



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

**STŘEDNÍ ODBORNÁ ŠKOLA ELEKTROTECHNICKÁ,
Centrum odborné přípravy,
Hluboká nad Vltavou**



Elektronika

Identifikace projektu

Název a číslo globálního grantu	Zvyšování kvality ve vzdělávání v Jihočeském kraji CZ.1.07/1.1.10
Registrační číslo projektu	CZ.1.07/1.1.10/01.0015
Název projektu	Inovace a vytvoření odborných učebních textů pro rozvoj klíčových kompetencí v návaznosti na rámcové vzdělávací programy
Název příjemce podpory	Střední odborná škola elektrotechnická, Centrum odborné přípravy, Hluboká nad Vltavou

Hluboká nad Vltavou 2011



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

**Na zpracování učebního textu Elektronika
se podíleli učitelé SOŠE, COP, Hluboká nad Vltavou:**

Boháč Zdeněk

Chaloupek Pavel

Salajka Libor

Donát Josef

Obsah

1 ZÁKLADY ELEKTRONIKY	15
1.1 Základní pojmy v elektronice	15
1.1.1 Elektrické obvody	15
1.1.2 Elektronická zařízení	15
1.1.3 Obvodová součástka	15
1.2 Lineární součástky	16
1.2.1 Rezistory	16
1.2.2 Kondenzátory	17
1.2.3 Cívky	19
1.3 Polovodičové součástky	20
1.3.1 Polovodičová dioda	20
1.3.2 Bipolární tranzistor	20
1.3.2.1 Proudový zesilovací činitel h_{21E}	21
1.4 Zdroje	22
1.4.1 Transformátory	22
1.4.1.1 Princip transformátoru	22
1.4.2 Usměrňovače	23
1.4.2.1 Jednocestný usměrňovač	23
1.4.2.2 Dvoucestný usměrňovač	24
1.4.2.3 Můstkové zapojení usměrňovače – Graetzův můstek	24
1.4.2.4 Jednocestný usměrňovač s kondenzátorem	25
1.4.2.5 Trojfázové usměrňovače	25
1.4.2.6 Zdvojovače napětí	26
1.4.2.7 Celovlnný zdvojovač	26
1.4.2.8 Řízené usměrňovače	26
1.4.3 Filtrační obvody	27
1.4.3.1 Aktivní vyhlazovací filtry	28
1.4.4 Stabilizátory napětí	29
1.4.4.1 Stabilizátory napětí s diskrétními součástkami	30
1.4.4.2 Stabilizátory pevného napětí s integrovanými obvody	30

1.4.4.3 Zdroj souměrného napětí	31
1.4.5 Spínané zdroje	31
1.5 Nízkofrekvenční zesilovače.....	32
1.5.1 Vlastnosti zesilovače	33
1.5.2 Nastavení a stabilizace pracovního bodu tranzistoru v zesilovači	34
1.5.3 Zpětné vazby.....	35
1.5.4 Jednostupňový střídavý zesilovač	37
1.5.5 Korekční zesilovače.....	38
1.5.6 Výkonový zesilovač s komplementárními tranzistory	38
1.5.7 Integrované obvody pro koncové zesilovače	39
1.5.8 Použití unipolárních tranzistorů v zesilovačích.....	39
1.6 Operační zesilovače.....	40
1.6.1 Vlastnosti	41
1.6.1.1 Vstupní napěťová nesymetrie	41
1.6.1.2 Vstupní klidový proud	41
1.6.1.2 Kmitočtová kompenzace	41
1.6.1.3 Rychlost přeběhu	41
1.6.1.4 Obvodové požadavky	41
1.6.2 Neinvertující zesilovač	42
1.6.3 Invertující zesilovač.....	43
1.7. Oscilátory	43
1.7.1 Nízkofrekvenční oscilátory.....	43
1.7.1.1 Dělení nízkofrekvenčních oscilátorů	44
1.7.1.2 Nízkofrekvenční oscilátory RC	44
1.7.2 Vysokofrekvenční oscilátory	46
1.7.2.1 Vznik oscilací	46
1.7.2.2 Colpittsův oscilátor.....	46
1.7.2.3 Parametry vysokofrekvenčních oscilátorů.....	47
1.7.3 Krystalové oscilátory	47
1.7.3.1 Rozdělení podle kmitočtového rozsahu.....	47
1.7.3.2 Rozdělení podle stability kmitočtu	47
1.7.3.3 Hlavní součásti krystalových oscilátorů	47

