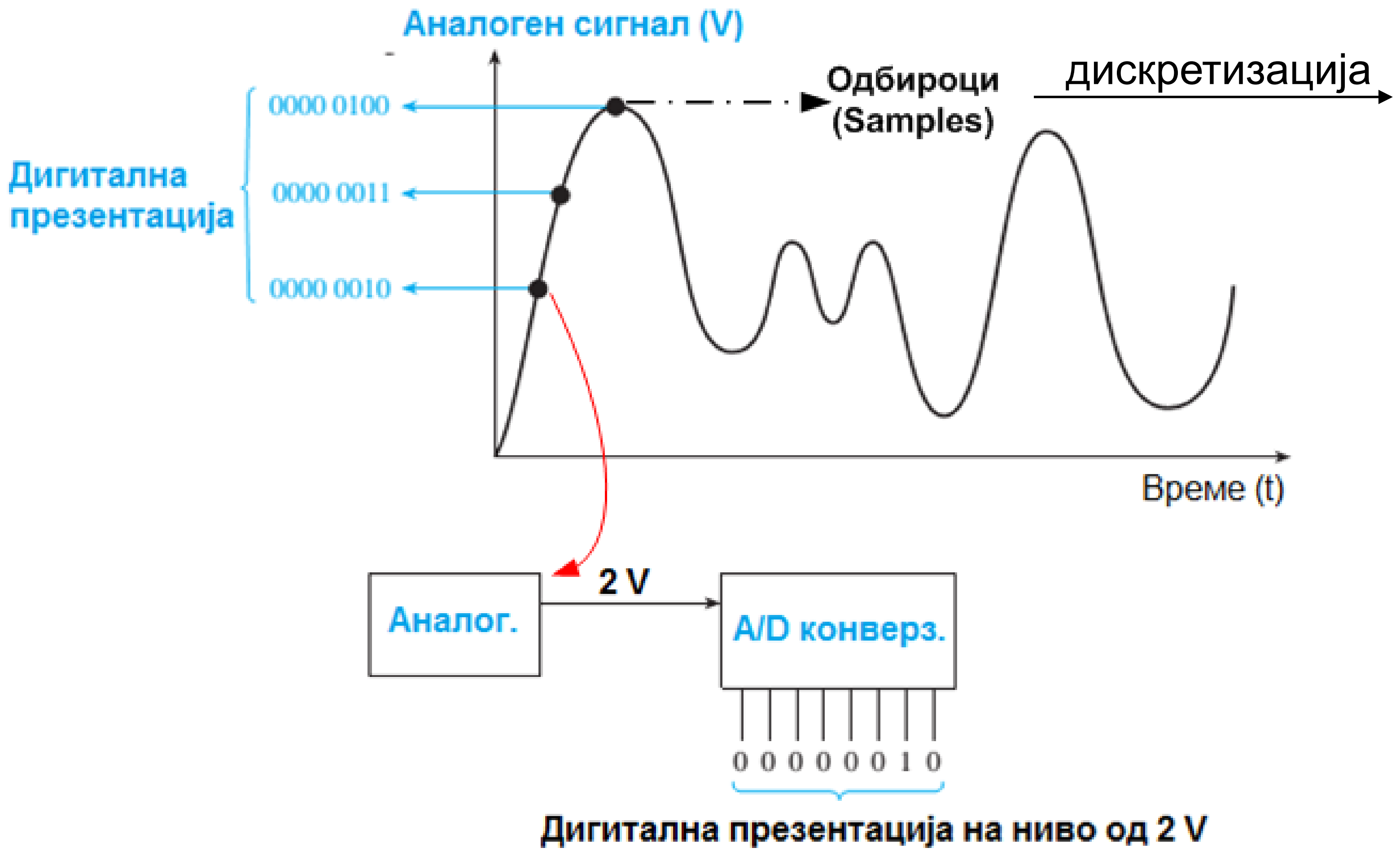


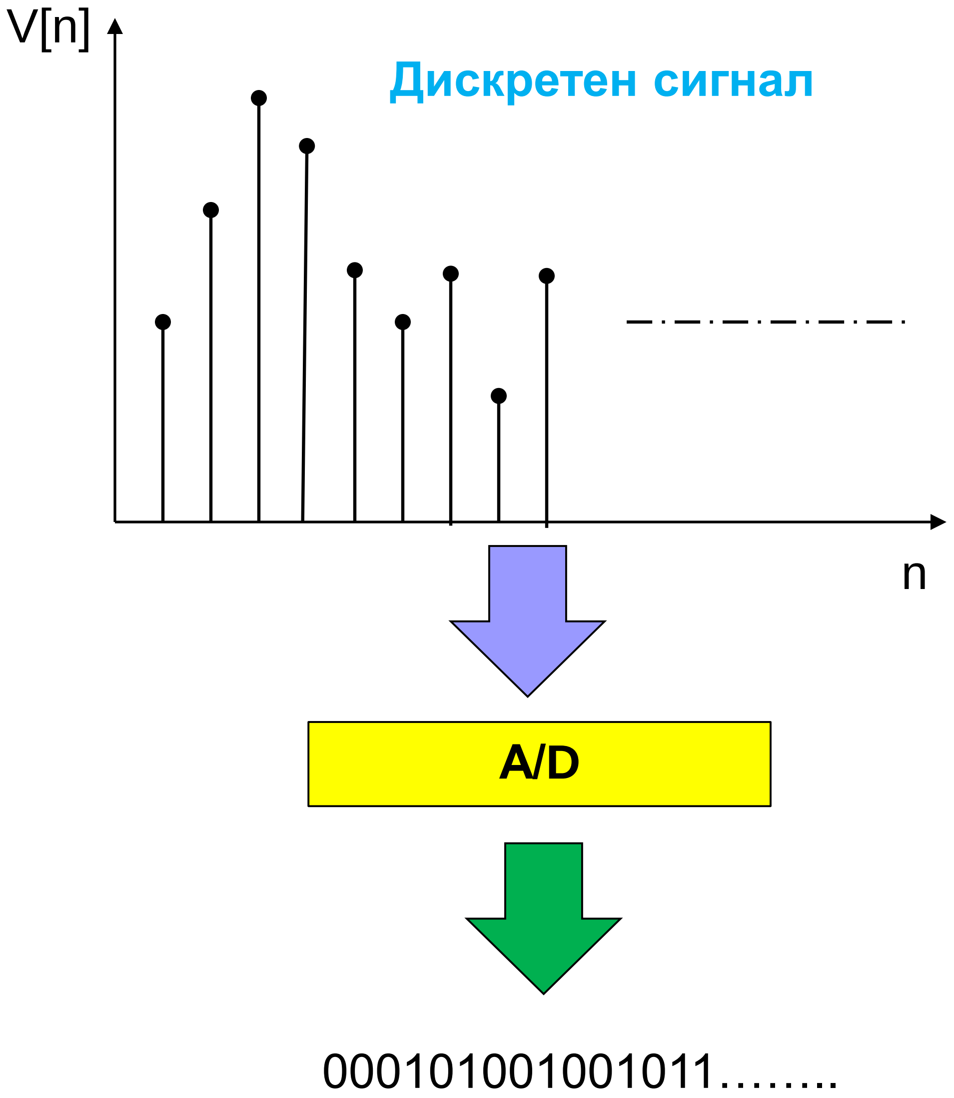




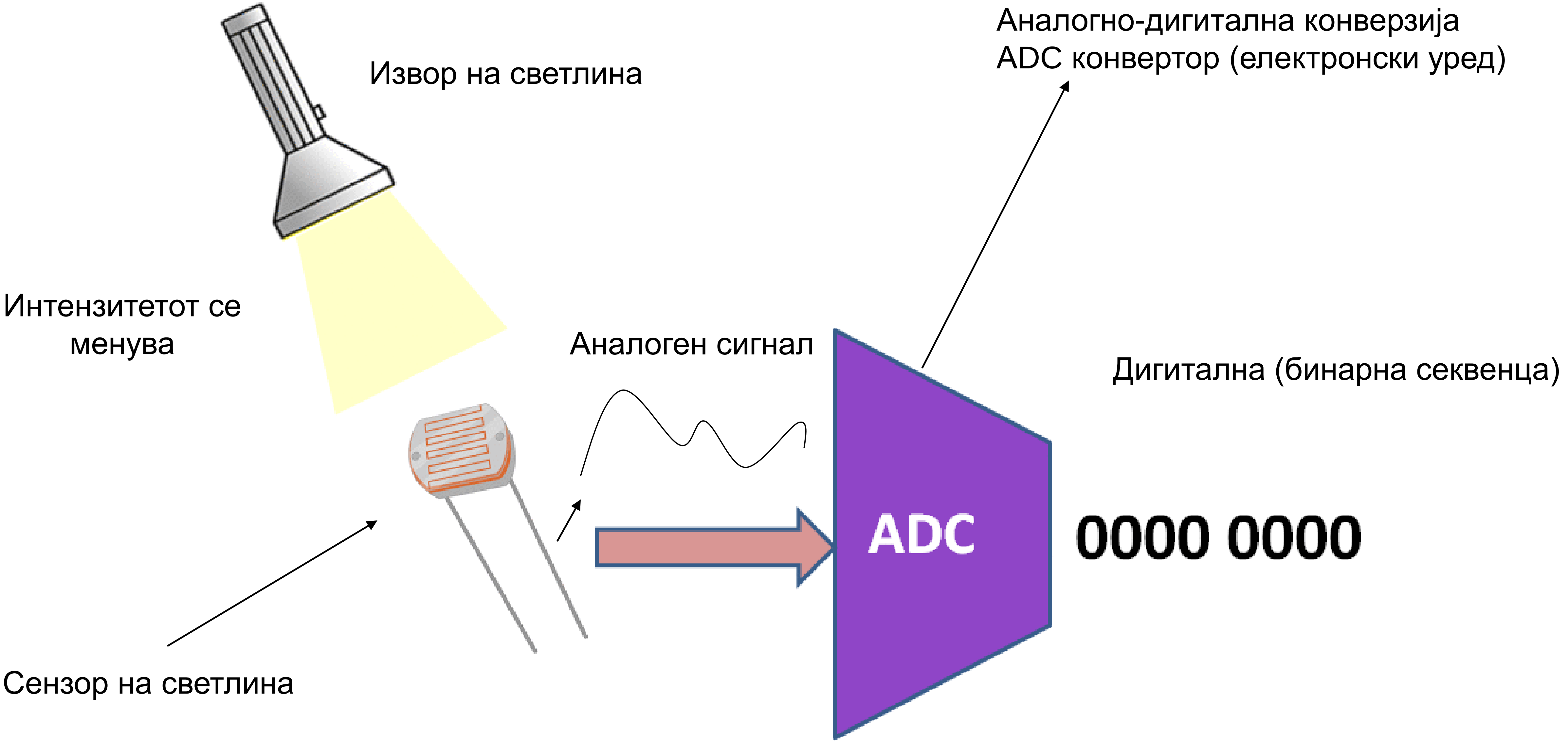
- Најпрост електричен механички прекинувач може да ги претстави двата симболи во бинарниот нумерички систем!
- **Транзистор** – основна градбена клетка на модерните VLSI интегрирани кола. При различни режими на работа одржува високо/ниско ниво помеѓу соодветни електроди

