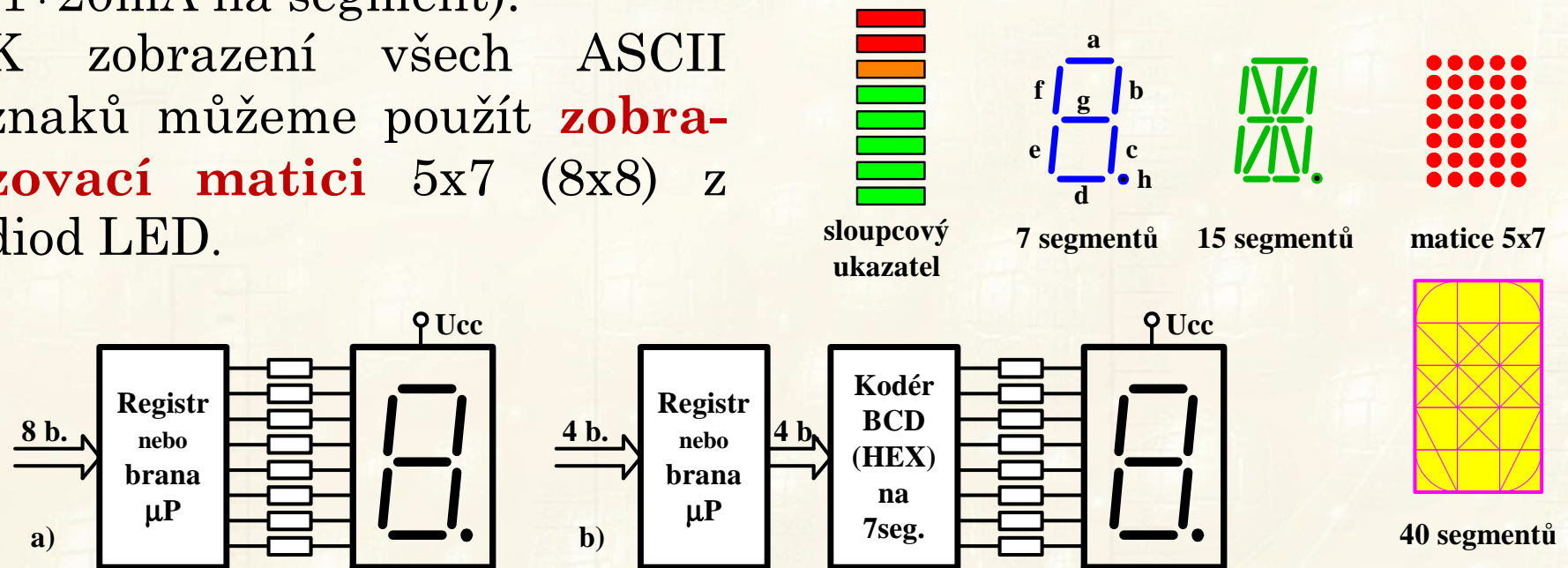


Zobrazovací jednotky

ZOBRAZOVACÍ JEDNOTKY LED – STATICKÉ ŘEŠENÍ

- ❖ **Dioda LED** - spínaná nebo zkratovaná výstupem brány procesoru. Proud diodou 1÷20mA, úbytek na diodě 2,1÷3,3V.
- ❖ **Sloupcový ukazatel** – hrubší zobrazení analogové nebo číslicové veličiny. Obsluha stejná jako u segmentových zobrazovačů nebo obvodem A277D.
- ❖ Zobrazování čísel a znaků - **sedmisegmentové** nebo **patnácti-segmentové zobrazovače**. Relativně velké proudové nároky (1÷20mA na segment).

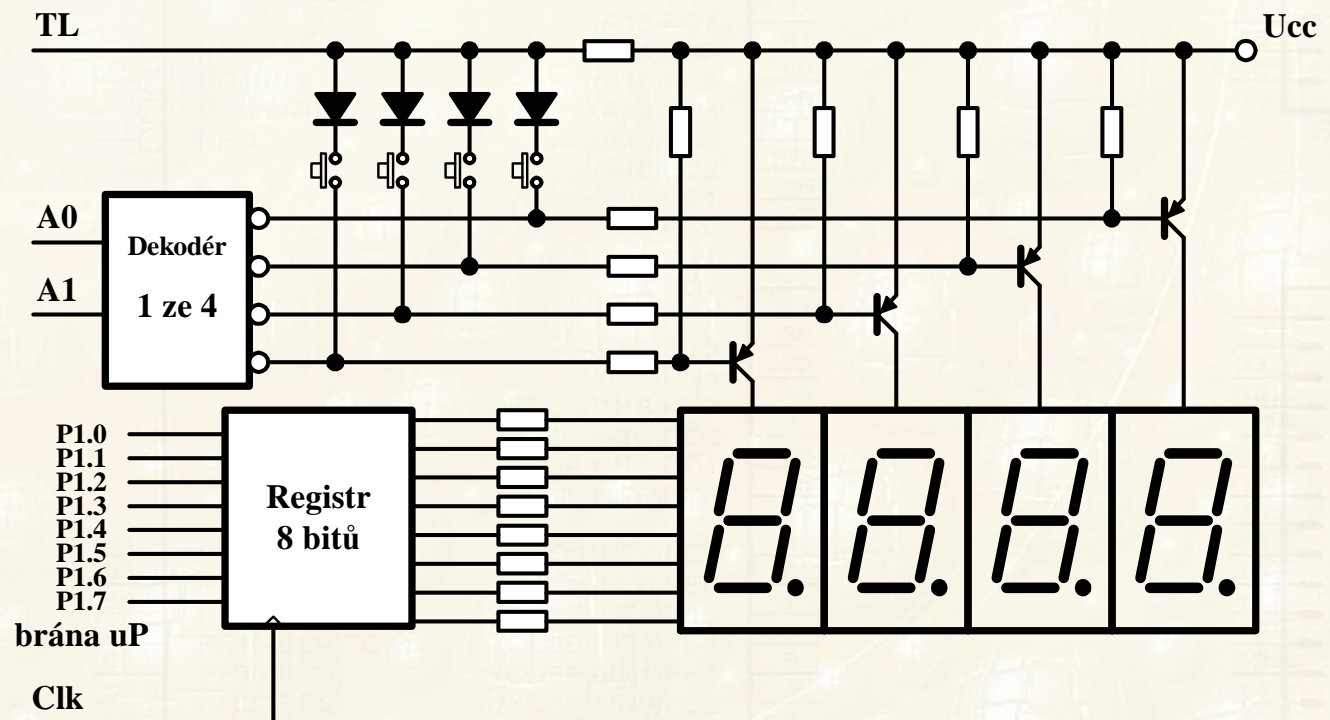
K zobrazení všech ASCII znaků můžeme použít **zobrazovací matici** 5x7 (8x8) z diod LED.



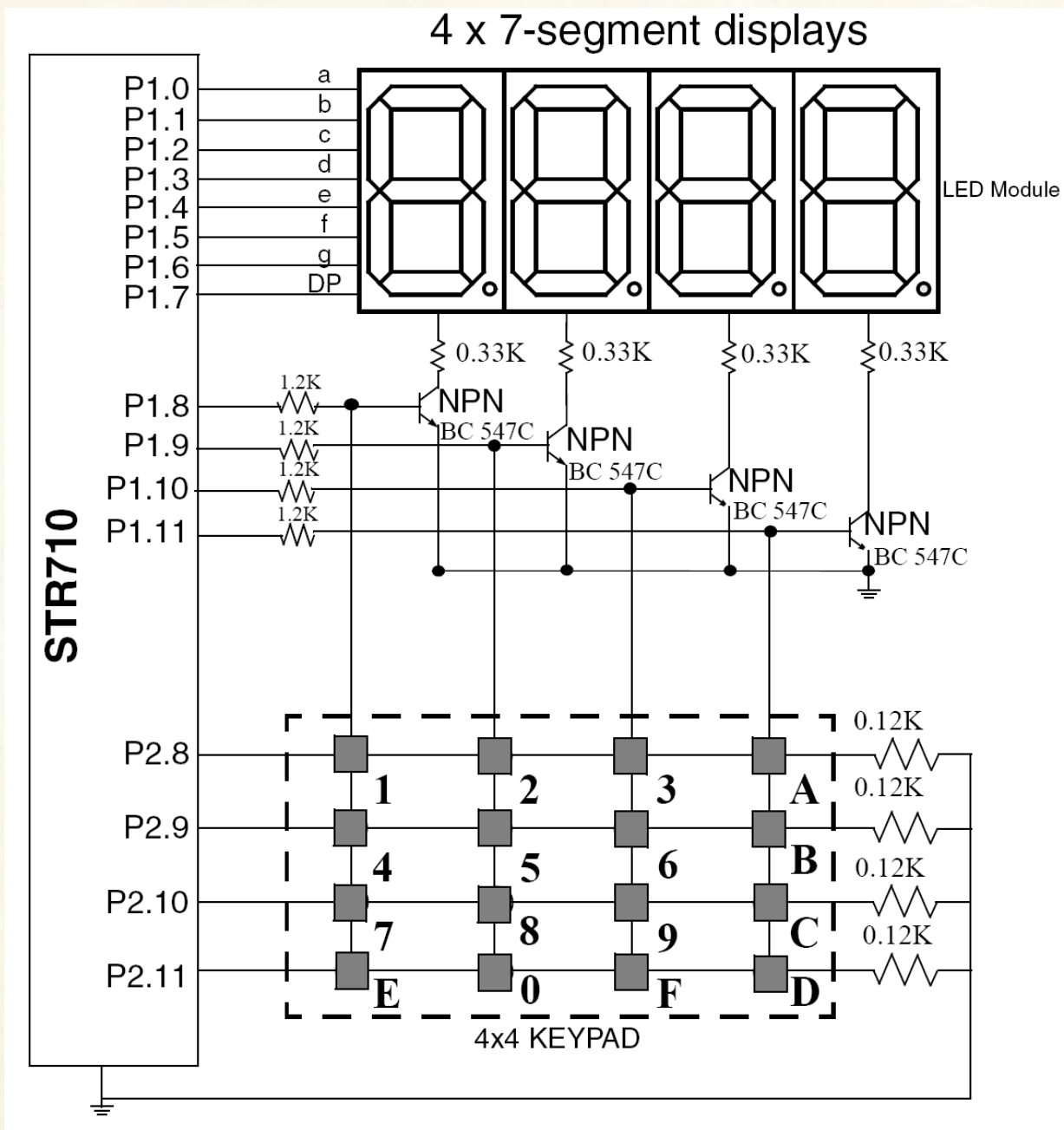
ZOBRAZOVACÍ JEDNOTKY LED – DYNAMICKÉ ŘEŠENÍ

Cenově levnějším zapojením segmentovek s využitím časového multiplexu je **dynamické zobrazování**.

- ❖ Zmenšení počtu potřebných vodičů k řízení displeje (minimálně $8+2(3)$).
- ❖ Jas segmentu určuje střední hodnota proudu, ale účinek není zcela stejný jako při statickém řešení.
- ❖ Frekvence spínání segmentu minimálně 75Hz.
- ❖ Impulzní proudy při realizaci na plošném spoji se 2 vrstvami mohou vést k problémům při schvalování EMC.



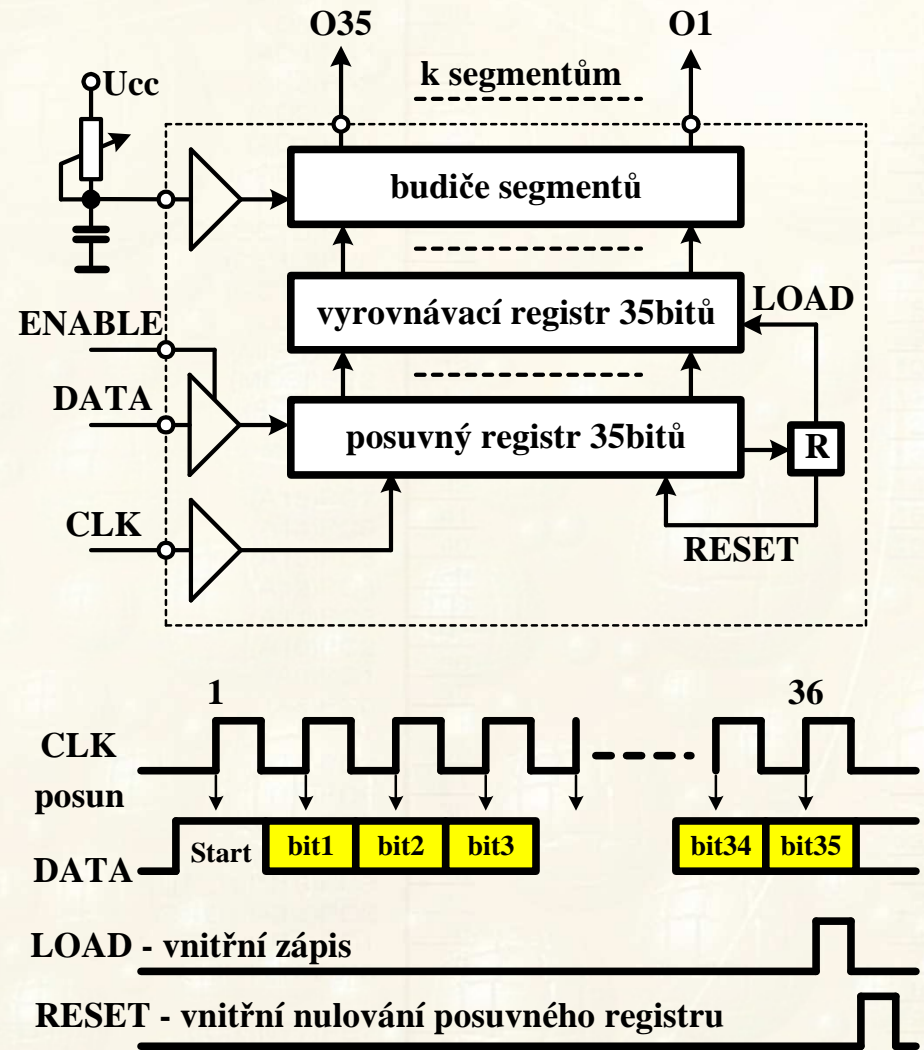
ZOBRAZOVACÍ JEDNOTKY LED – DYNAMICKÉ ŘEŠENÍ