1.7.3.4 Oscilátor Clappův	48
1.8 Modulace	48
1.8.1 Amplitudová modulace	49
1.8.2 Frekvenční modulace.....	50
1.9 Optoelektronika	51
1.9.1 Optoelektronika – prvky	51
1.9.1.1 LED dioda	52
1.9.1.2 Fotorezistor.....	52
1.9.1.3 Fotodioda	53
1.9.1.4 Fototranzistor.....	53
1.9.1.5 Optron – optoelektrický vazební člen.....	54
1.9.2 Optická vlákna a kabely	54
1.9.2.1 Optické vlákno.....	54
1.9.2.2 Typy optických vláken	55
1.9.2.3 Mnohovidová vlákna (MM, multimode).....	55
1.9.2.4 Jednovidová vlákna (SM, singlmode)	55
1.9.2.5 Opakovací otázky:	56
1.9.2.6 Odpovědi na opakovací otázky:	57
2 ČÍSLICOVÁ TECHNIKA	63
2.1 Základní pojmy.....	63
2.1.1 Analogový a číslicový signál.....	63
2.1.2 Impuls	64
2.1.3 Základní pojmy užívané v číslicové technice.....	65
2.1.4 Číselné soustavy	66
2.2 Logické funkce	68
2.2.1 Základní logické funkce - logický součet, součin, negace	69
2.2.1.1 Logický součet - OR.....	69
2.2.1.2 Logický součin - AND	70
2.2.1.3 Logický operátor NOT - invertor	70
2.2.1.4 Negace logického součtu - NOR	71
2.2.1.5 Negace logického součinu - NAND	71
2.2.2 Booleova algebra	72

Základní vztahy Booleovy algebry	72
Použití Booleovy algebry	74
2.2.3 Minimalizace s užitím Karnaughovy mapy	75
2.3 Logické členy	77
2.3.1 Rozdělení logických členů	77
2.3.2 Základní logický člen integrovaných obvodů TTL	78
2.3.3 Porovnání vlastností jednotlivých rodin logických obvodů	81
2.3.4 Vzájemné spojování obvodů CMOS a TTL	82
2.4 Kombinační logické obvody	84
2.4.1 Návrh kombinačních logických obvodů	84
2.4.2 Základní kombinační logické obvody	85
2.4.2.1 Kodéry a dekodéry	85
2.4.2.2 Dekodéry pro sedmissegmentové displeje	88
2.4.2.3 Multiplexer	91
2.4.2.4 Demultiplexer	93
2.4.2.5 Komparátor	94
2.4.2.6 Binární sčítačka	94
2.5.1 Klopné obvody	97
2.5.1.1 Klopný obvod R-S	97
2.5.1.2 Klopný obvod R-S-T	98
2.5.1.3 Klopný obvod D	100
2.5.1.4 Dvojitě klopné obvody R-S-T a J-K	100
2.5.2 Čítače	102
2.5.2.1 Dvojkové čítače asynchronní	102
2.5.2.2 Dvojkové čítače synchronní	106
2.5.2.3 Integrované čítače	107
2.5.3 Posuvné registry	109
2.5.3.1 Posuvné registry J-K	109
2.5.3.2 Posuvné registry D	110
2.6 Základy mikroprocesorové techniky	111
2.6.1 Struktura mikropočítače	111
2.6.2 Procesor	113