Аналогно-Дигитална конверзија A/D





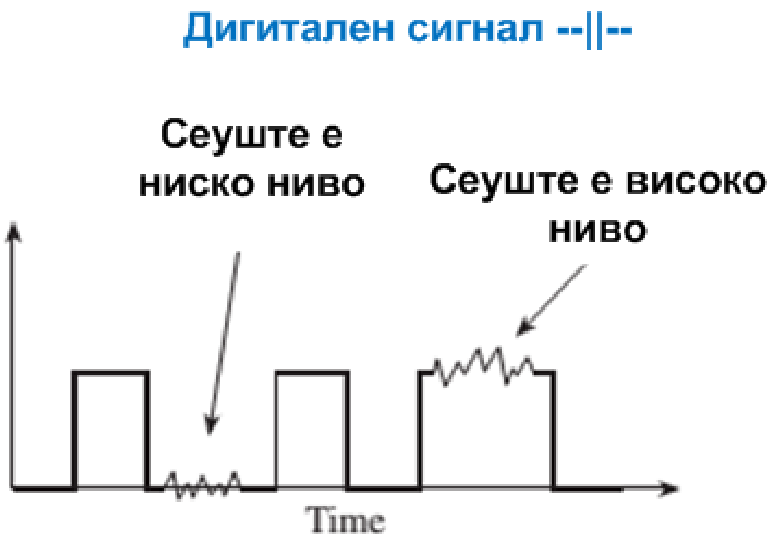
Анимација*



Извор на светлина
Аналогно-дигитална конверзија
ADC конвертор (електронски уред)
Интензитетот се менува
Аналоген сигнал
Дигитална (бинарна секвенца)
ADC
0000 0000
Сензор на светлина

Предности на дигиталните сигнали

- Не постои идеален пренос низ комуникациски канал (шум, интерференција , нелинеарности на опремата итн.)
- Дигиталните сигнали се далеку поотпорни на ефектите од шум/нелинеарности.

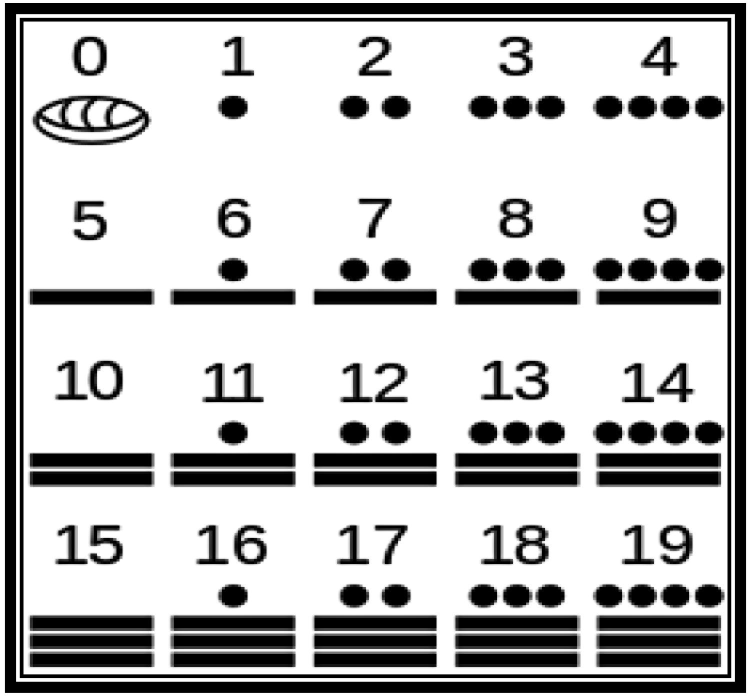


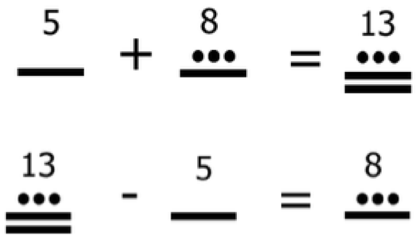
Бројни системи

- Основа при изучувањето на дигиталниот дизајн
- Бројните системи се математичка нотација за презентација на броеви (вредности) со помош на цифри или симболи на конзистентен начин.
- Во секојдневието користиме броен систем со основа 10,т.е **base-10** или т.н. **radix-10** броен систем (декаден) – 0,1,2,3,4,5,6,7,8,9
- Како би броеле во броен систем со основа 3,4,5...?
  - 0,1,2,10,11,12,20,21,22,100....... (**base-3**) – тернарен систем (**Ternary**)
  - 0,1,2,3,10,11,12,13,20,21,22,23,30,31,32,33,100……. **Quaternary (base-4)**
  - 0,1,2,3,4,10….,20,21,………. 33,34…….44,100….(base-5) **Quinary или Pental**
  - **Base-6: Senary или Heximal**
  - **Base-7: Septenary**
  - ……………………

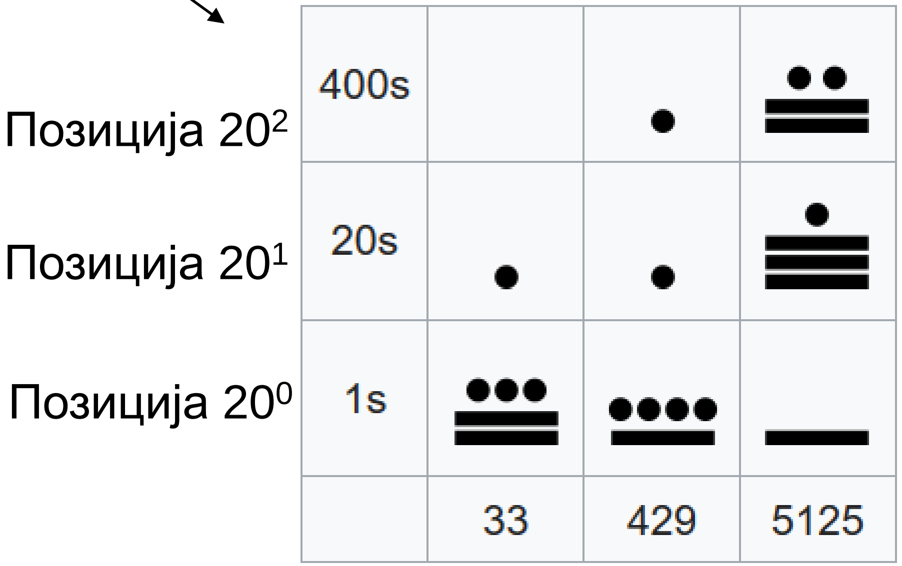
Броен систем на Маите (информативно)

- 1000 години пред Европската цивилизација (base-20, позиционен). Се нарекува Vigesimal Number System.