ZOBRAZOVACÍ JEDNOTKY LED – SÉRIOVÉ STATICKÉ ŘEŠENÍ

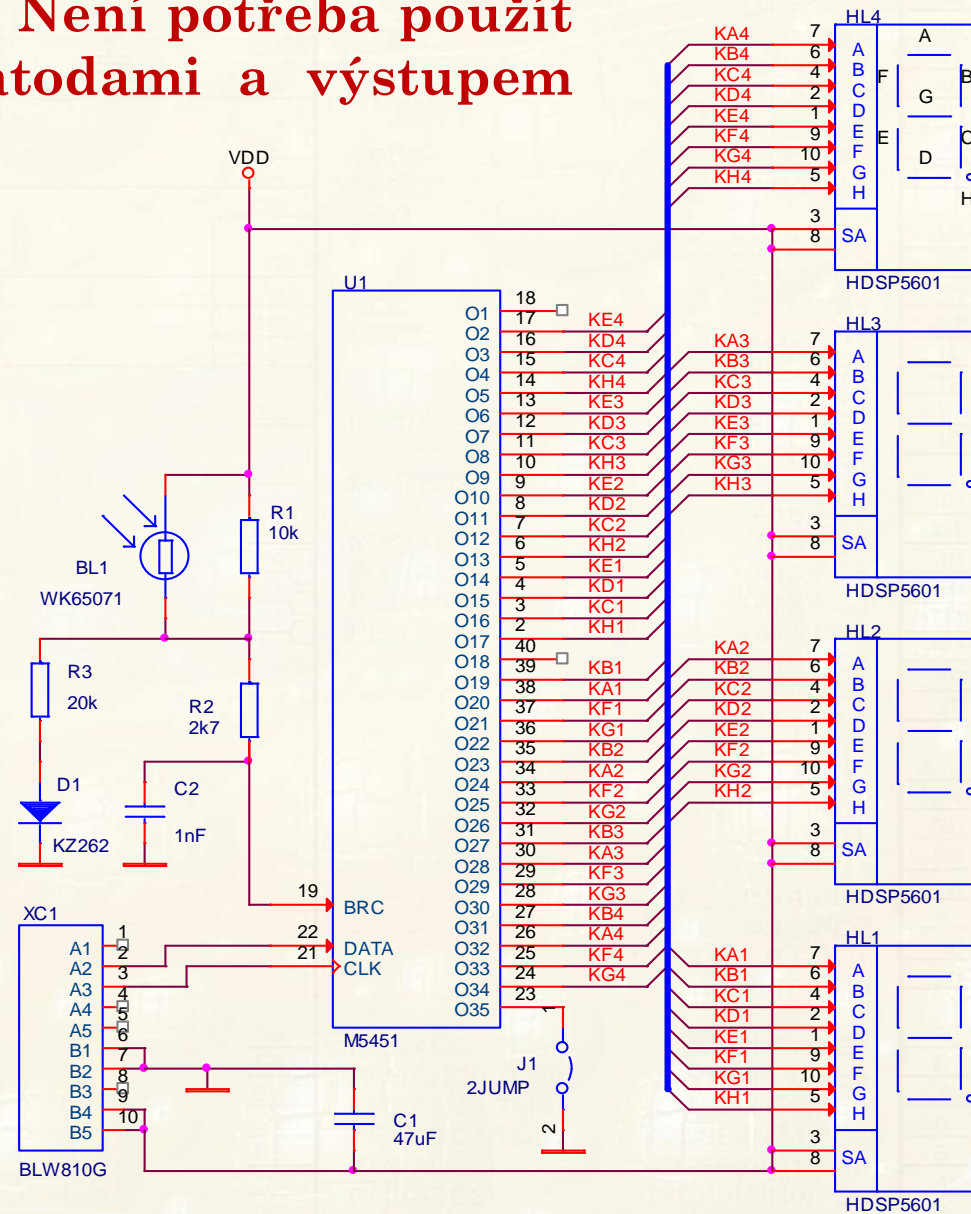
Řadiče zobrazovacích jednotek se sériovým přístupem a statickým režimem.

- ❖ Potřeba dvou nebo tří výstupů procesoru
- ❖ Řadič tvořen posuvným registrem s vyrovnávacím registrem a proudovými budiči segmentů LED zobrazovače.
- ❖ Příkladem řadiče M545x, které dovolují připojit 14, 15, 23, 34 a 35 segmentů zobrazovače s možností analogového řízení jasu.
- ❖ Obdobné řadiče jsou pro i pro sběrnice I²C a SPI.



ZOBRAZOVACÍ JEDNOTKY LED – SÉRIOVÉ STATICKÉ ŘEŠENÍ

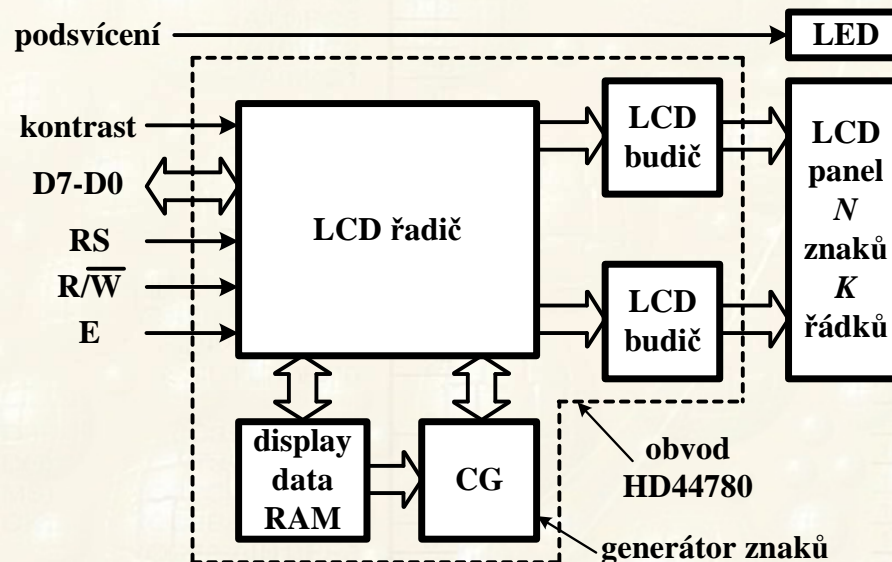
Výstupy řadiče LED jsou realizovány **zdroji proudu** ⇒ **Není potřeba použít odpory mezi katodami a výstupem IO.**



ZOBRAZOVACÍ JEDNOTKY LCD

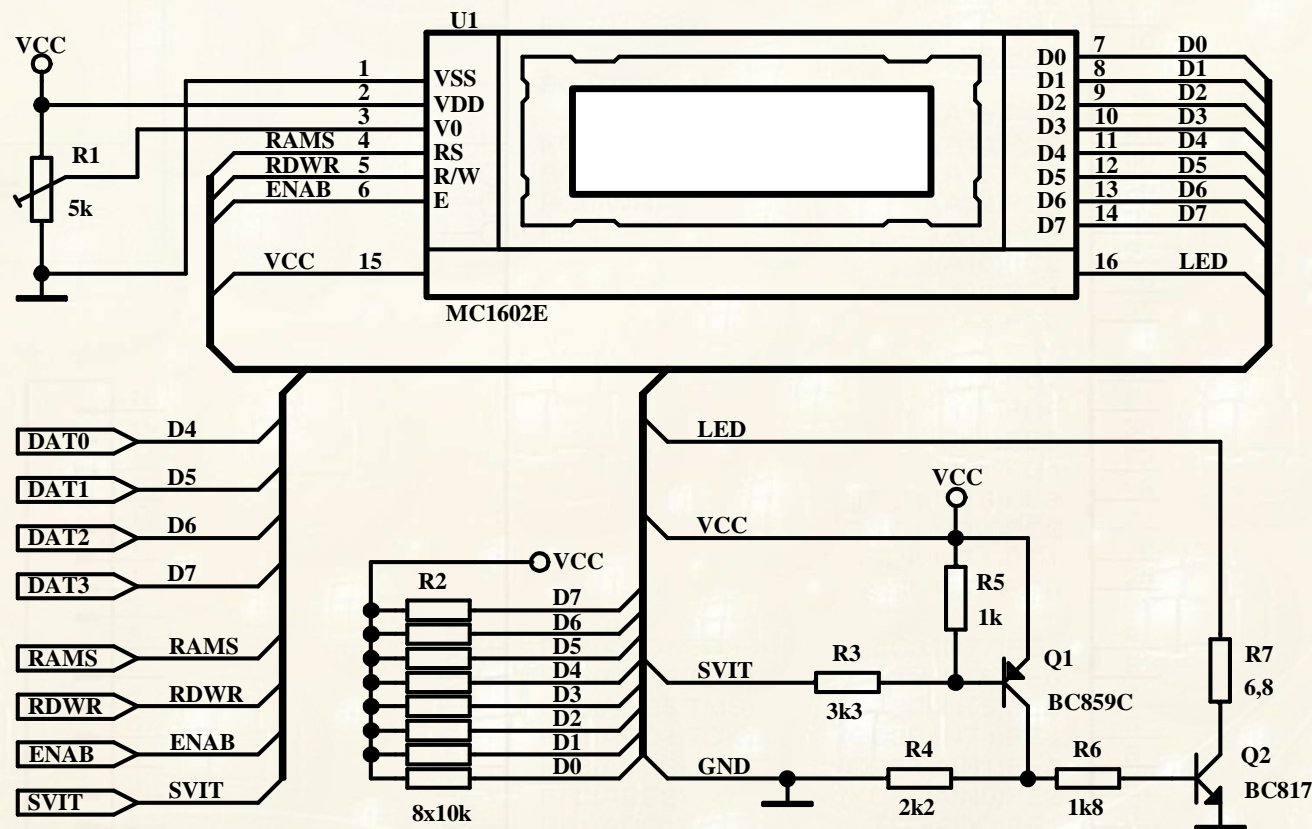
Pro zobrazování většího počtu znaků nebo **omezené spotřebě v aplikaci** můžeme použít **LCD zobrazovač**.

- ❖ LCD jsou obvykle v reflexním provedení s možností řízení kontrastu, ale jejich čitelnost je nižší, než u zobrazovačů LED.
- ❖ Čitelnost se zlepší použitím **displeje s podsvícením** za cenu vyšší spotřeby.
- ❖ Nastupující jsou zobrazovače OLED, životnost nižší než u LCD
- ❖ K připojení **alfanumerického LCD** je potřeba minimálně 7 vývodů.
- ❖ Displej může být připojen k procesorovému systému jako **vnější paměť RAM** nebo **jednotka připojená k V/V vývodům řízených programem**.



ZOBRAZOVACÍ JEDNOTKY LCD

- ❖ Alfanumerické LCD – 1, 2 nebo 4 řádky po 8, 16, 20, 24 nebo 40 znaky obvykle v matici 8x5 bodů.
- ❖ LCD obsahuje RAM 64x5 bitů pro vytvoření 8 uživatelských znaků.
- ❖ Komunikace po 8 (D7÷D0) nebo 4 datových vodičích (D7÷D4, vyšší a pak nižší půl byte) a 3 řídicí vodiče RS, E, R/W. Kontrast i podsvícení LCD zobrazovače se ovládají napětově. Nejčastější problémy jsou s časováním inicializační sekvence.



ZOBRAZOVACÍ JEDNOTKY LCD

		Higher 4-bit (D4 to D7) of Character Code (Hexadecimal)																			
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				
Lower 4-bit (D0 to D3) of Character Code (Hexadecimal)	0	CG RAM (1)			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	1	CG RAM (2)		.	!	@	#	\$	%	&	'	()	*	+	=	>	<	~	^	~
	2	CG RAM (3)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	3	CG RAM (4)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	4	CG RAM (5)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	5	CG RAM (6)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	6	CG RAM (7)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	7	CG RAM (8)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	8	CG RAM (1)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	9	CG RAM (2)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	A	CG RAM (3)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	B	CG RAM (4)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	C	CG RAM (5)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	D	CG RAM (6)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	E	CG RAM (7)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~
	F	CG RAM (8)		"	#	\$	%	&	'	()	*	+	=	>	<	~	^	~	^	~

ZOBRAZOVACÍ JEDNOTKY LCD - PŘÍKAZY

Instrukce	Signál										Popis	Čas
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0		
Smazat displej	0	0	0	0	0	0	0	0	0	1	Nastaví DDRAM=0	1.64 ms
Návrat na pozici 0	0	0	0	0	0	0	0	0	1	x	Kurzor vlevo nahoře	1.64 ms
Vstupní mód	0	0	0	0	0	0	0	1	I/D	S	Směr kurzoru	40 us
Kontrola on/off	0	0	0	0	0	0	1	D	C	B	Displeje, kurzoru, blikání	40 us
Posun kurzoru	0	0	0	0	0	1	S/C	R/L	x	x	Posun kurzoru a displeje bez změny DDRAM.	40 us
Funkční nastavení	0	0	0	0	1	DL	1	0	x	x	Komunikace 4 nebo 8 bitů	40 us
Adresa CGRAM	0	0	0	1	CGRAM adresa					Přenos dat tomto příkazu	40 us	
Adresa DDRAM	0	0	1	DDRAM adresa					Přenos dat tomto příkazu	40 us		
Čtení Busy a adresy	0	1	BF	CGRAM / DDRAM adresa						0 us		
Zápis do CG/DDRAM	0	1	Zápis dat						40 us			
Čtení z CG/DDRAM	1	1	Čtení dat						40 us			

I/D – 1- inkrementace, 0- dekrementace

D – 1- displej ON, 0- displej OFF

B – 1- bliká, 0- neblinká

R/L – 1- doprava, 0- doleva

BF – 1- zaneprázdněn, 0- ready

S – 1- posun, 0- zamrznutí

C – 1- kurzor ON, 0- kurzor OFF

S/C – 1- Posun displeje, 0- kurzoru

DL – 1- 8 bitů, 0-4 bity

S – 1- displej pos., 0- není posun

ZOBRAZOVACÍ JEDNOTKY LCD – ADRESY ZNAKŮ

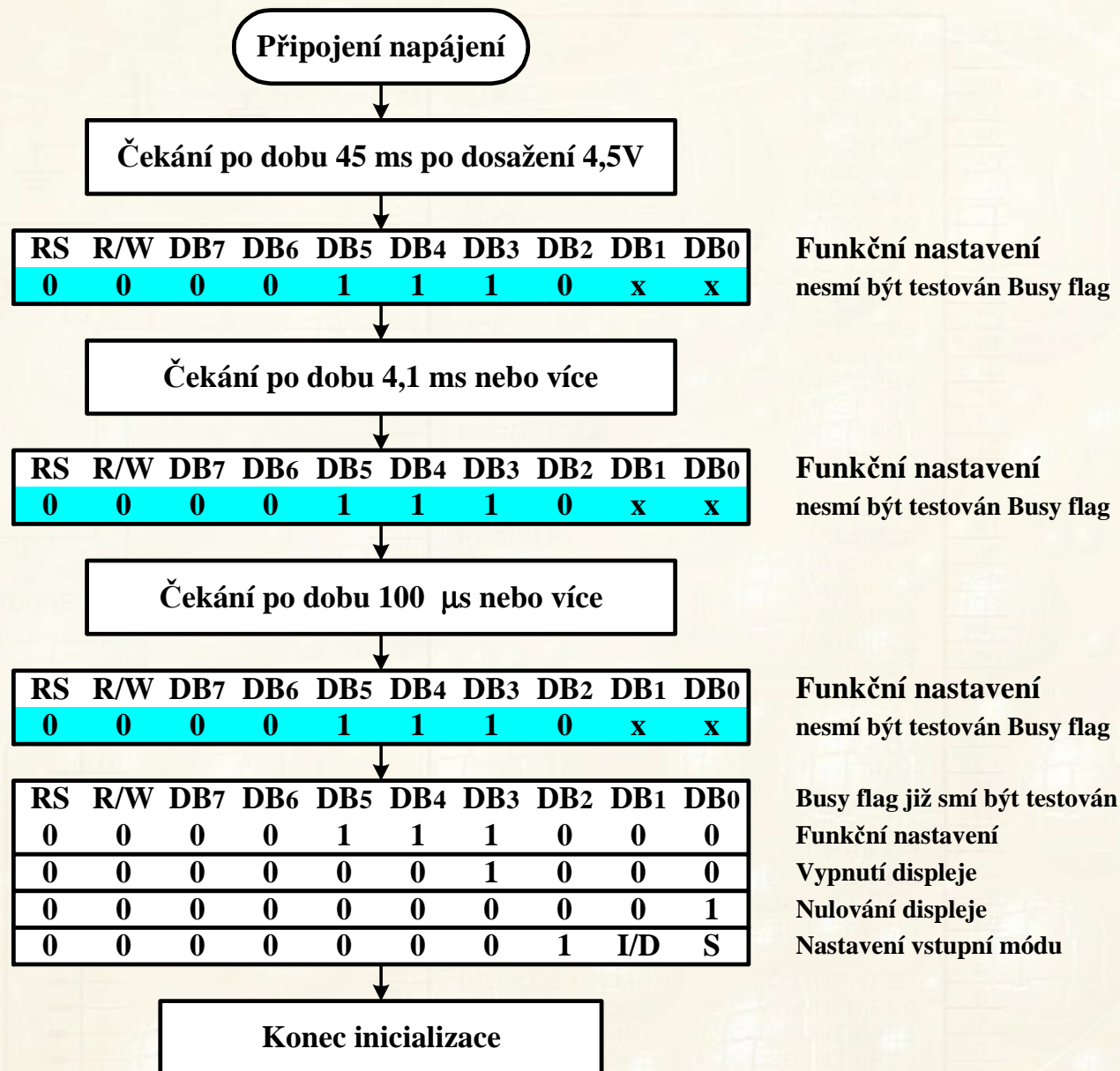
Adresa DDRAM - DDRAM bez posuvu																
Pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Řádek 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Řádek 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Adresa DDRAM - DDRAM posuvu doleva																
Pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Řádek 1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
Řádek 2	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50