2.6.3 Paměť	114
2.6.4 Otázky k procvičování	119
2.6.5 Odpovědi na otázky za textem	120
3. ZABEZPEČOVACÍ TECHNIKA	123
3.1 Úvod	123
3.2 Systémy elektronického zabezpečení	123
3.3 Termíny a definice hlavních pojmů	124
3.4 Stupně zabezpečení	125
3.5 Rozdělení jednotlivých komponent EZS na prvky:	126
3.6 Plášťová ochrana	127
3.6.1 Magnetické kontakty	128
3.6.2 Detektory skleněných výplní	129
3.7 Prostorová ochrana	130
3.7.1 Detektory infračerveného záření	130
3.7.2 Mikrovlnná čidla	132
3.7.3 Ultrazvuková čidla	133
3.7.4 Kombinovaná čidla	134
3.8 Tísňové hlásiče	135
3.8.1 Automatické tísňové hlásiče	135
3.8.2 Speciální tísňové hlásiče	136
3.8.3 Veřejné tísňové hlásiče	136
3.9 Perimetrická ochrana	137
3.9.1 Infračervené závory a bariéry	137
3.9.2 Perimetrická pasivní infračervená čidla (infrateleskopy)	138
3.10 Ústředny EZS	139
3.10.1 Smyčková ústředna	143
3.10.2 Ústředna s přímou adresací čidel	144
3.10.3 Ústředny smíšeného typu	144
3.10.4 Ústředny s bezdrátovým přenosem od čidel	145
3.11 Doplnková zařízení ústředny	146
3.11.1 Akustická signalizace	146
3.11.2 Optická signalizace	147

3.11.3 Grafické tablo	148
3.11.4 Tiskárny	148
3.12 Poplachová přenosová zařízení – komunikátory	149
3.12.1 Zařízení určená pro informování majitele objektu	149
3.12.2 Zařízení určená pro komunikaci s PCO/PPC	150
3.13 Indikační a ovládací zařízení	150
3.13.1 Kódové klávesnice.....	151
3.13.2 Kombinované indikační a ovládací díly	152
3.13.3 Ovládání kartou	152
3.14 Konstrukční cvičení.....	153
3.15 Praktická úloha	171
3.16 Opakovací otázky	172
3.16.1 Detektory infračerveného záření	172
3.16.2 Jednoduše vyvážená smyčka	174
3.16.3 Dvojitě vyvážená smyčka	174
3.16.4 Ústředna JA-82K OASIS	175
3.16.5. Výstupy sirén.....	176
4 PROGRAMOVÁNÍ JEDNOČIPOVÝCH MIKROPOČÍTAČŮ	177
4.1 Základní pojmy.....	177
4.2 Popis mikrokontroléru PIC16F84A.....	178
4.2.1 Základní vlastnosti.....	178
4.2.2 Popis periférií	178
4.2.3 Popis vývodů	179
4.2.4 Architektura	179
4.2.5 Uspořádání paměti.....	181
4.2.5.1 Paměť programu	181
4.2.5.2 Zásobník návratových adres	181
4.2.5.3 Speciální funkční registry a paměť dat.....	181
4.2.6 Porty	183
4.2.6.1 Port A.....	183
4.2.6.2 Port B.....	184
4.2.7 Čítač / časovač TMR0	184

4.2.8 Přerušení (interrupt).....	185
4.3 Základy jazyka C pro programování mikrokontrolérů	186
4.3.1 Jazyk C	186
4.3.2 Základy jazyka ANSI-C	187
4.3.3 Zápis programu v jazyce C	187
4.3.4 Základní elementy jazyka C	189
4.3.4.1 Klíčová slova	190
4.3.4.2 Identifikátory	190
4.3.4.3 Odsazovače (bílé znaky).....	191
4.3.4.4 Konstanty.....	191
4.3.4.5 Proměnné	191
4.3.4.5 Komentáře	192
4.3.4.6 Příkaz přiřazení.....	193
4.3.4.7 Operátory	193
4.3.5 Řízení toku programu	196
4.3.5.1 Podmíněný operátor if	196
4.3.5.2 Přepínač switch.....	197
4.3.5.3 Cykly	198
Cyklus while	199
Cyklus do - while.....	200
Cyklus for	200
4.3.6 Funkce	201
4.4 Řešené úlohy	204
4.4.1 Nastavení I/O bran.....	204
4.4.1.1 Rozsvícení LED diody	204
4.4.1.2 Rozsvícení LED diody po stisku tlačítka	206
4.4.2 Časové smyčky.....	209
4.4.2.1 Blikání LED diody	209
4.4.2.2 Běžící světlo	211
4.4.3 Čítač, časovač	213
4.4.4 Přerušení	216
4.4.5 Tlačítka	220