Повеќе-цифрените броеви се пишувале вертикално Секоја вертикална позиција е степен од основата



Бројни системи - Нотација

- Во секој броен систем постои правило за презентација на број (вредност) од постоечките симболи – нотација
- Постојат **позициони** и **непозициони** бројни системи
- Пример за непозиционен броен систем
  - **Римски нумерички систем** – базиран на 7 симболи (I,V,X,L,C,D,M) кои одговараат на вредностите (1,5,10,50,100,500, 1000)
  - **MCMXCVI = 1996, но MM = 2000!?!?**
  - **X • C = M, L/V=X (Комплексно? Непрактично!)**
- „Подобро“ т.е. попрактично решение за реализација на математички операции: **Позиционен** броен систем
  - Декаден, бинарен, октален, хексадецимален
  - Основа (base) или радикс (radix)

Б.С – Позициона нотација

- Секој дигит (цифра) на позиција се репрезентира со множење со **основата на степен** од таа позиција
- Base 10, Radix-10
- Децимален борен систем: 0,1,2,3,4,5,6,7,8,9
- Во *Base-b* систем, основата *b* претставува бројот на првите *b* природни броеви и нулата (0). Генералната форма на позиционен броен систем е:

$$(a_n a_{n-1} \cdots a_1 a_0 . c_1 c_2 c_3 \cdots)_b = \sum_{k=0}^{n} a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}.$$

- Позицијата k е логаритам од соодветната тежина b^k
- $k=\log_b(b^k)$

Б.С – Позициона нотација

Во 4-цифрен децимален број, најмалку-значајната позиција (најдесно) има тежински фактор од 10^0 додека најзначајната позиција (најлево) има тежински фактор од 10^3

10^3	10^2	10^1	10^0

Каде

$$10^3 = 1000$$
$$10^2 = 100$$
$$10^1 = 10$$
$$10^0 = 1$$

За да се евалуира декадниот број 4623, секој дигит се множи со соодветниот тежински фактор

4 6 2 3

$$\begin{aligned} 3 \times 10^0 &= \quad 3 \\ 2 \times 10^1 &= \quad 20 \\ 6 \times 10^2 &= \quad 600 \\ 4 \times 10^3 &= \underline{+4000} \\ & \quad\ 4623 \end{aligned}$$

Резултат

Бинарен броен систем

"There are 10 kinds of people, those who understand binary and those who don't."

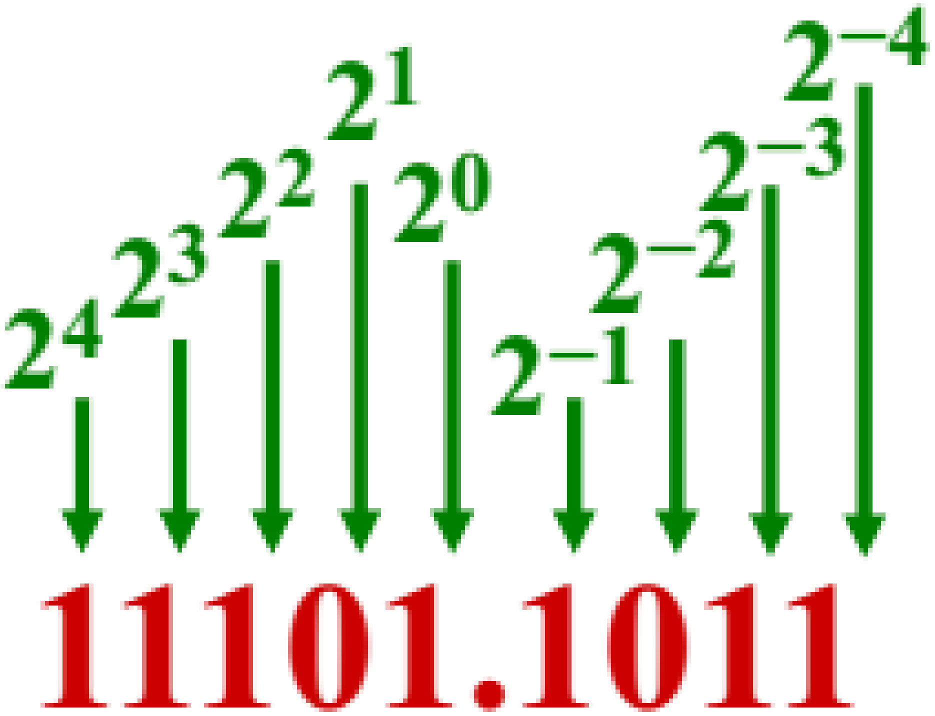
- Дигиталните системи користат бинарен броен систем (**1** и **0**)
- Двете цифри (дигити) можат да се претставаат со две различни напонски нивоа: Пр. +5 V = **1**, ~0 V= **0**
- Тежински фактори во бинарниот систем:

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

$$\begin{aligned} 2^0 &= 1 \\ 2^1 &= 2 \\ 2^2 &= 4 \\ 2^3 &= 8 \\ 2^4 &= 16 \\ 2^5 &= 32 \\ 2^6 &= 64 \\ 2^7 &= 128 \end{aligned}$$