Adresa DDRAM - DDRAM posuvu doprava																
Pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Řádek 1	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
Řádek 2	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

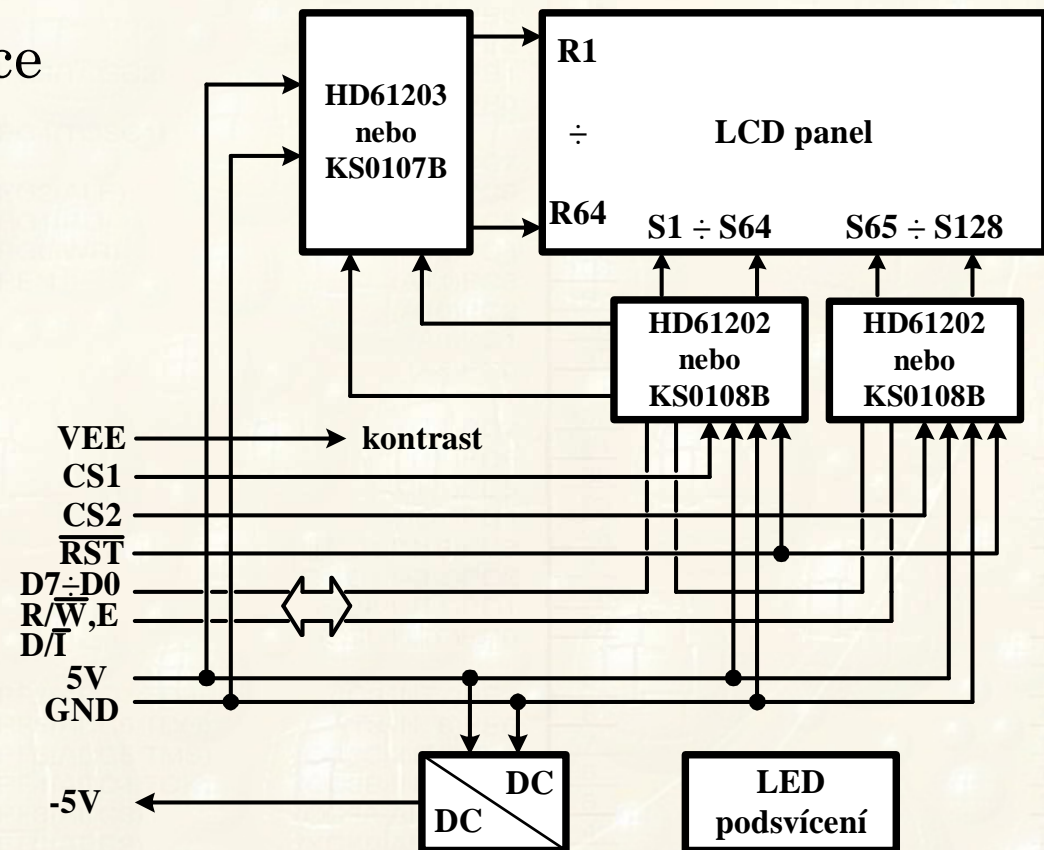
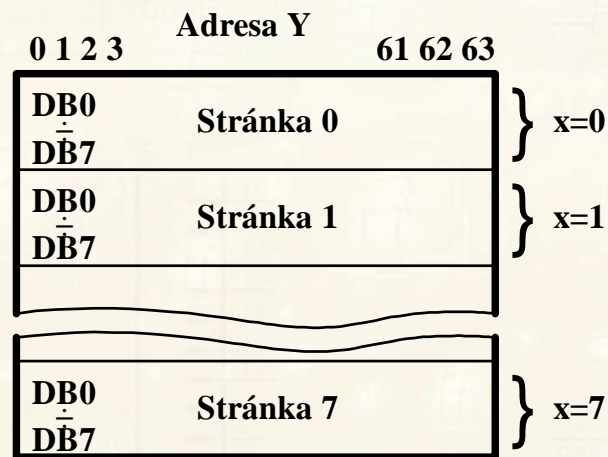
Adresa DDRAM – DDRAM bez posuvu																
Pozice LCD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Řádek 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Řádek 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
Řádek 3	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Řádek 4	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

ZOBRAZOVACÍ JEDNOTKY LCD – INICIALIZACE



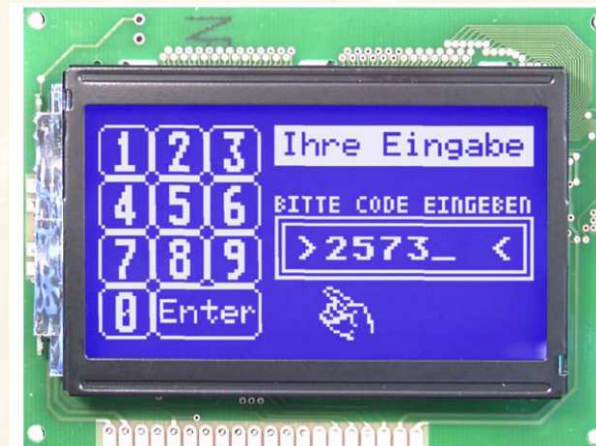
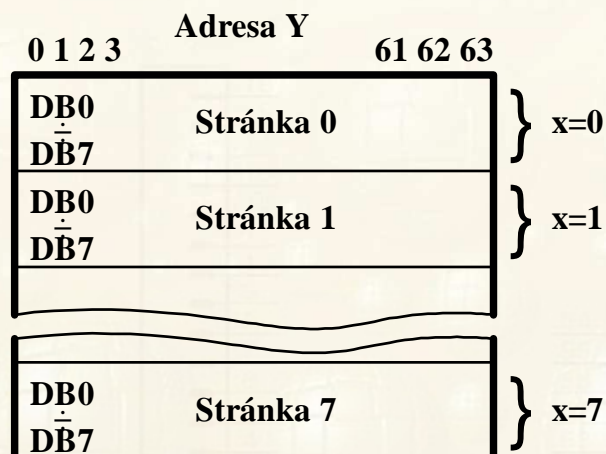
ZOBRAZOVACÍ JEDNOTKY LCD – GRAFICKÉ

- ❖ Grafické moduly LCD se vyrábí s řadičem i bez něj, ale pro spojení s jednočipovými μP je vhodné si vybrat modul s řadičem.
- ❖ Standardní jsou provedení 128x64, 128x128, 240x64, 240x240 bodů.
- ❖ Proti alfanumerickým modulům je rozhraní rozšířeno o dva signály CS2 a nulovací signál RST. Datová sběrnice je pouze 8-bitová.

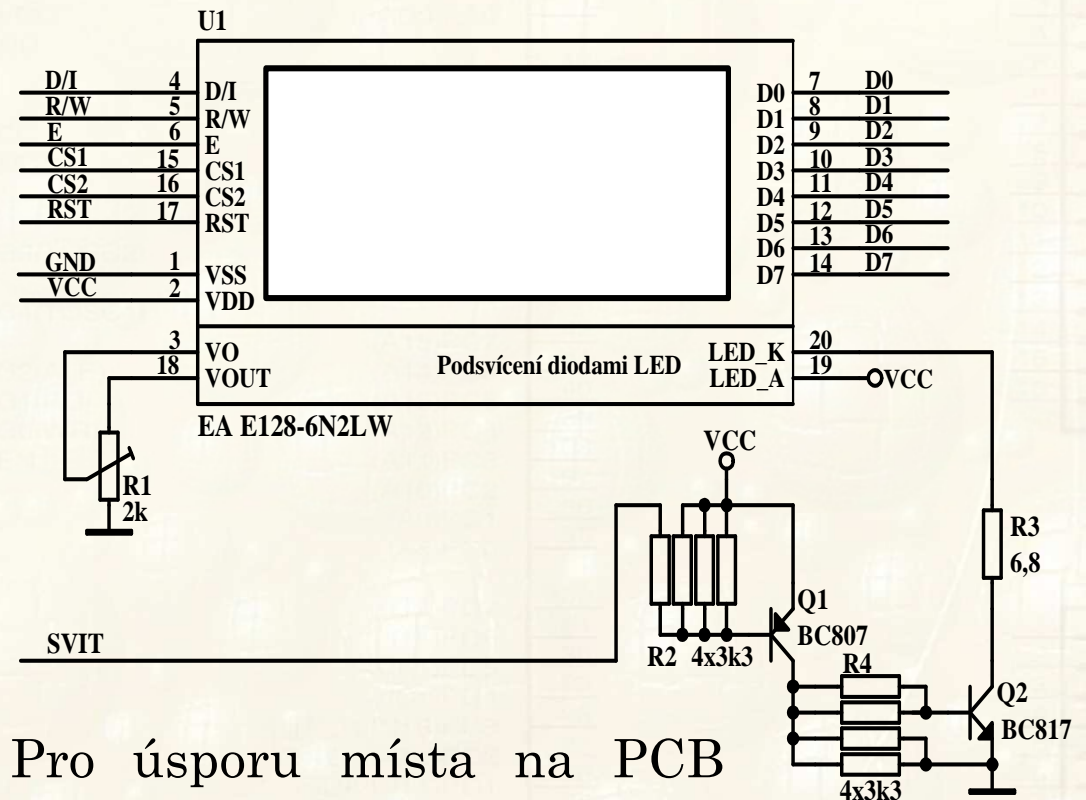
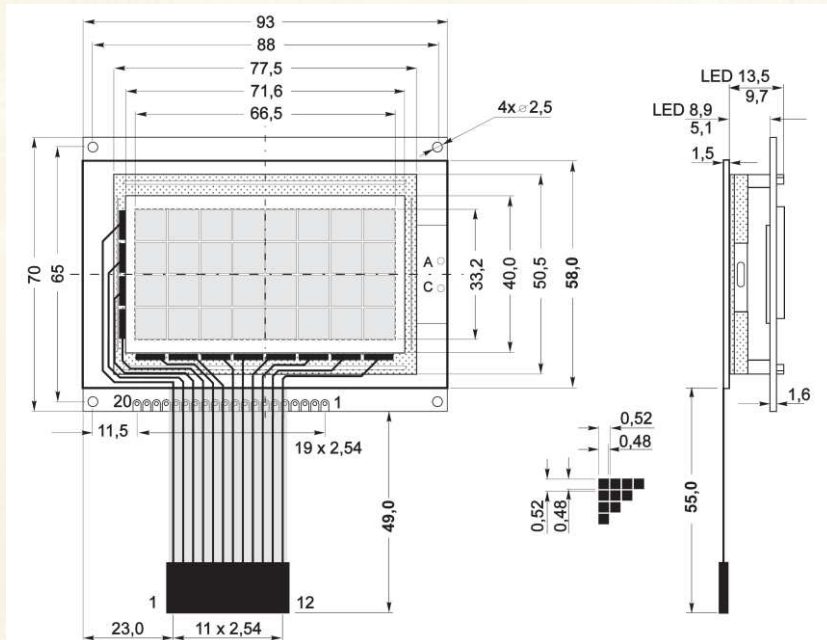
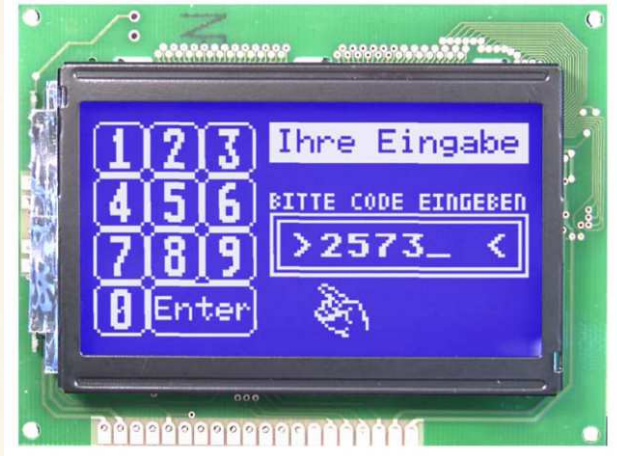


ZOBRAZOVACÍ JEDNOTKY LCD – GRAFICKÉ

- ❖ **Regulace kontrastu VEE obvykle v intervalu $-2V \div 0V$.**
- ❖ Zápis bytu se zobrazuje jako sloupec o 8 bodech v dané stránce X a sloupci Y.
- ❖ Zobrazované znaky mohou být složeny nebo vygenerovány integrovaným generátorem znaků.
- ❖ Některé moduly LCD je možné doplnit **tlačítkovou membránou** s určitým rozdělením zobrazované plochy na **m řádků a n sloupců**. V místech políčka můžeme vytvořit obraz tlačítka s potřebným označením.
- ❖ Novou funkci pak realizujeme programovými prostředky.



ZOBRAZOVACÍ JEDNOTKY LCD – GRAFICKÉ

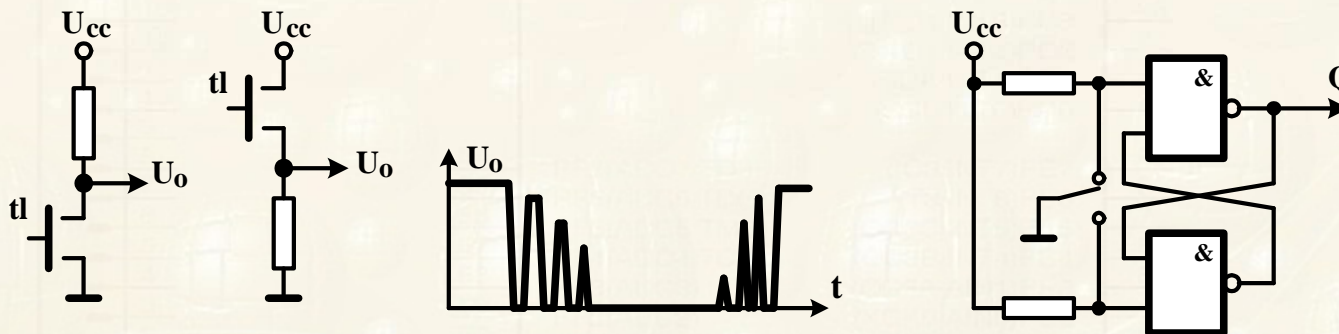


Pro úsporu místa na PCB použity sdružené odpory 4x603. Nebo použít digitální tranzistory, kde odpory jsou integrovány v tranzistoru..