4.4.5.1 Multifunkční tlačítko	220
4.4.5.2 Multifunkční tlačítko s ošetřením zákrmitů	221
4.4.5.3 Maticová klávesnice	223
4.4.6 Krokový motor	228
4.4.7 Servomotor	231
4.4.8 Piezoměnič	234
4.4.9 Zobrazovače.....	236
4.4.9.1 Sedmisegmentový zobrazovač	236
4.4.9.2 Sedmisegmentový zobrazovač – multiplexní řízení.....	238
4.4.9.3 LCD zobrazovač	240
4.4.10 Paměť EEPROM	243
4.4.11 Komunikace PIC16F84A pomocí 1-vodičové sběrnice DALLAS	245
4.5. Příloha – Prostředí Mikroelektronika MikroC	248
Mikroelektronika - Založení nového projektu v MikroC for PIC	248
Vložení zdrojového souboru do projektu	250
Překlad (kompilace) zdrojového souboru.....	251
Simulace	251
Rychlý přehled.....	253
Otázky k procvičování.....	253
Odpovědi na otázky za textem.....	254
Test: základy elektroniky.....	257
Test: číslicová technika	261
Test: zabezpečovací technika	265
Test: programování jednočipových mikropočítačů	268
Použitá literatura.....	271
5 Přílohy	273
5.1 Zásady BOZP	273
5.1.1 Obecné zásady bezpečnosti práce.....	273
5.1.2 Zásady ochrany před antistatickou elektřinou	273
5.1.3 Obecné zásady bezpečnosti práce na školním pracovišti	273
5.1.4 Povinnosti žáků	274
5.1.5 Obecné zásady požární bezpečnosti	274

5.1.6 První pomoc při úrazu elektrickým proudem	275
5.2 VF obvody	276
5.2.1 Vysokofrekvenční zesilovače	276
5.2.2 Vysokofrekvenční zesilovač s jednoduchým laděným obvodem.....	276
5.2.3 Vázané rezonanční obvody.....	277
5.3.4 Širokopásmové zesilovače.....	278
5.4 Spínací, impulsní a tvarovací obvody.....	279
5.4.1 Spínací obvody	279
5.4.2 Spínací diody	279
5.4.3 Spínací tranzistor	280
5.4.4 Tvarovací a impulsní obvody	280
5.4.5 Tvarování impulsu pomocí pasivních součástek	282
5.4.6 Tvarování impulsu pomocí aktivních součástek	283
5.5 Převodníky A/D a D/A	284
5.5.1 Převodníky.....	284
5.5.2 Funkce D/A převodníků	284
5.5.3 Typy D/A převodníků.....	285
5.5.4 princip D/A převodníku váhovou strukturou odporové sítě.....	286
5.5.5 Vlastnosti A/D převodníků.....	287
5.5.6 Typy A/D převodníků.....	288
5.6 Tvarovací obvody	290
5.6.1 Schmittův klopný obvod.....	290
5.6.2 Měření, oprava a diagnostika číslicových obvodů	291
5.7 Průmyslová televize – kamerové systémy.....	295
5.8 Programování PIC - IR přenos	298
5.8.1 IR přenos	298
5.8.2 Zjednodušená realizace mikrokontrolérem PIC16F84A	299

1 ZÁKLADY ELEKTRONIKY

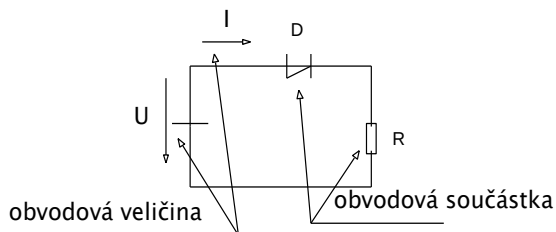
1.1 Základní pojmy v elektronice

1.1.1 Elektrické obvody

Elektronické obvody jsou konstrukční útvary vzniklé spojením elektronických součástek se zdrojem elektrické energie.