Бинарен броен систем

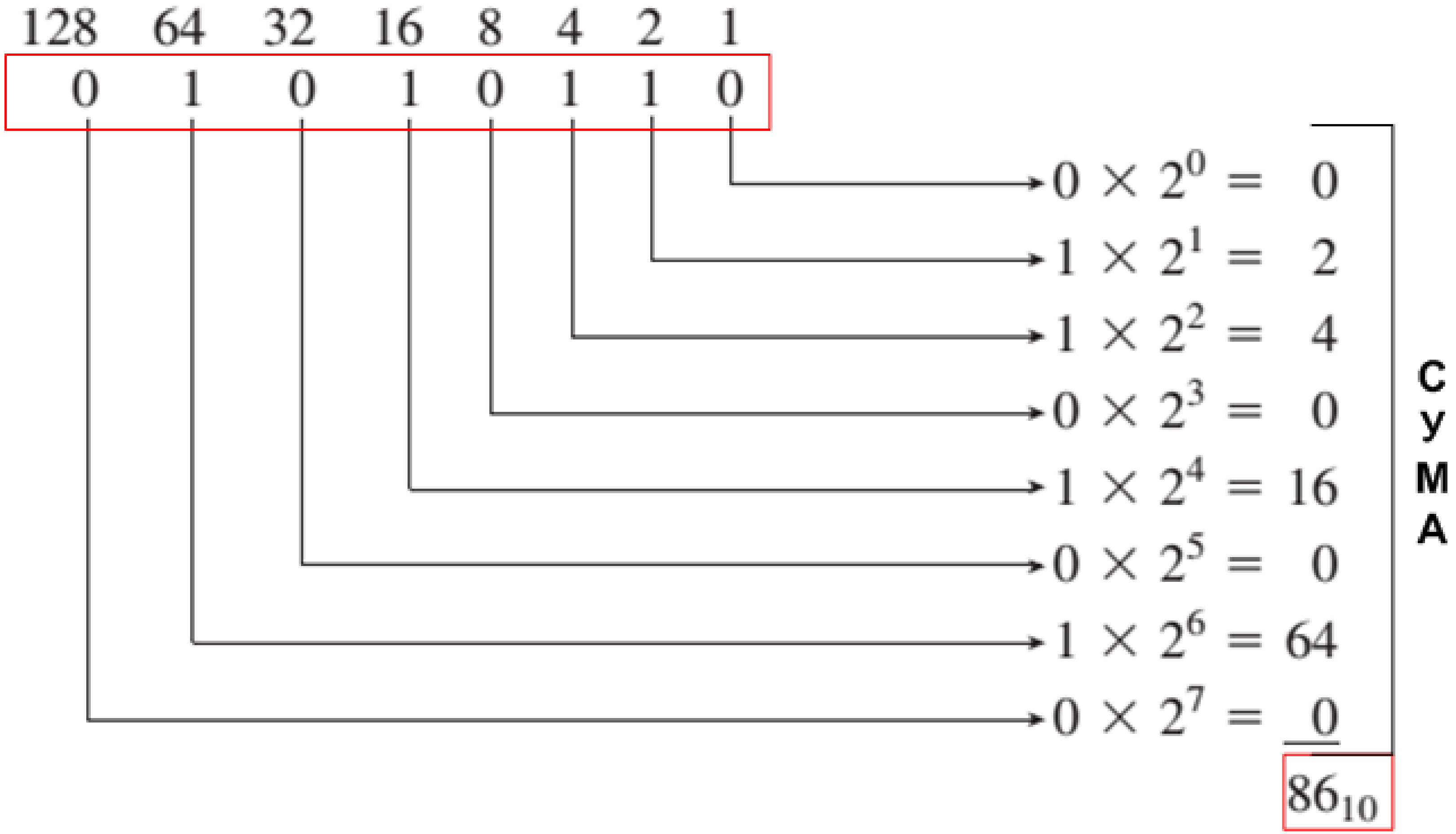
- Бинарниот систем е позиционен
- Секоја позиција има тежинска вредност 2^k
- k=0,1,2….n



$$(a_n a_{n-1} \cdots a_1 a_0 . c_1 c_2 c_3 \cdots)_b = \sum_{k=0}^{n} a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}.$$

Бинарен броен систем

$$(a_n a_{n-1} \cdots a_1 a_0 . c_1 c_2 c_3 \cdots)_b = \sum_{k=0}^{n} a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}.$$



Конверзија од бинарен во декаден броен систем

- Да се изврши конверзија на бинарниот број **1011.1010** во декаден
- Го множиме секој дигит со соодветниот тежински фактор и ги собираме ненултите позиции
- Позициите со 0 ги изоставаме!