Tlačítka
Klávesnice
Dotyková řešení

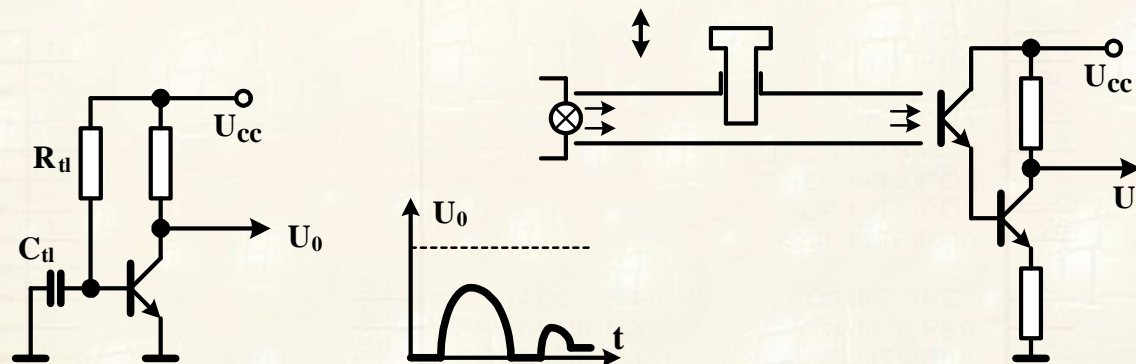
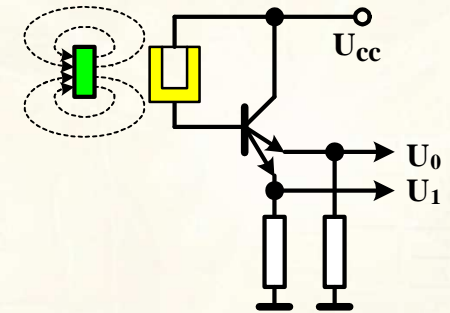
TLAČÍTKA – PASIVNÍ

- ❖ **Tlačítko** - nejjednodušší spojení člověka s mikropočítačem. Tlačítka můžeme rozdělit na
 - **Pasivní** - mechanický kontakt, membránové tlačítko
 - **Aktivní** - Halova sonda, odporové, kapacitní nebo optické, které obvykle neposkytují "**čistý, logický signál**".
- ❖ Výstup tlačítka je nutno obvodově nebo programově ošetřit – pomalý děj s možným vznikem zákmitů.
- ❖ **Mechanický kontakt** – vstup μP se připojuje do země nebo k napájení. Ani tzv. mžikový spínač nezabrání zákmitům (nedokonalý kontakt, mechanické kmity). Odstranění RC-dolní propustí + LKO s hysterezí, paměťovou funkcí (přepínací kontakt + PČ RS) nebo programovým zpracováním.



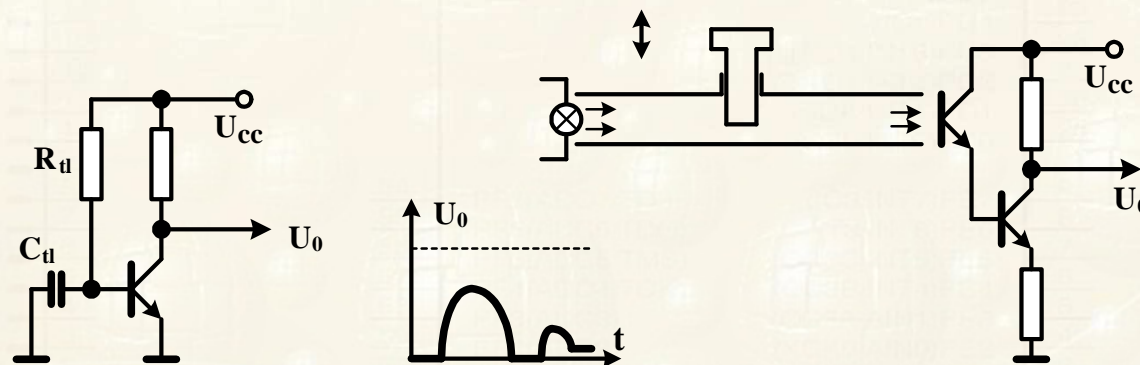
TLAČÍTKA – PASIVNÍ

- ❖ Tlačítko s **Halovým generátorem** se v μP technice používalo dříve (klávesnice), dnes aplikace pro bezkontaktní snímání otáček apod.
- ❖ **Membránové tlačítko** - změna odporu. Zapojení tlačítka závisí na změnách odporu a musí být upraveno tak, aby bylo dosaženo obou logických úrovní. Obsluha může být principiálně stejná jako tlačítko s mechanickým kontaktem.



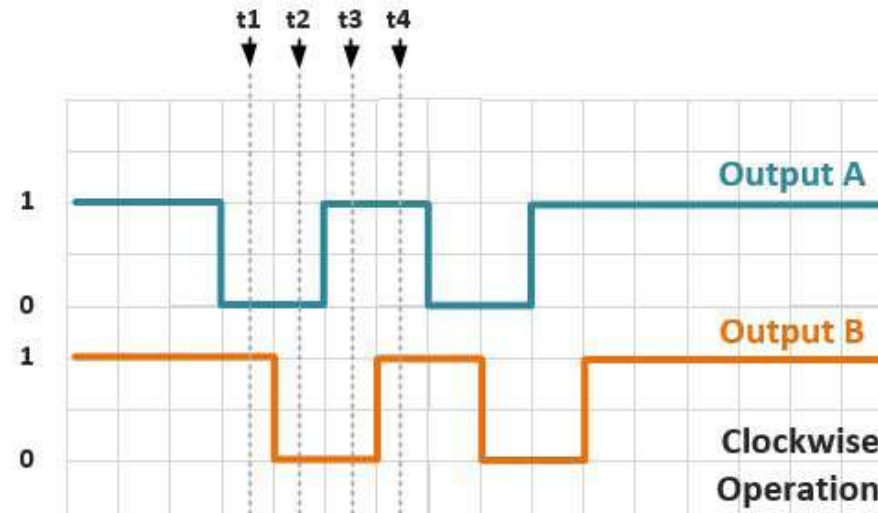
TLAČÍTKA – AKTIVNÍ

- ❖ **Kapacitní tlačítko** je dnes hojně uplatňované a podporované IO i vstupy μP . Identifikuje změnu kapacity izolované plošky připojené k obvodu přiblížením prstu. Hlavní výhodou je odstranění mechanických kontaktů vysoká odolnost proti nešetrnému zacházení.
- ❖ **Membránové tlačítko** - založené na změně kapacity může mít problematickou úroveň výstupního signálu v závislosti na rychlosti stisku klávesy.
- ❖ Stejný problém může mít tlačítko založené na **optickém principu** (nedochází k úplnému přerušování optického svazku). Nutná úprava výstupů na logické úrovni a rychlost přeběhu (**Smithovým klopným obvodem**).



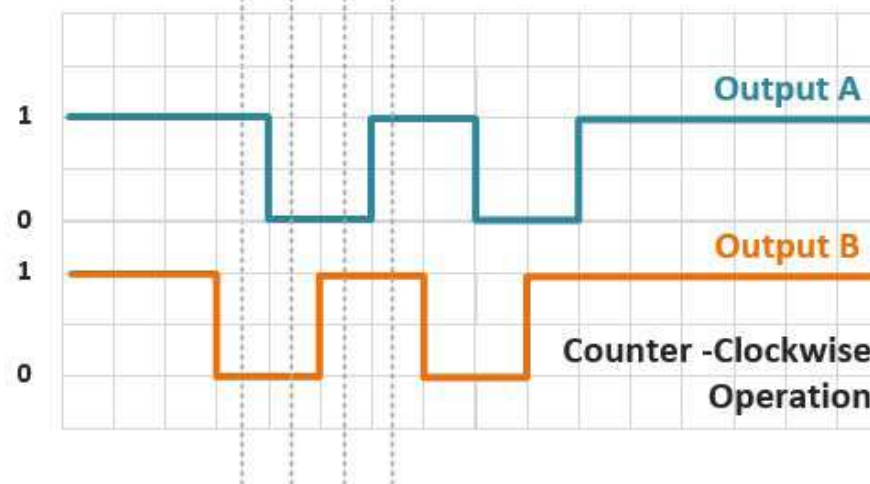
TLAČÍTKA – AKTIVNÍ/PASIVNÍ

- ❖ **Rotační enkodér** je hojně využívaný v aplikacích ke zvětšování /zmenšování nastavované veličiny závislé na směru otáčení osy.



Clockwise Sequence

	A	B
t1	0	1
t2	0	0
t3	1	0
t4	1	1



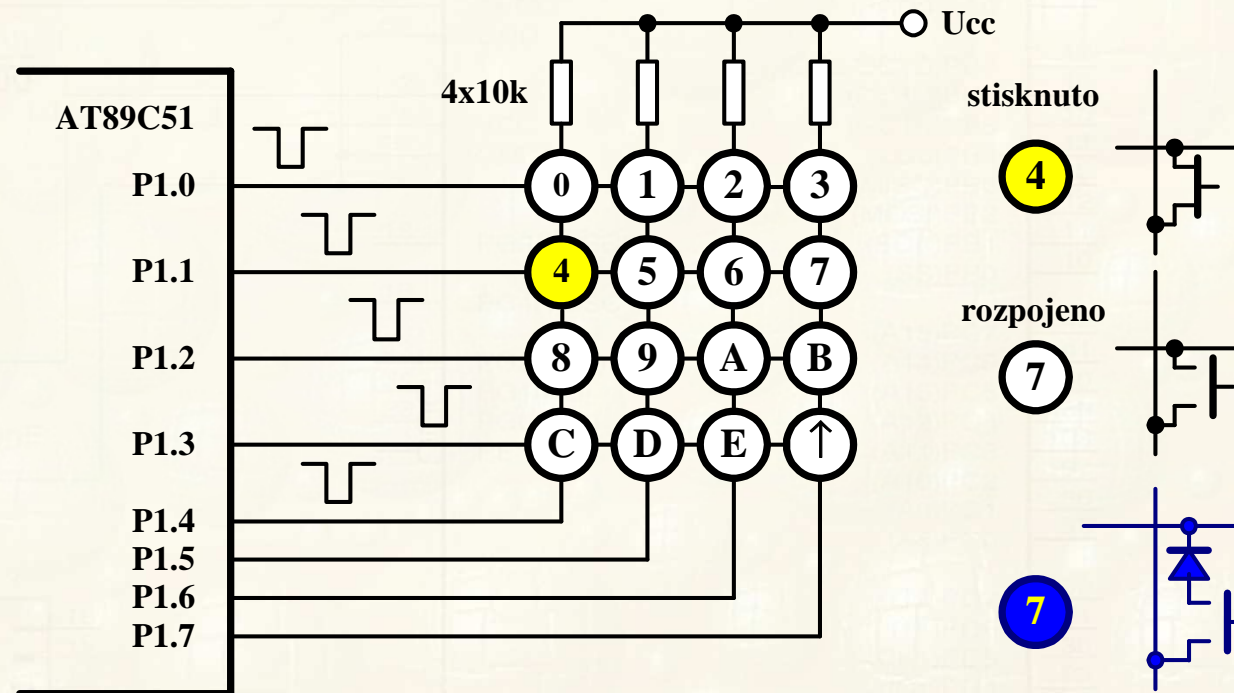
Counter-Clockwise Sequence

	A	B
t1	1	0
t2	0	0
t3	0	1
t4	1	1

Větší počet tlačítek slučujeme do klávesnice **s maticovým** uspořádáním, které minimalizuje počet vývodů potřebných k jejich obsluze a identifikaci zmáčknuté klávesy.

- ❖ Obsluha klávesnice závisí na počtu vývodů a způsobu ovládní
 - Opakované čtení - obsluha součástí main()
 - Obsluha v přerušení
 - Způsob ošetření zákmitů
 - Počtu potenciálně zmáčknutých kláves.
- ❖ Klávesnice s 16 tlačítky – potřeba čtyř odporů, čtyři výstupy OK (PushPull - dynamicky 1x výstup/3x vstup) a čtyři vstupy procesoru.
- ❖ Pro stisk jedné nebo dvou kláves bude zapojení vyhovující.
- ❖ Stisk tří kláves - **shotkyho diody** do série s kontaktem.

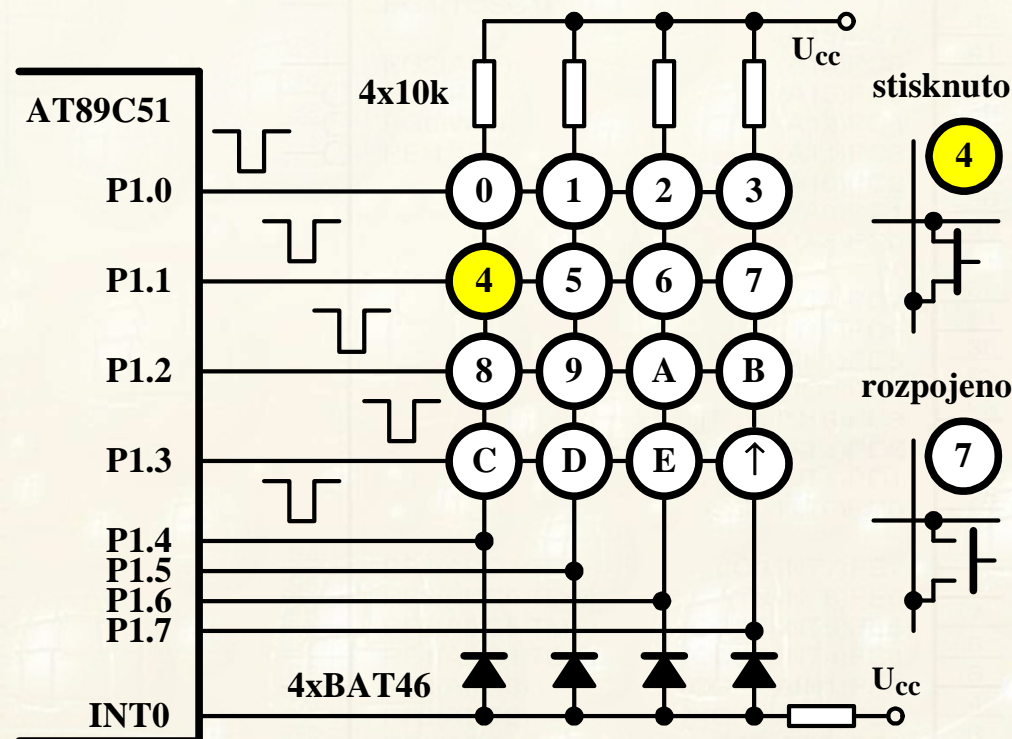
KLÁVESNICE



KLÁVESNICE – S PŘERUŠENÍM

Obsluha klávesnice je stejná jako v předcházejícím případě.