1.1.2 Elektronická zařízení

Zpracovávají signály buď analogově nebo číslicově, nebo tyto signály vytvářejí.



Obr. 1.0 Elektrický obvod

1.1.3 Obvodová součástka

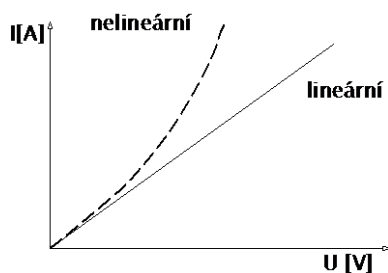
Je neoddělitelná součást obvodu, která má dané elektrické vlastnosti (parametry).

Součástky můžeme posuzovat podle několika hledisek:

a) Podle voltampérové charakteristiky je dělíme na:

Lineární prvky – mají V-A charakteristiku dané elektronické součástky (např. rezistoru) přímkovou.

Nelineární prvky – mají V-A charakteristiku zakřivenou (např. dioda).



Obr. 1.02: VA Nelineárních a lineárních prvků

b) Podle množství vývodů, kterými je prvek připojen rozlišujeme:

dvoupóly – odpory, diody

čtyřpóly – transformátory

vícépóly – integrované obvody

c) Z energetického hlediska (jejich funkce v obvodu) prvky můžeme rozdělit na:

aktivní – v obvodu se chovají jako zdroje. Jejich elektrické vlastnosti jsou proměnlivé a říditelné změnou napětí nebo proudu přivedeného na jejich vývody, zprostředkovávají přenos energie ze stejnosměrného zdroje např. elektronky, tranzistory, operační zesilovače a některé druhy diod.

pasivní – v obvodu se chovají jako spotřebiče elektrické energie. Mají elektrické vlastnosti stálé a v širokých mezích, nezávisle na přiváděném proudu nebo napětí. Jsou to např. rezistory, kondenzátory, cívky, potenciometry atd.

konstrukční uplatňují se v elektronickém zařízení buď funkcí čistě mechanickou (skříně, kostry, panely, ovládací knoflíky aj.) nebo funkcí elektromechanickou. Mezi pomocné materiály zahrnujeme pájecí prostředky (cín, kalafunu).

d) Podle kmitočtové závislosti:

1. kmitočtově závislé prvky-(cívky, kondenzátory)
2. kmitočtově nezávislé prvky-(rezistory)

1.2 Lineární součástky

1.2.1 Rezistory

Rezistor je pasivní elektrická součástka s lineární závislostí, která je v rozsahu běžných frekvencí frekvenčně nezávislá. Rezistory dělíme na vrstvé (uhlíkové a metalizované) a drátové. Základní jednotkou odporu je 1Ω (1 ohm).

Výpočet odporu $R = \frac{U}{I}$

výpočet výkonu $P = U * I$

Vrstvové rezistory tvoří vrstva čistého uhlíku, oxidu kovu nebo odporové slitiny. Odporový materiál je nanesen na keramické tělísko válcového tvaru nebo na trubku. Odpor vrstevného rezistoru je při výrobě nastaven vybroušením šroubovicové drážky do odporové vrstvy, čímž se zvětší délka odporové dráhy, ale vzniká i určitá indukčnost.

Všechny běžné **drátové rezistory** mají indukčnost, proto se hodí pouze pro použití v obvodech se stejnosměrným proudem, popřípadě při nízkého kmitočtu. Vrstvové rezistory, vyrobené vybrušováním šroubovicové odporové dráhy, mají menší vlastní indukčnost a jsou tedy také kmitočtově méně závislé.

Rezistory se používají jako vazební prvky mezi obvody, jako prvek pro srážení napětí, jako útlumové články, jako náhradní zátěže nebo jako děliče napětí v měřidlech, jako předřadné odpory nebo bočníky.

Při **sériovém řazení rezistorů** se celkový odpor sčítá. Při **paralelním řazení** se celkový odpor zmenšuje. Při řazení kombinovaném musíme řešit obvod postupně, po částech.