$$1\ 0\ 1\ 1\ .\ 1\ 0\ 1\ 0$$

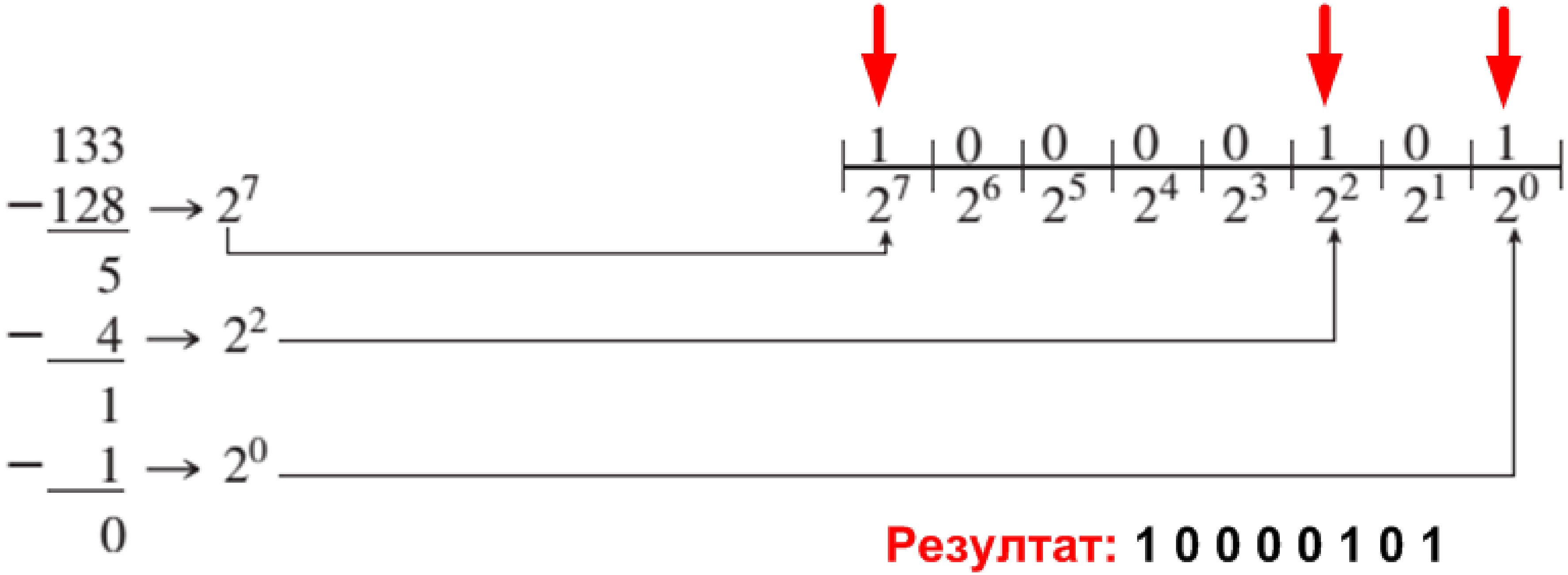
$$\begin{aligned}
1 \times 2^{-3} &= 0.125 \\
1 \times 2^{-1} &= 0.500 \\
1 \times 2^{0} &= 1 \\
1 \times 2^{1} &= 2 \\
1 \times 2^{3} &= \underline{8} \\
& 11.625_{10}
\end{aligned}$$

Конверзија од декаден во бинарен

- Генерално постојат два методи
  - **Метод 1.** Барање на најголемиот најблизок тежински фактор до бројот, а потоа сукцесивно одземање до 0
  - **Метод 2.** Сукцесивно делење со два

Метод 1.

- Да се претвори 133_{10} во бинарен број



Резултат: 1 0 0 0 0 1 0 1

Метод 2. (Сукцесивно делење со основата (2))

- Да се претвори 152_{10} во бинарен броен систем

$152 \div 2 = 76$	Остаток	0	(LSB)
$76 \div 2 = 38$	Остаток	0	
$38 \div 2 = 19$	Остаток	0	
$19 \div 2 = 9$	Остаток	1	
$9 \div 2 = 4$	Остаток	1	
$4 \div 2 = 2$	Остаток	0	
$2 \div 2 = 1$	Остаток	0	
$1 \div 2 = 0$	Остаток	1	(MSB)

Резултат: 1 0 0 1 1 0 0 0

Конверзија на децимален декаден во бинарен

- Пример: Да се претвори 0.375_{10} во бинарен
- Начин: Сукцесивнно множење со 2 на делот после запирката (Fraction) се додека не се елиминира, или не детектираме периодично повторување.
- <u>Секоја позиција во бинарната фракција е позначајна во споредба со декадната</u>!

0.375: $0.375 \times 2 = 0.\underline{75}$

$0.75 \times 2 = 1.\underline{5}$

$0.5 \times 2 = 1.0$

$0.375_{10} = 0.011_2$.

Конверзија на децимален декаден во бинарен

0.427:

$$0.427 \times 2 = 0.854$$
$$0.854 \times 2 = 1.708$$
$$0.708 \times 2 = 1.416$$
$$0.416 \times 2 = 0.832$$
$$0.832 \times 2 = 1.664$$
$$0.664 \times 2 = 1.328$$
$$0.328 \times 2 = 0.656$$
$$0.656 \times 2 = 1.312$$

$$0.427_{10} \approx 0.01101101_2$$

0.1 во бинарен

0.1 * 2 = 0.2 -> 0

0.2 * 2 = 0.4 -> 0

0.4 * 2 = 0.8 -> 0

0.8 * 2 = 1.6 -> 1

0.6 * 2 = 1.2 -> 1

0.2 * 2 = 0.4 -> 0

0.4 * 2 = 0.8 -> 0

0.8 * 2 = 1.6 -> 1

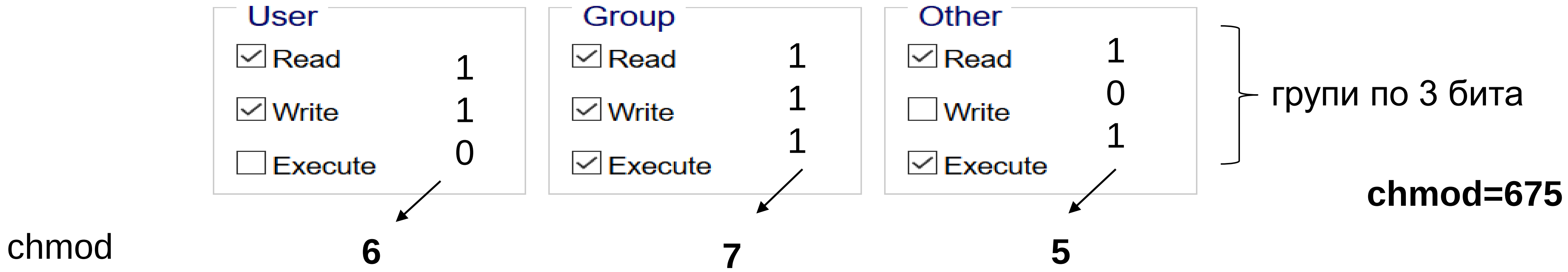
0.6 * 2 = 1.2 -> 1

Периодично повторување

Резултат: 0.00011(0011) периода.