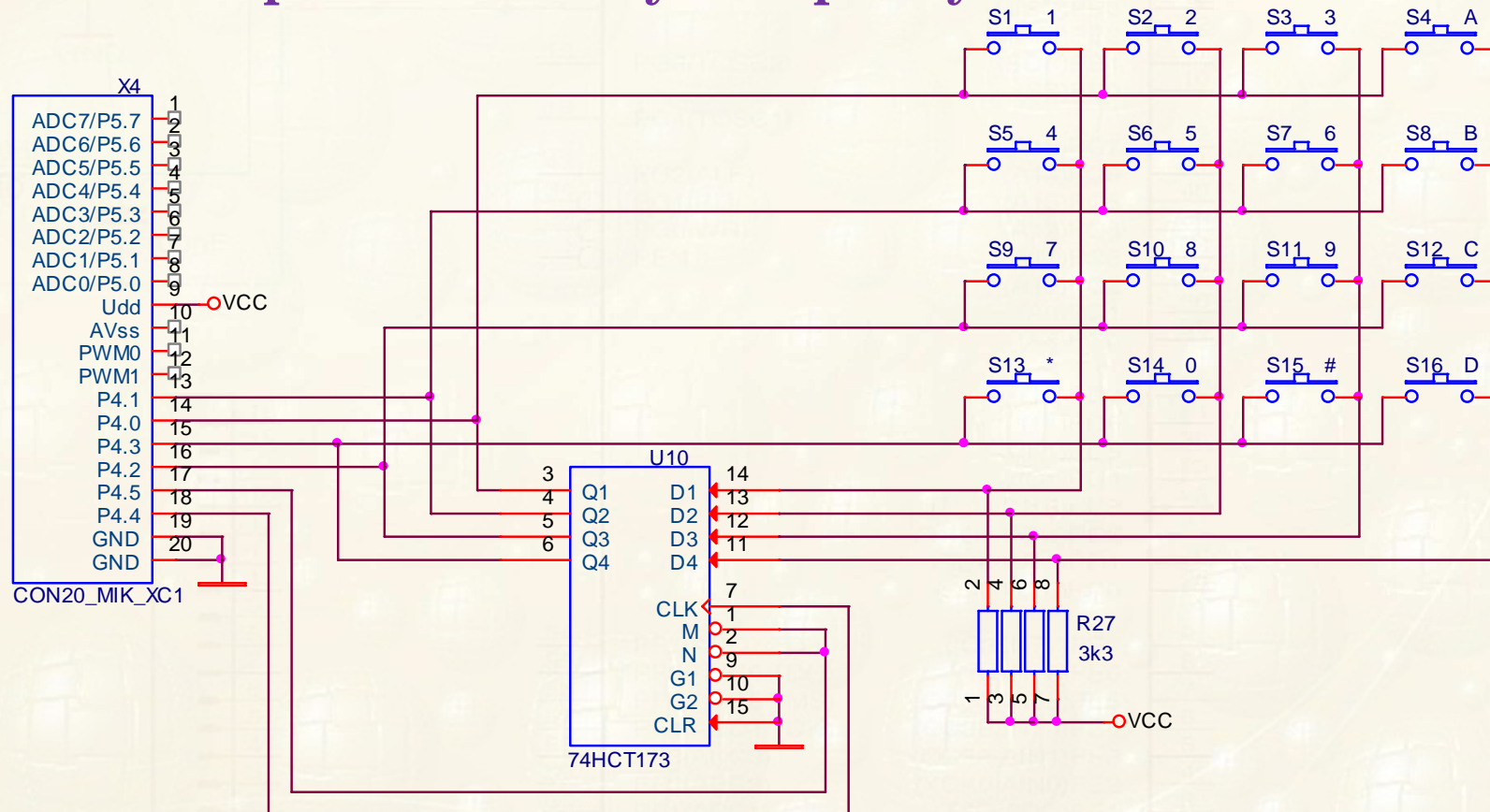
- ❖ Stav sloupců zjišťován v obsluze přerušení od vstupu INT.
- ❖ **Ošetřená** tlačítka lze vyhodnotit přímo v přerušovací rutině.
- ❖ Programové ošetření a opakování dané klávesy – komplikace
- ❖ Okamžitá **reakce na zmáčknutí klávesy**. Žádost o přerušování - DTL logika z Schotkyho diod (montážní součin - AND).



VYUŽITÍ TŘÍSTAVOVÉHO OBVODU K DOČASNÉMU UCHOVÁNÍ INFORMACE

Maticová klávesnice 4x4 + registr s třístavovým výstupem.

- Vývody P4.3÷P4.0 využívány ve vstupním i výstupním režimu.
- CLK U10 uložení sloupcové informace do registru.
- Nastavení brány P4 do vstupního režimu, aktivace třístavového výstupu registru, čtení stavu sloupců.
- **Nevhodné pro více stisky sloupcových tlačítek.**

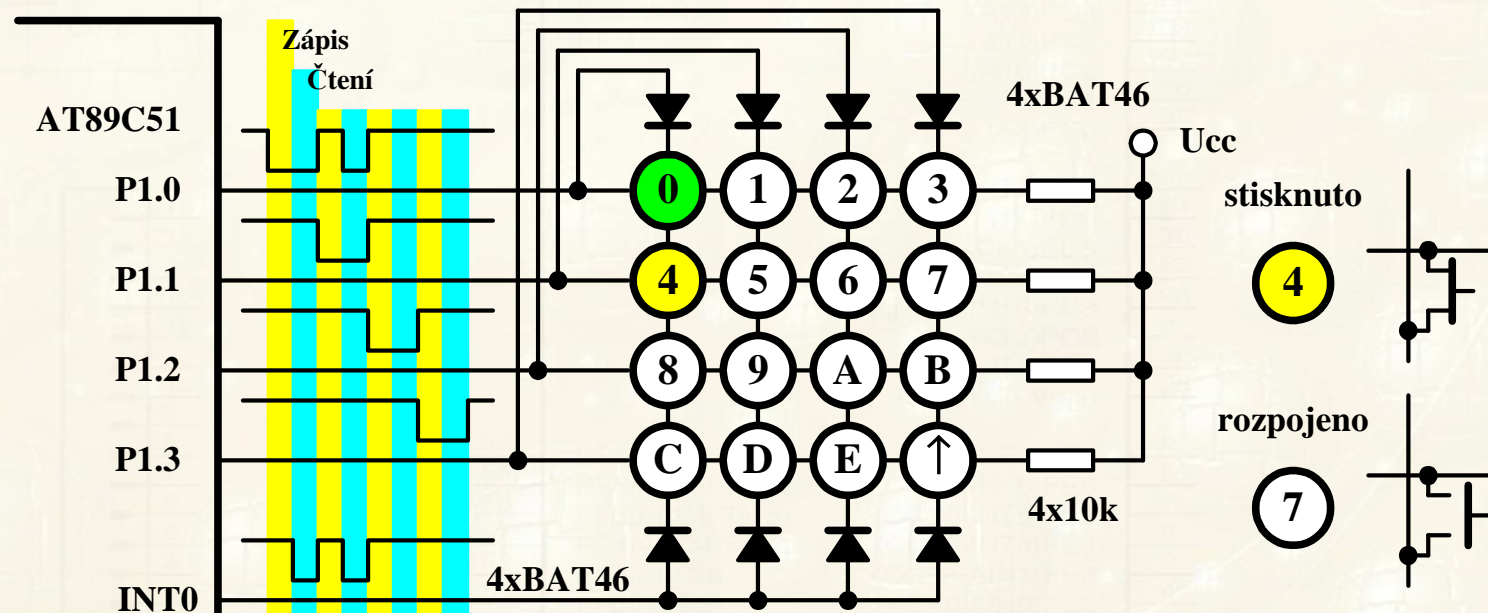


KLÁVESNICE – ŘEŠENÍ S MINIMÁLNÍM POČTEM V/V

Klávesnice s μP s **pseudoobousměrnými V/V vývody.**

- ❖ Na řádky matice je postupně připojována hodnota log.0.
- ❖ V přerušení (montážní součin) přečteme řádky matice.
- ❖ Přečteme-li stejnou hodnotu – tlačítko na diagonále. Řádek a sloupec n. Stav poznáme na vývodu INT0.
- ❖ Řádek v log.1 čtený se stavem log.0 – určuje stisknutou klávesu.
- ❖ Uvedené zapojení je **vhodné pro stisk jedné klávesy.**

Problém: dvě klávesy ve sloupci, dvě diagonální klávesy.



Při obsluze přístroje s klávesnicí je třeba zvážit výhodnost

- ❖ Ošetření zákmitů programovým řešením
- ❖ S obvodovým řešením
- ❖ S využitím přerušovacího systému

Zda je přístroj určen

- ❖ Pro mládež – rychlá reakce, rychlé opakování – jednotky 0,1s
- ❖ Seniors - pomalá reakce a pomalé opakování – násobky 0,5s
- ❖ Pro širokou populaci
- ❖ Zda funkce zmáčknuté klávesy se má opakovat
- ❖ Zda procesor zpracovává další přerušení, která nesmí být zpožděna nebo zastavena.
- ❖ U měřicích přístrojů musí být zcela funkční (provádí měření veličiny, její vyhodnocení a nastavení výstupních relé, atd.) i v nastavovacím režimu a pokud možno se stejnou periodou.

KLÁVESNICE – „PRŮCHODOVÉ“ ŘEŠENÍ

Periodická obsluha v **hlavní smyčce**. Výhody - nastavení doby stisku, intervalu vyhodnocení, priority dvojstisku, opakování funkce tlačítka s ohledem na uživatele.

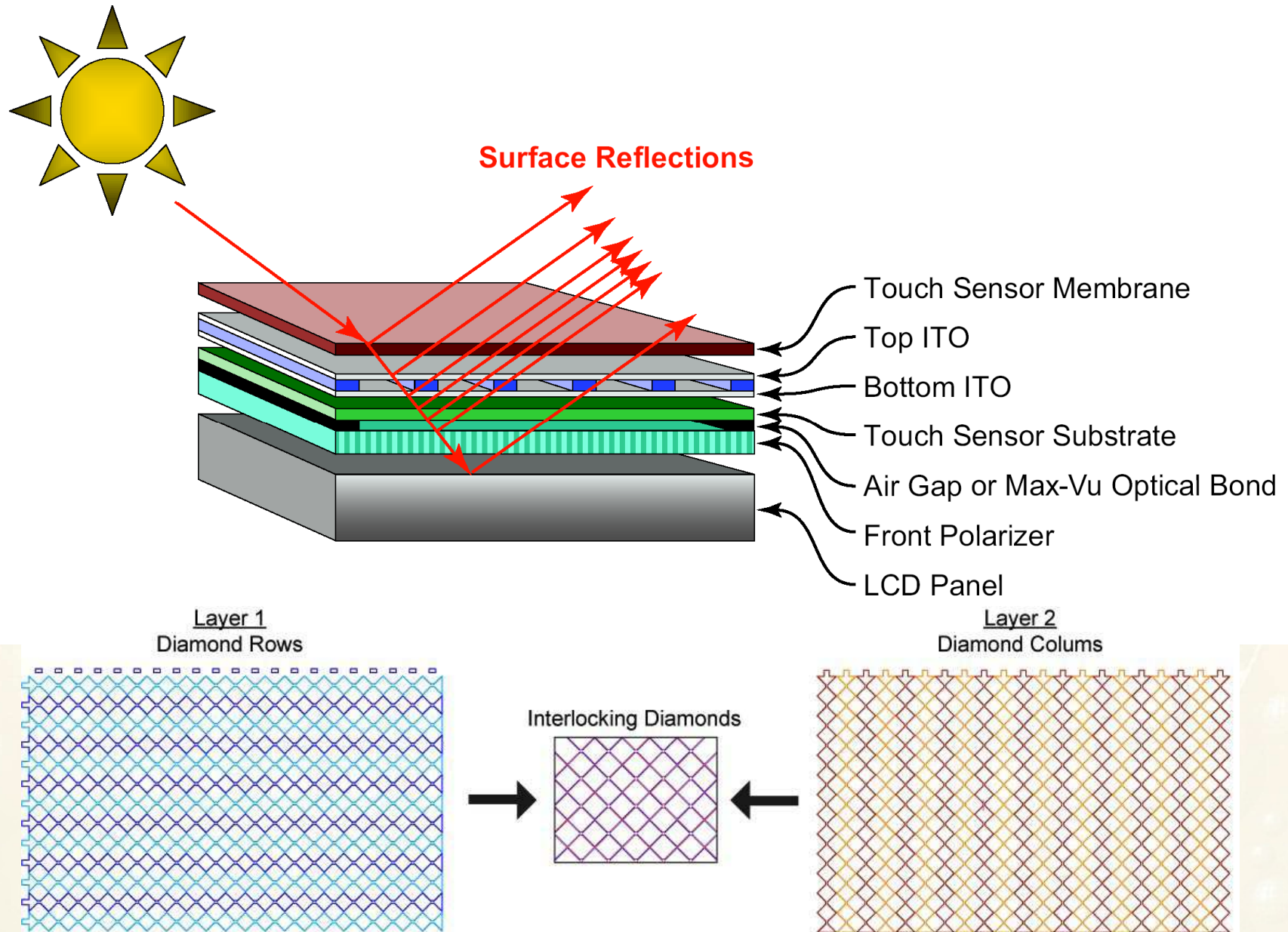
```
unsigned char test_tlac(void)                /* Průchodová verze */
{
    unsigned char y;
    if (set==0) {if (pset<5) pset++;bset=1;}
    else { if ((pset<5)&&(bset==1)) { bset=0; pset=0;}}
    if (sel==0) {if (psel<5) psel++;bsel=1;}
    else { if ((psel<5)&&(bsel==1)) { bsel=0; psel=0;}}
    if (prog==0) {if (pprog<5) pprog++;bprog=1;}
    else { if ((pprog<5)&&(bprog==1)) { bprog=0; pprog=0;}}
    if (enter==0) {if (penter<5) penter++; benter=1;}
    else { if ((penter<5)&&(benter==1)) { benter=0; penter=0;}}

    if (old==bitdes) return(0);              /* počkej na uplynutí
0.5 s */
    old=bitdes;      y=ZADNE;                // test zmačknutých
tlačítek
    if (pset>=5)     y=H_SET;                 /* SET */
    if (psel>=5)     y=H_SEL;                 /* SEL */
    if (pprog>=5)    y=H_PROG;                /* Prog */
    if (penter>=5)   y=H_ENTER;              /* Potvrzení */
    pset=psel=pprog=penter=0;
    bprog=benter=bset=bsel=0; return(y);
}
```

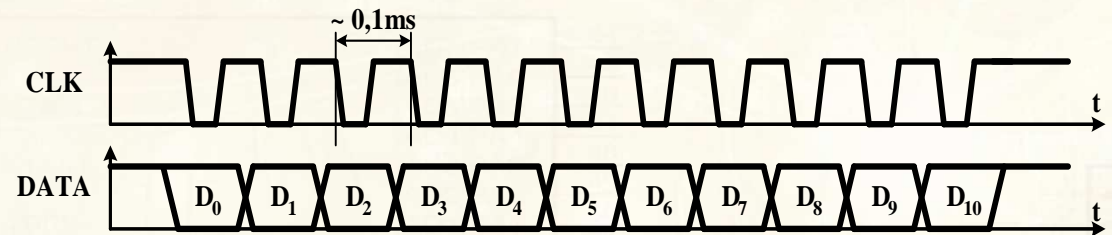
Tlačítkové vstupy mohou být nahrazeny

- ❖ Infračerveným přenosem pomocí kódů dálkových ovladačů (RC5, IrDA, atd.)
- ❖ RF přenosem (FSK, Zigbee, Bluetooth, atd.)
- ❖ Mobilním telefonem (WiFi, Bluetooth).
- ❖ **V multimediálních přístrojích** se nyní uplatňují grafické displeje kombinované s odporovou nebo kapacitní membránovou klávesnicí nebo klávesnicí založenou na principu kapacitního tlačítka s analogovou identifikací souřadnice X a Y dotyku.

KLÁVESNICE – KOMBINACE DISPLEJE A KLÁVESNICE



KLÁVESNICE PC – PŘÍJEM A EMULACE



- ❖ **Emulace klávesnice** - existují zařízení, jejichž výstup se připojuje k počítači PC
- ❖ **Příjem znaků z PC** - periférie generující kódy klávesnice PC, které jsou zpracovávány mikropočítačem. Například čtečky čárkových kódů, některé čtečky bezkontaktních karet, atd.