Rezistory dělíme na pevné, nastavitelné a proměnné. Pevné rezistory mají odporovou vrstvu chráněnou speciálním ochranným lakem, smaltem, nebo je povrch zastříknut do plastické hmoty.

Nastavitelné rezistory mají tři vývody upravené tak, že prostřední vývod lze posunovat po odporové vrstvě a tím trvale nastavit potřebnou hodnotu odporu. Proměnnými rezistory jsou buď potenciometry (plynulá změna pomocí ovládacího knoflíku), nebo potenciometrické trimry (nastavitelné pomocí nástroje). Potenciometry se mohou lišit konstrukcí (např. otočné, přímé, dvojité, tandemové), hodnotou, velikostí a průběhem odporové dráhy (lineární, logaritmická, speciální aj.).

1.2.2 Kondenzátory

Kondenzátory jsou dvojpólové součástky konstruované takovým způsobem, aby vykazovaly kapacitu definované velikosti. Jsou tvořeny dvěma vodivými elektrodami, které jsou navzájem odděleny dielektrikem. Stejnosměrný proud kondenzátorem neprochází a pro střídavý proud představuje kondenzátor určitý střídavý odpor – kapacitní reaktanci. Základní jednotkou kapacity je Farad [F].

$$X_c = \frac{1}{\omega C}$$

Podle použitého dielektrika rozdělujeme kondenzátory na tyto hlavní skupiny: vzduchové, s papírovým dielektrikem, z metalizovaného papíru, s plastickou fólií, slídové, keramické, skleněné, elektrolytické.

Podle konstrukce rozlišujeme kondenzátory na:

s pevnou hodnotou (s kapacitou, kterou nelze měnit)

s proměnnou kapacitou (ladicí a dolad'ovací)

Řazením kondenzátorů paralelně se celková kapacita zvětšuje. Sériovým řazením kondenzátorů se kapacita zmenšuje.

Použití kondenzátorů v elektronických obvodech je velice časté. Kondenzátor může sloužit jako vazební prvek mezi stupni zesilovače. V napájecích obvodech bývá základním filtračním prvkem. Kondenzátor slouží k blokování nežádoucích rušivých signálů. Bývá součástí laděných (rezonančních) obvodů.

U všech druhů kondenzátorů je nutné nepřekročit povolené napájecí napětí. U kondenzátorů elektrolytických je důležité dodržet polaritu vývodů. Obsahují elektrolyt a dielektrikum je tvořeno tenkou vrstvou oxidu hliníku. Proti ostatním druhům kondenzátorů mají elektrolytické kondenzátory nejmenší vnitřní odpor. Tento odpor během stárnutí elektrolytického kondenzátoru dále klesá.

Zvláštním druhem elektrolytických kondenzátorů jsou kondenzátory tantalové. Jejich vlastnosti jsou stálejší a mají menší ztráty a menší rozměry.

Ladící kondenzátory jsou tvořeny soustavou pevných, statorových desek, izolovaných od soustavy desek rotorových. Mezi statorové desky se při otáčení osy zasouvá soustava desek rotorových. Vzájemným překrýváním desek se kapacita kondenzátoru zvětšuje. Mezi rotorem a statorem zůstává buď malá vzduchová mezera (kondenzátor vzduchový), nebo se rotorové desky zasouvají mezi fólie ze speciálních dielektrických materiálů, kterými jsou statorové desky proloženy. Ladící kondenzátory se vyrábějí jako jednoduché, dvojité i trojitě.

Dolad'ovací kondenzátory – kondenzátorové trimry se vyrábějí jako skleněné nebo keramické trubičky, na povrchu postříbřené. Druhou elektrodu tvoří kovový píst, který se do trubičky zašroubovává. Podle velikosti trimru může být nastavená kapacita 0,3pF až 1pF,

nejvýše 1,5pF až 15pF. Vyrábí se i kondenzátorové trimry s kapacitami desítek až stovek pikofaradů a provozním napětím do stovek voltů.