Октален броен сисем (Base 8)

- Окталниот броен систем е метод за групирање на бинарни броеви во група од 3 бита
- 8 цифри: 0,1,2,3,4,5,6,7
- Една примена на окталниот броен систем: се користи во т.н 3-битни кодови кои означуваат инструкции или привилегии (read/write/execute). Пример кај Unix (Linux):



Декаден	Бинарен	Октален
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12

Октален броен систем - Конверзии

- Да се претвори 624_8 во бинарен систем
- Секоја цифра се претставува со 3 бита

$$\overbrace{1\ 1\ 0}^{6}\ \overbrace{0\ 1\ 0}^{2}\ \overbrace{1\ 0\ 0}^{4} = 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0_2$$

- Да се претвори 326_8 во декаден систем
  - Секоја цифра се множи со тежински фактор (со основа 8 и експонент позиција)

3 2 6

$$\begin{aligned}
6 \times 8^0 &= 6 \times 1 = 6 \\
2 \times 8^1 &= 2 \times 8 = 16 \\
3 \times 8^2 &= 3 \times 64 = \underline{192} \\
& \quad 214_{10}
\end{aligned}$$

Октален броен систем - Конверзии

- Претворање на декаден во октален броен систем
- Делиме со основата (8) „до крај“
- Да се претвори 486_{10} во октален

$486 \div 8 = 60$	Остаток	6	
$60 \div 8 = 7$	Остаток	4	746_8
$7 \div 8 = 0$	Остаток	7	

$$486_{10} = 746_8$$

Проверка:

7 4 6

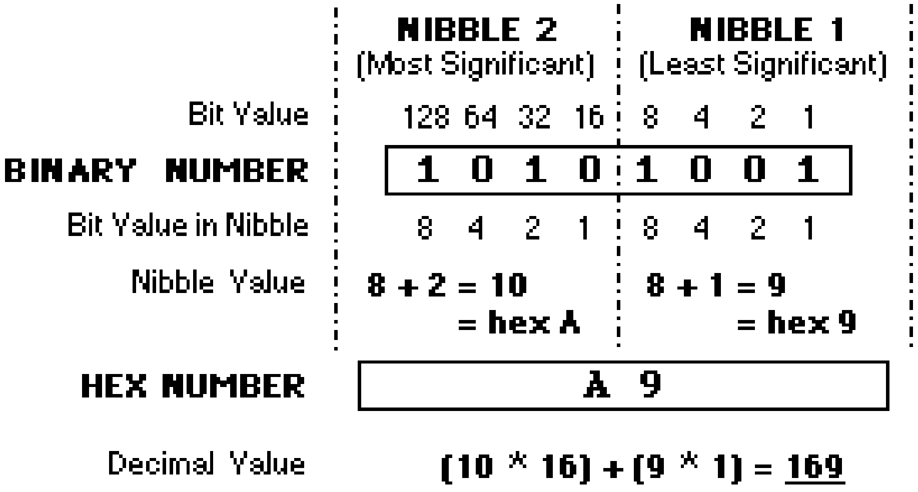
$$6 \times 8^0 = 6$$
$$4 \times 8^1 = 32$$
$$7 \times 8^2 = \underline{448}$$
$$486 \quad \surd$$

Хексадецимален (Base 16)

- Слично како кај окталниот, Хексадецималниот систем е метода за упростување на пристапот до адресите и читањето на инструкциите во компјутерскиот систем.
- Групирање на 4 бита (2^4=16)
- Затоа инструкциите или податоците во 8-,16- и 32-битните системи можат да се претставаат како 2,4 или 8-дигитен хексадецимален код
- 16 различни дигити
- Дигити: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**, A, B, C, D, E, F
- **Вредностите во основниот систем поголеми од 9 се претставени со букви A-F**

Хексадецимален (Base 16)

- Со цел да се потенцира Хексадецимална презентација, се индицира со индекс 16 или најчесто во компјутерската техника со буквата H. Пример: $A1B_H$ или $A1B_{16}$
- Со два хексадецимални дигити може да се претстави 1 Бајт (1Byte=8bits)
- 1 хексадецимален дигит – група од 4 бита се вика nibble (нибл)



Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	1 0
17	0001 0001	1 1
18	0001 0010	1 2
19	0001 0011	1 3
20	0001 0100	1 4

Хексадецимален - Конверзии

- Да се претвори бинарниот број 01101101 во хексадецимален

$$\underbrace{0\;1\;1\;0}_{6}\quad\underbrace{1\;1\;0\;1_2}_{D}\quad = 6D_{16}$$

- Да се претвори хексадецималниот $A9_{16}$ во бинарен

$$\overbrace{1\;0\;1\;0}^{A}\quad\overbrace{1\;0\;0\;1}^{9} = 10101001_2$$

Хексадецимален - Конверзии

- Да се претвори хексадецималниот број $2A6_{16}$ во декаден
  - Секоја цифра се множи со тежински фактор (со основа 16 и експонент позицијата)

$$2 \quad A \quad 6$$

$$
\begin{aligned}
6 \times 16^0 &= 6 \times 1 = 6 \\
A \times 16^1 &= 10 \times 16 = 160 \\
2 \times 16^2 &= 2 \times 256 = \underline{512} \\
& \qquad\qquad\quad 678_{10}
\end{aligned}
$$

Хексадецимален - Конверзии

- Да се претвори декадниот број 151_{10} во хексадецимален

$151 \div 16 = 9$	Остаток	7	(LSD)
$9 \div 16 = 0$	Остаток	9	(MSD)

$$151_{10} = 97_{16}$$

Least Significant Digit

Most Significant Digit

Проверка:

97_{16}

$$7 \times 16^0 = 7$$
$$9 \times 16^1 = 144$$
$$+ \quad 151 \quad \surd$$

Бинарни кодови (1 дел)