Klávesnice je připojena 5 vodiči Clk(1), Data(2), Reset(3), Zem(4) a U_{CC} (5). Komunikuje s PC obousměrně 11 bitovým sériovým synchronním přenosem. Data potvrzuje vzestupná hrana CLK, minimální mezera mezi dvěma kódy vyslanými je 1,2 ms. Zmáčknutí klávesy - je vyslán kód zmáčknutí a zmáčknuté klávesy. Uvolnění - je vyslán kód uvolnění (0x7E0, (0xF0)) a kód uvolněné klávesy. **Kódy čísel z numerické klávesnice a čísel z písmenové řady pod F1 až F12 jsou rozdílné !!!**

KLÁVESNICE PC – KÓDY KLÁVES

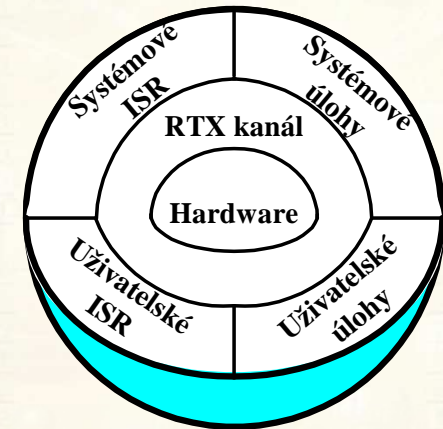
tabulka kódů a jmen kódů kláves klávesnice											
pořadí	kód	jméno	pořadí	kód	jméno	pořadí	kód	jméno	pořadí	kód	jméno
0	\$07E0		32	\$0452		64	\$0000		96	\$041C	`
1	\$060A	F1	33	\$06F6	N-	65	\$0438	A	97	\$07C2	PAUSE
2	\$060C	F2	34	\$0708	YSRQ	66	\$0464	B	98	\$05C0	GR2
3	\$0408	F3	35	\$06EE	NUML	67	\$0642	C	99	\$0000	
4	\$0618	F4	36	\$06FC	SCROL	68	\$0446	D	100	\$0622	RALT
5	\$0606	F5	37	\$04B0	CAPSL	69	\$0648	E	101	\$0628	RCTRL
6	\$0416	F6	38	\$04C2	&	70	\$0656	F	102	\$06E2	DEL
7	\$0506	F7	39	\$04A4	'	71	\$0468	G	103	\$06D2	END
8	\$0614	F8	40	\$03ED	PCLED	72	\$0666	H	104	\$04F4	PGD
9	\$0402	F9	41	\$07F4	PCLED	73	\$0486	I	105	\$06FA	PGU
10	\$0612	F10	42	\$04F8	*	74	\$0476	J	106	\$06D8	HOME
11	\$06F0	F11	43	\$04F2	+	75	\$0684	K	107	\$04E0	INS
12	\$040E	F12	44	\$0682	,	76	\$0696	L	108	\$04D6	LEFT
13	\$06B4	ENT	45	\$069C	.	77	\$0674	M	109	\$06E8	RIGHT
14	\$04EC	ESC	46	\$0492	.	78	\$0462	N	110	\$04EA	UP
15	\$0628	CTRL	47	\$0494	/	79	\$0688	O	111	\$06E4	DOWN
16	\$0622	ALT	48	\$048A	0	80	\$069A	P	112	\$06B4	NENT
17	\$0624	SHIFT	49	\$042C	1	81	\$042A	Q	113	\$0494	N/
18	\$06B2	RSHIFT	50	\$063C	2	82	\$065A	R	114	\$0624	ESH
19	\$041A	TAB	51	\$044C	3	83	\$0636	S	115	\$04F8	PRSCR
20	\$06CC	BKSP	52	\$044A	4	84	\$0458	T	116	\$06FC	BREAK
21	\$04E0	N0	53	\$065C	5	85	\$0678	U	117	\$0000	
22	\$06D2	N1	54	\$066C	6	86	\$0454	V	118	\$0000	
23	\$06E4	N2	55	\$047A	7	87	\$063A	W	119	\$0000	
24	\$04F4	N3	56	\$047C	8	88	\$0644	X	120	\$0000	
25	\$04D6	N4	57	\$048C	9	89	\$066A	Y	121	\$0000	
26	\$04E6	N5	58	\$0000		90	\$0434	Z	122	\$0000	
27	\$06E8	N6	59	\$0498	;	91	\$04A8	[123	\$0000	
28	\$06D8	N7	60	\$0000		92	\$04BA	\	124	\$0000	
29	\$04EA	N8	61	\$06AA	=	93	\$04B6]	125	\$0000	
30	\$06FA	N9	62	\$0000		94	\$0000		126	\$0000	
31	\$06E2	N.	63	\$0000		95	\$0000		127	\$0000	

Multitasking

MULTITASKING – ZPRACOVÁNÍ VÍCE ÚLOH

Základní typy aplikací s mikroprocesorem:

- ❖ **Zpracování "v reálném čase,,** - úloha musí být vykonána v definovaném intervalu např. perioda vzorkování.
- ❖ **Multiprogramování (multitasking)** – zpracování několik nezávislých úloh současně na jednom procesoru.

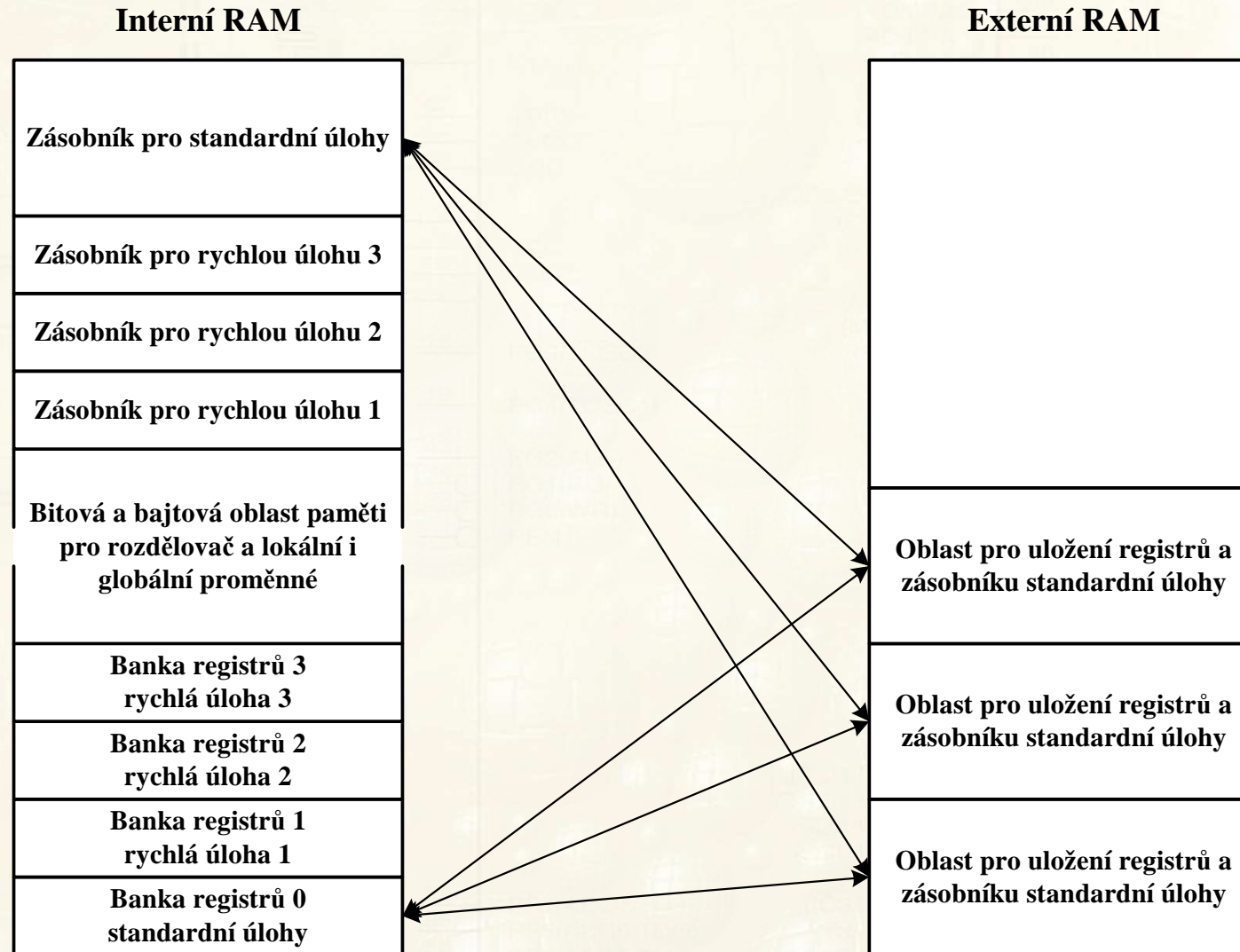


Nejjednodušší řešení - pomocí přerušovacího systému a časovače. Každé úloze přidělíme určitou dobu na zpracování.

Program RTX (Real-Time Multitasking Executive) - řeší problémy s organizací, obsluhou, spouštěním jednotlivých úloh i jejich vzájemnou komunikací.

Jednotlivé úlohy (**task**) - realizovány jako nezávislé programové procesy. RTX - snadná integrace do vývojového prostředí, kompletní podpora volání úloh, obnova stavu registrů, zásobníku úlohy a přerušovacího systému. RTX potřebuje jen několik systémových zdrojů a lze jej využít i v časově kritických aplikacích.

MULTITASKING – ZPRACOVÁNÍ VÍCE ÚLOH UKÁZKA PRO 8051



Komunikace mezi úlohami a jejich synchronizace je zajišťována:

- ❖ **Signály** - nejrychlejší forma synchronizace úloh, žádné zprávy, pouze aktivují úlohu.
- ❖ **Zprávy** - přenášeny přes **poštovní schránky** – princip FIFO (First-In, First-Out). Úlohy čekají ve frontě na přijetí zprávy. Nejdéle čekající úloha obdrží zprávu jako první.
- ❖ **Semaforey** - **protokolové mechanismy** – umožňují sdílení společných prostředků bez přístupových konfliktů. Společné prostředky jsou řízeny tak, aby v daném čase je používala pouze jedna úloha. Přistupuje-li k jednotce více úloh, potom přístup je povolen jedné a další jsou zařazeny do pořadníku a čekají tak dlouho, dokud povolená úloha nedokončí svoji činnost.

Úlohy mohou čekat na rozmanité události a nemusí při tom spotřebovávat čas procesoru. Událostí může být přijetí zprávy, signálu, přerušení, přetečení času nebo jejich kombinace.

Multiprocesorové systémy

Multiprocessorový systém se 2 nebo více procesory můžeme obecně vytvořit propojením přes

- ♣ sériové rozhraní
- ♣ paralelní rozhraní
- ♣ společnou sběrnici
- ♣ pomocí FIFO pamětí
- ♣ globální (dvoubránovou) paměť

Volba propojení závisí na

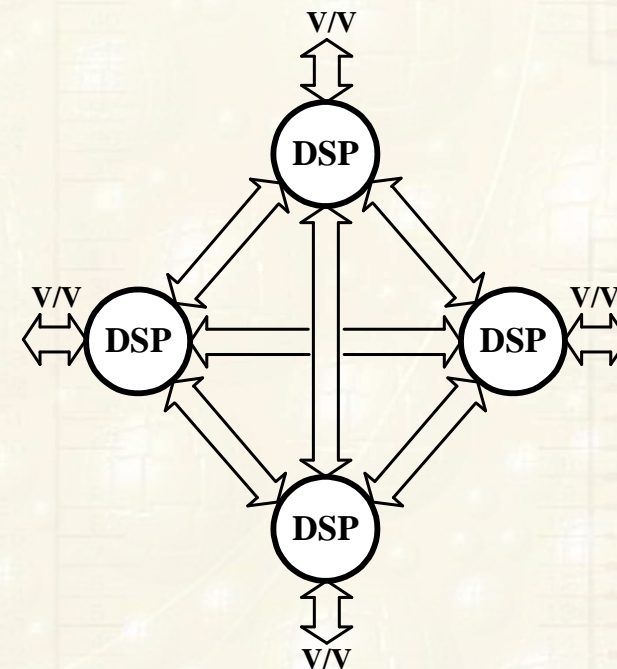
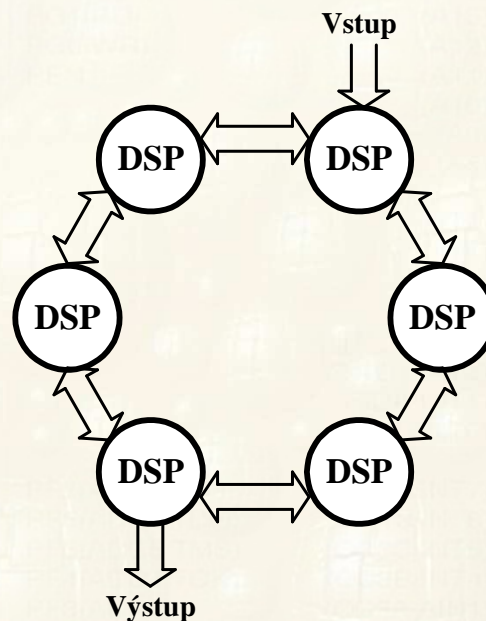
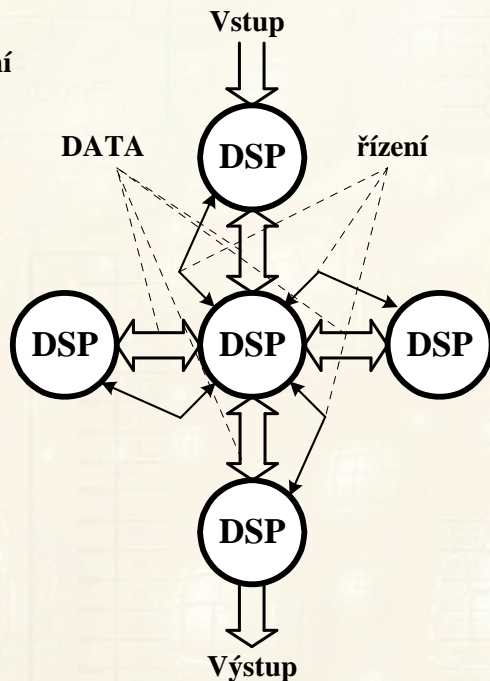
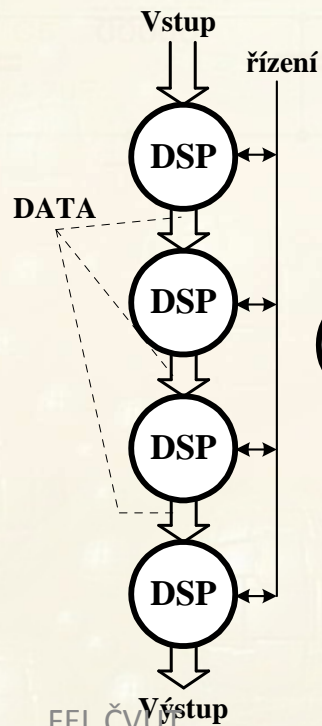
- ♣ potřebných datových tocích
- ♣ zvolené struktuře
- ♣ možnostech použitých procesorech

Dnes jsou v zařízeních s **extrémním výpočetním výkonem** multiprocessorové systémy nahrazovány procesory **VLIW** nebo HW řešením v **DSP programovatelných polích** následované **VLIW**.
Důvod ústupu - bezpečnější tvorba programu proti rozdělení úlohy do programů pro jednotlivé procesory a synchronizaci datových toků mezi nimi.

ARCHITEKTURY MULTIPROCESOROVÝCH SOUSTAV

Mezi nejznámější architektury

- ❖ **Průtoková,**
- ❖ **Hvězda**
- ❖ **Ring.**
- ❖ **Univerzální multiprocessorové desky** - umožňují všechny uvedené typy. Přenosy dat realizovány - sběrnicí, dvoubránovou nebo FIFO pamětí.

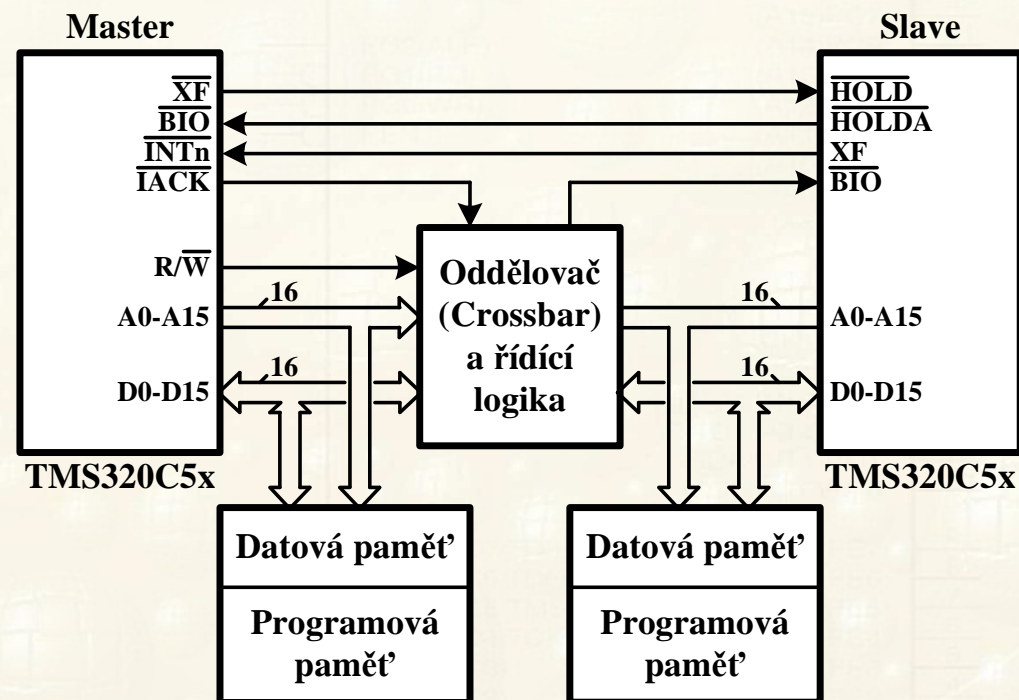


MULTIPROCESOROVÁ KOMUNIKACE MASTER-SLAVE

Master-Slave konfigurace - Master předává po sběrnici do vnějších/vnitřních pamětí signálových procesorů nová **data** nebo **programy** (úlohy) ke zpracování.

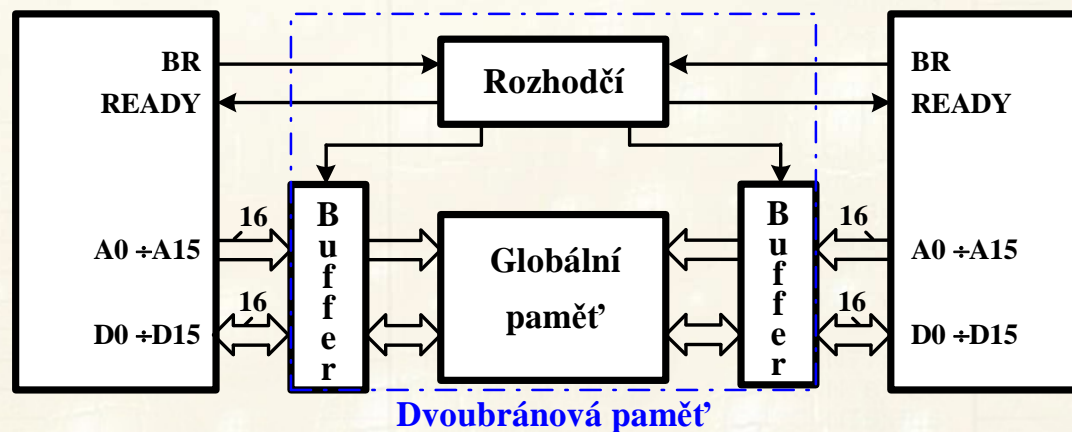
Master požádá Slave o stav vysoké impedance na lokální sběrnici – propojení sběrnic obvody Crossbar – přenos dat nebo programu – uvolnění řízení lokální sběrnice procesoru Slave.

Podřízený procesor informuje o zpracování Master např. přerušením.



MULTIPROCESOROVÁ KOMUNIKACE S GLOBÁLNÍ PAMĚTÍ

- ❖ Procesory si přes **globální paměť** předávají data.
- ❖ Ani jeden procesor nemá preferovaném postavení.
- ❖ μ P sdílející paměť realizují **čtení** nebo **zápis** do této paměti.
- ❖ Struktura musí obsahovat **rozhodčího**, který povolí přístup jen jednomu μ P a propojí jeho sběrnice s adresovými a datovými vodiči paměti.
- ❖ Zbývajícím μ P jsou pro přístup vnuceny **čekací stavy**.
- ❖ Priorita rozhodčího - pevně nastavená, proměnná, rotující.
- ❖ Globální paměť, rozhodčí a oddělovače = **dvoubránová paměť**.



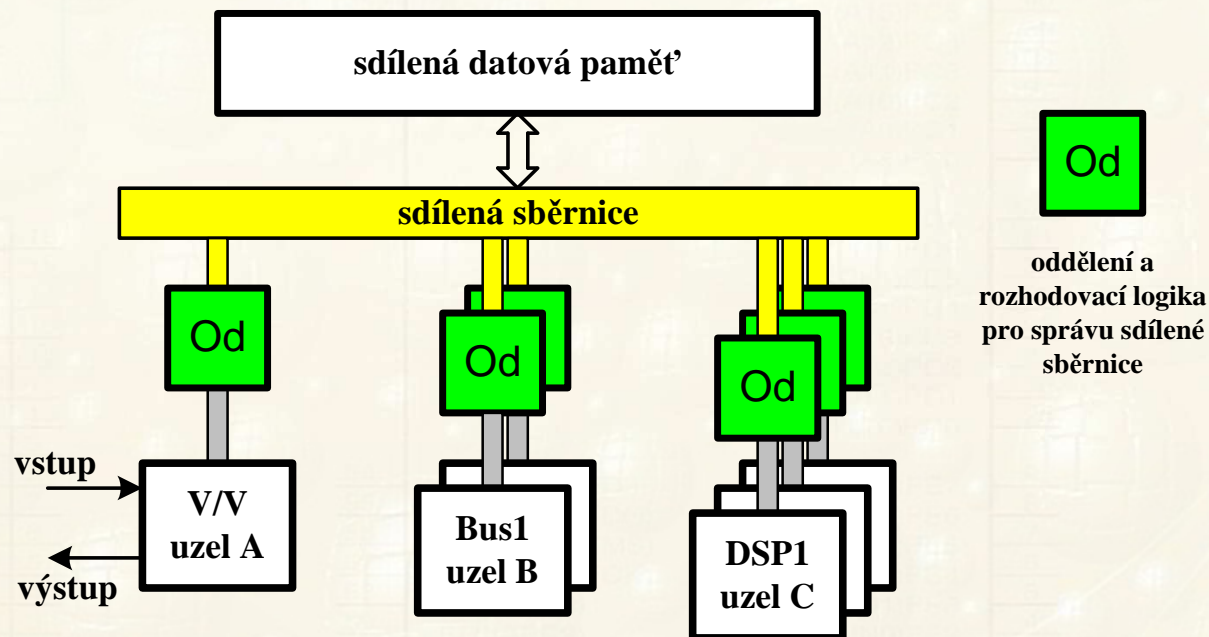
*) Signálem READY nebo WAIT disponují jen některé procesory, obvykle to jsou signálové a univerzální. U některých procesorů je pouze generátor čekacích stavů nebo programovatelná doba přístupu do paměti.

MULTIPROCESOROVÁ KOMUNIKACE SE SDÍLENOU PAMĚTÍ

Alternativou k struktuře s globální pamětí je multiprocessorový systém se **sdílenou sběrnicí** a pamětí. Konfigurace v podstatě představuje propojení lokálních sběrnic jednotlivých procesorů se sdílenou sběrnicí, ke které je připojena sdílená paměť.

Sdílenou sběrnicí může být **interní sběrnice** multiprocessorového systému nebo **sběrnice hostitelského rozhraní** (PCI).

Problémy - s programováním jednotlivých úloh, s rychlostí přístupu a datovými toky po sdílené sběrnici. Přístupové doby funkčních uzlů (např. DSP) mohou být relativně dlouhé (počet uzlů a ovládací logice).

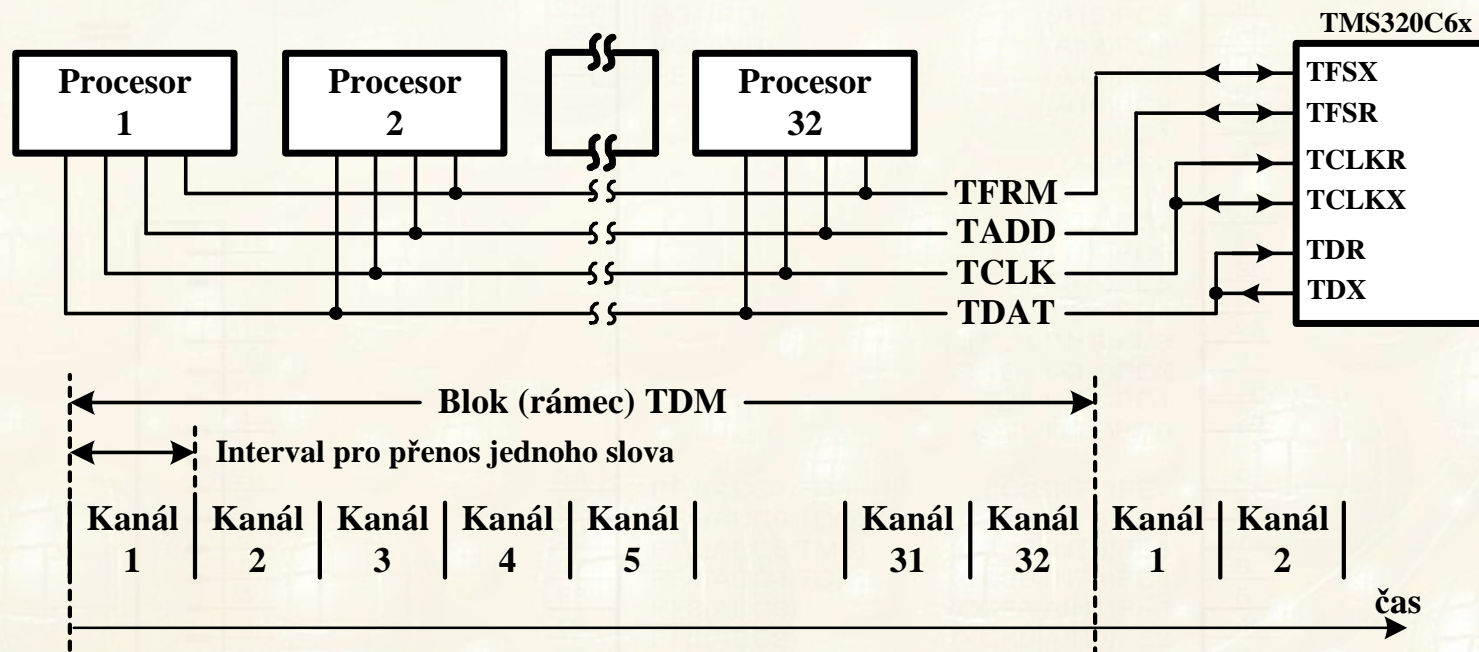


SYNCHRONNÍ SÉRIOVÁ MULTIPROCESOROVÁ KOMUNIKACE TDM

Pro datové přenosy (1÷25Mb/s) můžeme vytvořit strukturu se **sériovou časově dělenou sběrnicí TDM**.

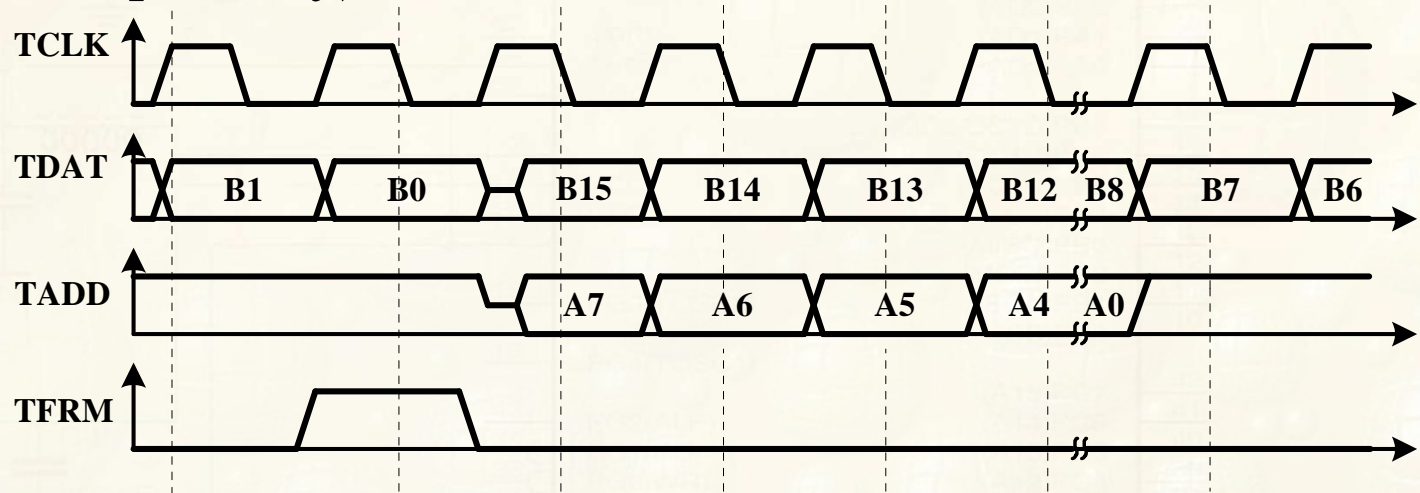
Většina běžných signálových procesorů umožňuje komunikaci mezi dvěma až 8 (32) procesory.

Procesor vysílá v sériovém synchronním módu s rámcovou synchronizací (TFRM) pevný nebo programovatelný počet bitů, případně doplněný o adresu (TADD) v jednom z intervalů. Logickým součinem TADD s interní maskou procesor povoluje nebo zakazuje příjem informací od daného procesoru.

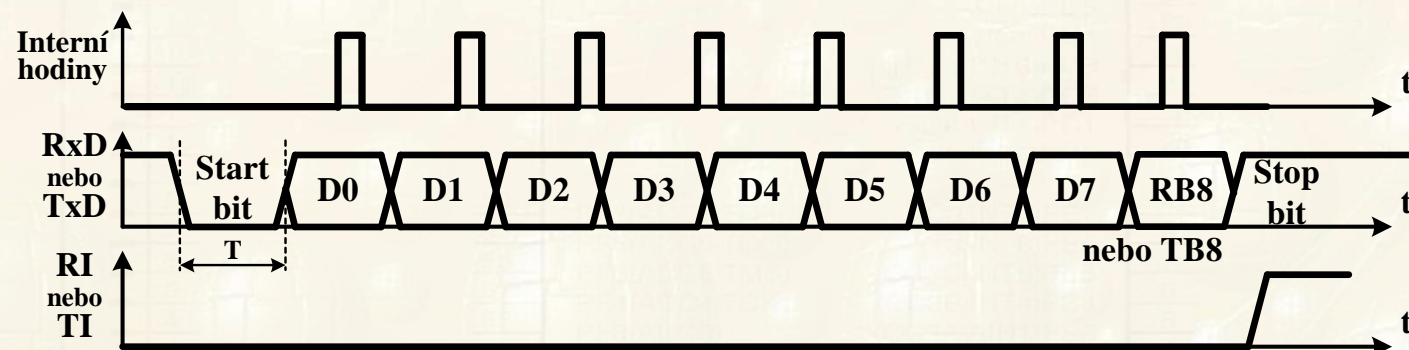


SÉRIOVÁ MULTIPROCESOROVÁ KOMUNIKACE

Příklad sériové multiprocesorové komunikace TDM na procesorech TMS320C54xx - synchronní sériový přenos s rámcovou synchronizací (kontinuální přenosy).