- Бинарниот броен систем е основата на денешната дигитална електроника. За да претставиме било каква информација (не само број/вредност) потребно ни е т.н. бинарно кодирање.
- Доколку претставуваме само вредност/број во бинарен облик, тогаш тоа е т.н. Straight Binary Code (вредност конвертирана во бинарен броен систем)
- Дигиталните кодови претставуваат симболичка репрезентација на информација која може да биде во форма на нумерички, алфанумерички и специјални карактери
- Бинарниот код претставува група од n битови кои претставуваат единствен симбол (број, буква, специјален карактер, инструкција...)
- Интерпретацијата на информација во бинарна форма е можна само ако точно се знае кодот со кој е претставена.
- Постојат различни типови на кодови со различна намена
- *****!** <u>Анализата и изучувањето на бинарните кодови бара познавања и од бинарни аритметички операции и концепти кои ќе ги изучуваме понатаму, и следствено ќе се навраќаме на нивната примена во реализацијата на бинарните кодови</u> *******

BCD (Binary-Coded-Decimal)

- NBCD – Natural BCD кодот претставува метод на кодирање на сите 10 дигити од декадниот систем со 4-битен бинарен код.
- Да се претстави 496_{10} во BCD

$$\overbrace{0100}^{4}\ \overbrace{1001}^{9}\ \overbrace{0110}^{6} = 0100\ \ 1001\ \ 0110_{BCD}$$

- Да се претвори $0111\ 0101\ 1000_{BCD}$ во декаден

$$\underbrace{0111}_{7}\ \underbrace{0101}_{5}\ \underbrace{1000}_{8} = 758_{10}$$

Decimal	Binary	Octal	Hexadecimal	BCD
0	0000	0	0	0000
1	0001	1	1	0001
2	0010	2	2	0010
3	0011	3	3	0011
4	0100	4	4	0100
5	0101	5	5	0101
6	0110	6	6	0110
7	0111	7	7	0111
8	1000	1 0	8	1000
9	1001	1 1	9	1001
10	1010	1 2	A	0001 0000
11	1011	1 3	B	0001 0001
12	1100	1 4	C	0001 0010
13	1101	1 5	D	0001 0011
14	1110	1 6	E	0001 0100
15	1111	1 7	F	0001 0101
16	0001 0000	2 0	1 0	0001 0110
17	0001 0001	2 1	1 1	0001 0111
18	0001 0010	2 2	1 2	0001 1000
19	0001 0011	2 3	1 3	0001 1001
20	0001 0100	2 4	1 4	0010 0000

ASCII Код

- При манипулацијата со податоците од и кон компјутерскиот систем не ни се доволни само нумерички репрезентации
- Секако дека постојат и други карактери како букви (големи, мали), специјални знаци итн.
- Имајќи во предвид дека дигиталниот систем работи само со 1 и 0, се јавува потреба од код со кој ќе се претставаат сите алфа-нумерички податоци (букви, броеви, специјални знаци – симболи)

ASCII Код

- Компјутерската индустрија воспостави/прифати влезно-излезен код -**American Standard Code for Information Interchange (ASCII)**
- ASCII кодот користи **7 бита** да ги претстави сите алфанумерички карактери во I/O операциите
- Со користење на 7 бита се воспоставуваат 128 различни кодни комбинации кои се нарекуваат кодни точки или кодни позиции (code point, code position)
- 0_H до $7F_H$

ASCII Табела

LSB \ MSB	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC_1	!	1	A	Q	a	q
0010	STX	DC_2	”	2	B	R	b	r
0011	ETX	DC_3	#	3	C	S	c	s
0100	EOT	DC_4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	—	o	DEL

	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT	DC4	$	4	D	T	d	t
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D	] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F

M

004D

Code point for *M.*
But "**004D**"?

Тестирајте ги ASCII кодовите во Word. Напишете го ASCII кодот на карактерот, а потоа притиснете Alt+X

Пример 004D -> Alt+X = M

C++ (конверзија на карактер во ASCII)

```
#include <iostream>
using namespace std;

int main()
{
 char c;
 cout << "Enter a character: ";
 cin >> c;
 cout << "ASCII Value of " << c << " is " << int(c);
 return 0;
}
```

Контролни I/O операции

Definitions of control abbreviations:

ACK	Acknowledge	FS	Form separator
BEL	Bell	GS	Group separator
BS	Backspace	HT	Horizontal tab
CAN	Cancel	LF	Line feed
CR	Carriage return	NAK	Negative acknowledge
DC_1–DC_4	Direct control	NUL	Null
DEL	Delete idle	RS	Record separator
DLE	Data link escape	SI	Shift in
EM	End of medium	SO	Shift out
ENQ	Enquiry	SOH	Start of heading
EOT	End of transmission	SP	Space
ESC	Escape	STX	Start text
ETB	End of transmission block	SUB	Substitute
ETX	End text	SYN	Synchronous idle
FF	Form feed	US	Unit separator
		VT	Vertical tab

Кодови (1)...

- Класификација на кодовите:
  - Тежински и не-тежински (Weighted and Non-Weighted)
    - Тежински – BCD, 8421, 5211,2421,3321, 4321
    - Не-тежински - Excess-3, Gray
  - Само-комплементарачки и секвенцијални
    - Само-комплементирачки: 5211,2421,3321, 4321
  - Алфанумерички кодови: ASCII (7bit, 8 bit), EBCDIC (8bit)
  - Кодови за детекција и корекција на грешка (Се изучуваат и во предметот Архитектура на компјутери - проф. Пеце Митревски) – пример (Hamming Code)
- Како што веќе напомнавме, реализацијата на кодовите т.е. кодирањето бара одредени познавања од бинарни операции и дигитални кола. Затоа во текот на курсот во неколку наврати ќе ги споменуваме.