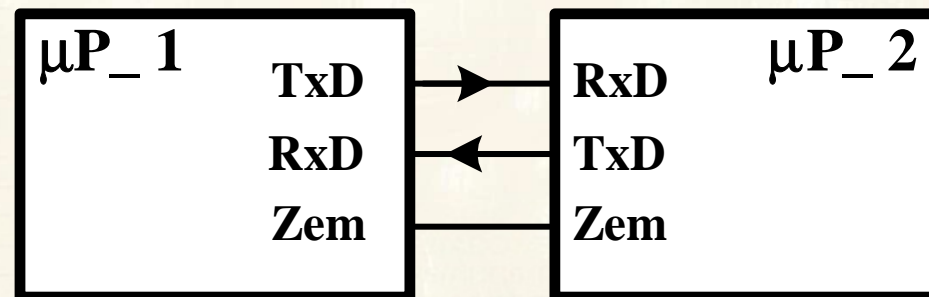


U jednočipových procesorů se multiprocesorová komunikace realizuje v osmibitovém nebo devítibitovém asynchronním sériovém přenosu (přesnost generátorů interního hodinového kmitočtu).



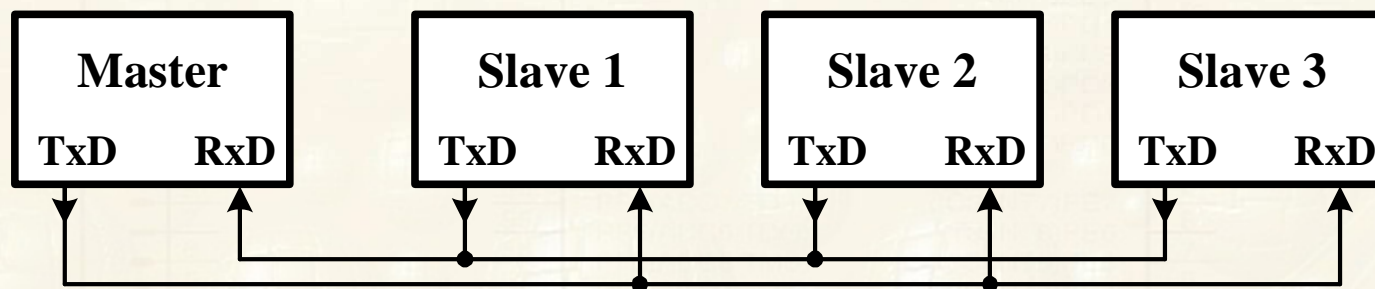
Dvouprocesorový systém komunikující přes sériový kanál.

- V asynchronním módu spojením vývodů RxD1-TxD2, TxD1-RxD2 a zemního vodiče (minimální varianta, full duplex).
- V synchronním módu spojením vývodů TxD1-TxD2 (hodinový signál), RxD1-RxD2 (přenášená data) (half duplex). Možné problémy, není-li **rámcová synchronizace**.
- Přímé propojení na úrovni TTL - vhodné pouze pro plošný spoj.
- Pro pseudoobousměrné vývody TxD (vstup a výstup s otevřeným kolektorem), můžeme realizovat spojení **half duplex** jedním vodičem.



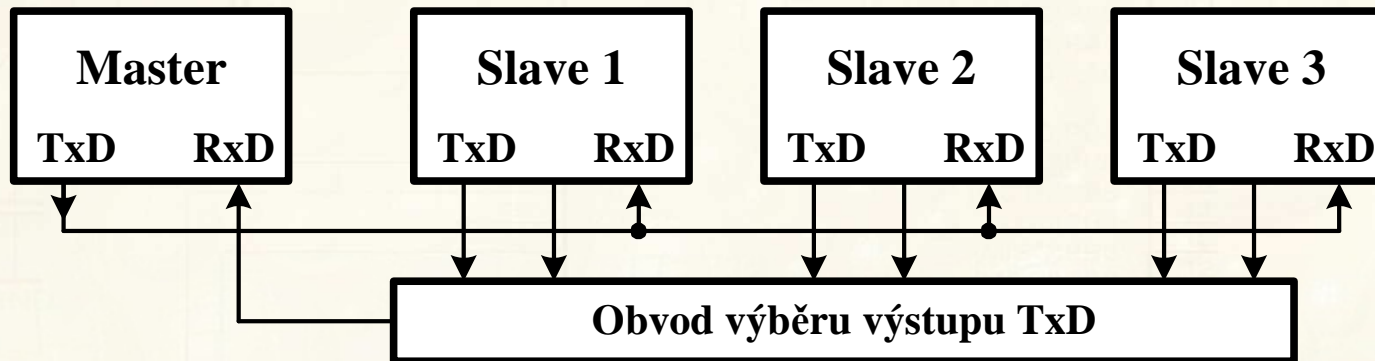
Komunikace mezi několika procesory pomocí sériového kanálu.

- Při propojení většího počtu procesorů je vhodnější využívat asynchronní sériový přenos s **komunikačním protokolem**.
- Každý procesor Slave bude muset mít **přidělenou adresu** a komunikace bude probíhat ve dohodnutém protokolu.
- Možnosti přidělování adres
- Pro vývody TxD pseudoobousměrné a s otevřeným kolektorem můžeme propojit procesory ve variantě **full duplex** pomocí dvou vodičů.



ASYNCHRONNÍ SÉRIOVÁ MULTIPROCESOROVÁ KOMUNIKACE

- Pro dvoustavový výstup TxD – musíme zavést výběr vývodu TxD daného Slave.



- Propojení na větší vzdálenost (10m÷1,2km) je vhodné realizovat **komunikačním rozhraním**
 - **V24 (RS232), RS-485, RS422 - Možné problémy?**
 - RF spojením (433MHz, 868MHz), WiFi, Bluetooth (2,4GHz), ZigBee (2,45 GHz), Zwave.
 - Na větší vzdálenosti modem a telefonní linka, GSM modem, radioreléový spoj.

Pravidla pro multiprocesorovou komunikaci po sériovém kanálu

- ❖ Při komunikaci **half duplex** smí **vysílat** jen jeden procesor.
- ❖ Při komunikaci **full duplex** smí **vysílat master** nebo **slave**, případně master a jeden slave současně.
- ❖ **Komunikační protokol s adresací**
 - s **pevnou délkou**
 - s **proměnnou délkou a ukončujícím znakem**
 - s **proměnnou délkou a informací o délce zprávy**
- ❖ **Nezbytné je rozlišení** adresy Slave od přenášených dat.
- ❖ **Komunikace od procesoru master by neměla nezatěžovat procesory**, kterých se netýká.
- ❖ **Změna směru přenosu** u převodníků s řízením směru by neměla trvala dlouhou dobu (např. vložení intervalu 1ms, apod.)

MULTIPROCESOROVÁ KOMUNIKACE JEDNOČIPOVÝCH μ P

- ❖ Multiprocessorová komunikace typu (Master-Slave).
 - **9 bitová** – pak 9 bit určuje adresu (log.1), data (log.0).
 - **8 bitová** – pak 8 bit odlišuje adresu/data, přenášet můžeme pouze hodnoty (znaky - ASCII) od 00h do 7Fh.

<Adresa>, <1 bytedat>, <2 bytedat>, , <n bytedat>, <ukončující znak>

10001010	1	10101010	0	01010101	0	00000000	0
adresa		55h		AAh		00h	

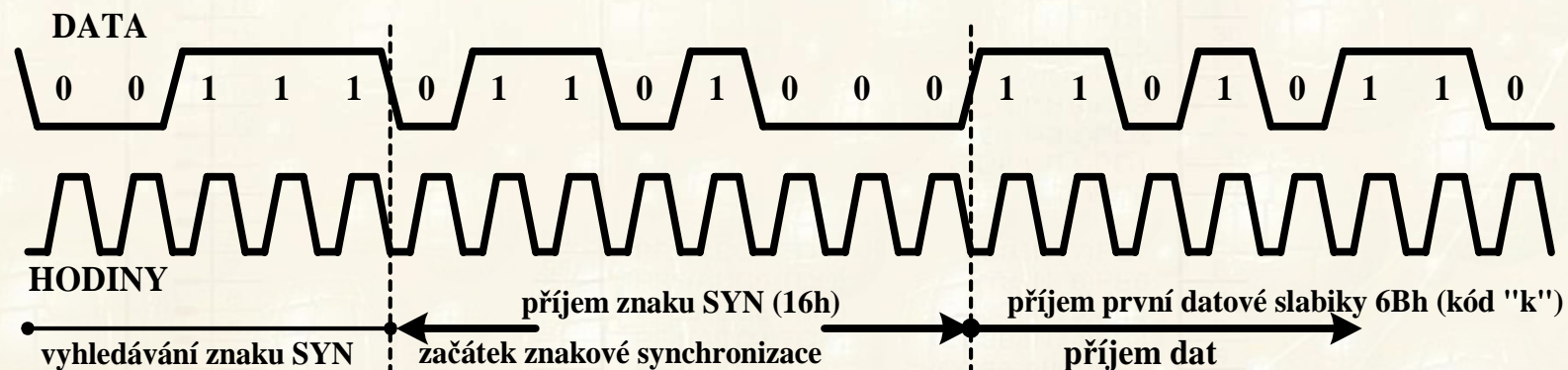
- Zobrazené bitů = LSB po MSB (standardní asynchronní přenos)
- Adresa - délka zprávy (bez ukončujícího znaku)
- **K šetření výpočetního výkonu podřízených procesorů** slouží bit (8051-SM2, AVR-MPCM_n, ARM-WAKE/RWU) povoluje/brání obsluze přerušení pro datový byte (9-bitový režim).
 - **Adresa** vyvolá přerušení u všech podřízených procesorů.
 - **Adresovaný procesor** nastaví bit tak, aby přijímal **datovou informaci**.

Přenos je realizován po sběrnici, jejíž parametry jsou definovány **nultou (fyzickou) vrstvou**.

V **první vrstvě** jsou definovány formáty přenášených dat a komunikační protokoly - pravidla přenosu a způsob zabezpečení.

Sériové přenosy můžeme rozdělit podle bitové synchronizace

- **asynchronní**, bitová synchronizace je nezávisle zajištěná na přijímací i vysílací straně.
- **synchronní**, synchronizační signál pro bitovou synchronizaci se přenáší současně s daty
 - Znakem
 - Rámcovou synchronizací



Podle typu přenášených dat můžeme sériové přenosy rozdělit

- **znakové** - data jsou přenášena alfanumerickým kódem
- **bitové** - data reprezentují skupiny bitů, které mohou, ale nemusí být dále strukturovány.

V **aplikacích s jednočipovými mikroprocesory** se nejčastěji používají

- **firemní standardy.**
- **uživatelské protokoly**

Za přenášenou **adresou** následují **příkazy**

- k nastavení jednotek (stave)
- k odpovědi (identifikace přítomnosti, stavu, atd.)
- k předání naměřených dat

V komunikaci může být zavedena jedna centrální adresa k oslovení všech procesorů najednou (počáteční nulování, zjištění připojených jednotek k jednotlivým adresám, atd.).

POZOR na existenci dvou jednotek se stejnou adresou!!!

- Při propojení bod-bod se stále setkáváme asynchronním sériovým přenosem s rozhraním RS232C (2400 až 115200 Baudů).
- Pro síťové přenosy po vodičích dominuje rozhraní RS485, RS422 (9600 do 57600 Baudů)
- Stále se rozšiřuje škála **bezdrátových síťových prostředků**.
- **Většina protokolů představuje znakově orientované přenosy, jednotlivé znaky v ASCII kódu. Znaky 00h÷1Fh = řídicí znaky.**
- Protokoly využívající RS232C, RS485, RS422 používají 8 nebo 9 bitovou komunikaci.
- Může se jednat o bitový přenos doplněný paritou nebo identifikačním bitem rozlišující adresu (log.1) nebo data (log.0) pro podporu multiprocessorové komunikace.

Pokud přenášené hodnoty jsou 8/9 bitové pak mohou představovat:

- 8 bitovou binární hodnotou, případně + 1bit (parita, adresa/data)
- 7 bitovou hodnotu (ASCII znak) +1 bit (parita, adresa/data)
- 8 bitovou hodnotu rozdělenou na dvě 4-bitové části (vyjádřené ASCII kódem, **INTEL HEX**).

Zabezpečení přenášené informace bývá

- sudou nebo lichou paritou
- blokovou paritou
- kontrolním součtem EX-OR,
- cyklickým CRC polynomem,
- aritmetickým kontrolním součtem modulo 256 (65536) nebo jejich kombinacemi.

Z neznámějších znakových protokolů jmenujme protokol

- **ModBUS** (programovatelné automaty)
- **ADAM** (sběr dat, monitorování a řízení v průmyslových aplikacích)
- **DIN Messbus** a **PROFIBUS** (oblasti měřicí techniky).
- Z bitových protokolů jmenujme například
- **NET92** (zabezpečovací systémy)
- **BITBUS** (aplikacích s povinnou datovou integritou (Ethernet, ISDN, atd.)).
- V současné době vzniká stále větší potřeba propojovat podřízené jednotky mezi sebou nejen k vytváření měřících, monitorovacích, přístupových a bezpečnostních systémů, ale i v oblastech běžného života v domácnostech k měření, regulaci, ale i pro jednoduché ovládání například světel, atd.

Nejjednodušší ovládací obvody obvykle pracují na některé z bezdrátových sítí:

- ZigBee
- Zwave
- WiFi
- Bluetooth

Řada vyráběných obvodů již v sobě obsahuje prostředky pro konfiguraci sítě (např. **ZigBee Stack**). Propojení μ P s těmito obvody se obvykle uskutečňuje přes sériové rozhraní nebo některou z přístrojových sběrnic.

Potřeba propojování podřízených jednotek k monitorovacím systémům způsobila vznik jednodušších i složitějších komunikačních protokolů \Rightarrow

- Komplikace při propojování různých přístrojů do sítí
- Problémy s distribucí informací mimo operátorská stanoviště

Z uvedených problémů vzniká potřeba sjednocování komunikací a minimalizace komunikačních protokolů.

V současnosti stále řada řídicích uzlů s programovatelnými automaty, měřícími a monitorovacími systémy komunikuje v daném komunikačním protokolu, ale přenos dat mimo tyto uzly se prosazuje na bázi komunikačního média

Ethernet - s protokoly na bázi standardu TCP/IP.

Nyní velký rozmach pod označením **IoT – Internet věcí**

- Vznik MIT 1999
- Řada μ P modulů s možností připojení na internet
 - ✓ Možnost monitorování ve veřejném nebo neveřejném protokolu
 - ✓ Vzdálený přístup k realizovaným aplikacím (servis, nastavení)
 - ✓ Možnost dálkové inovace SW
 - ✓ Současně s těmito výhodami **přichází i nebezpečí neoprávněného zásahu do nastavení zařízení apod.**