1.2.3 Cívky

Cívky jsou dvoupólové součástky zhotovené vinutím závitů, aby vytvořily vlastní indukčnost L . Jednotkou indukčnosti je 1 H (Henry). Cívku vytvářejí závity vodiče, které jsou uspořádány do jedné nebo několika vrstev. Prostor, který závity obepínají, má obvykle kruhový, čtvercový nebo obdélníkový průřez. Indukčnost cívek závisí na počtu závitů, jejich geometrickém uspořádání a na magnetických vlastnostech prostředí, které závity obepínají i které cívky obklopuje. Stejnosměrný proud cívkou prochází a je omezen pouze odporem vodiče vinutí. Pro střídavý proud představuje cívka střídavý odpor, nazývaný indukční reaktance. Protože cívkou stejnosměrný proud prochází, ale střídavý proud je tlumen, nazýváme indukčnosti pracující v obvodu tímto způsobem tlumivkami. Čím větší má cívka indukčnost a čím vyšší je kmitočet proudu, který cívkou prochází, tím má cívka větší reaktanci.

$$X_L = \omega L$$

Podle konstrukce je možné rozdělit cívky na cívky bez jádra a cívky s jádrem. Vysokofrekvenční cívky jsou buď bez jádra (vzduchové), nebo s jádrem z feritu tvaru trnu, doladovacího šroubu, hrníčku aj.

Cívky se v elektronických obvodech vyskytují velice často. Setkáváme se s nimi v laděných obvodech oscilátorů, radiových přijímačů, televizorů aj. Při řazení cívek počítáme výslednou indukčnost podobně jako odpor u rezistorů. Jsou-li dvě cívky v těsné blízkosti, vzniká mezi nimi indukční vazba. Tohoto jevu se využívá u transformátorů.

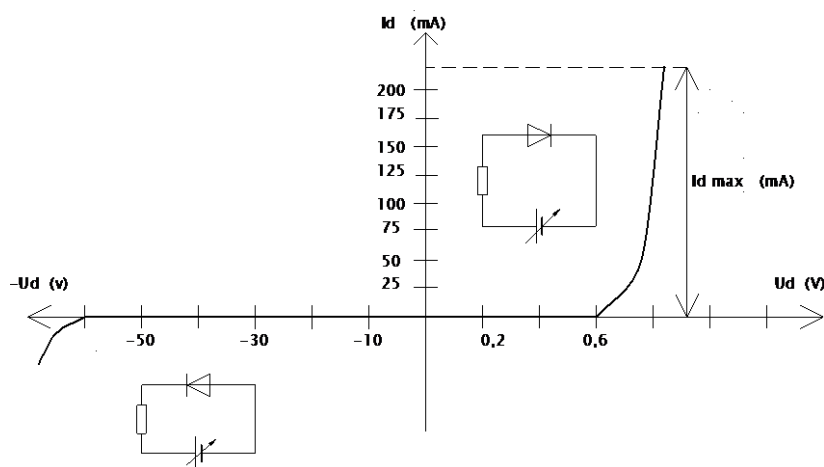
Transformátor má dvě vinutí – cívku primární a cívku sekundární. Transformuje se střídavé napětí, střídavý proud a též impedance.

Relé patří mezi elektrotechnické součástky používané jako řízené spínací prvky. Malým proudem se ovládá elektromagnet, přepínající soustavu galvanicky oddělených výkonových kontaktů.

1.3 Polovodičové součástky

1.3.1 Polovodičová dioda

Polovodičová dioda je polovodičová součástka s jedním přechodem PN, která propouští proud pouze v jednom směru. Aby vůbec došlo k otevření diody, musíme nejprve překonat tzv. prahové napětí (asi 0,6V). Zbytkový proud v závěrném směru lze většinou zanedbat. S rostoucí teplotou závěrný proud narůstá. Reálnou diodu lze napěťově namáhat jen do určitého napětí a potom může nastat průraz (zda bude destruktivní nebo nedestruktivní, záleží na velikosti výkonu, který se na přechodu změní v teplo – především tedy na velikosti procházejícího proudu diodou).



Obr. 1.03: Voltampérová charakteristika diody

Nejčastější použití diod:

diody pro všeobecné - usměrnění malých signálů;

stabilizační diody (Zenerovy diody) - pro stabilizaci napětí;

kapacitní diody (varikap) - pro ladění rezonančních obvodů;

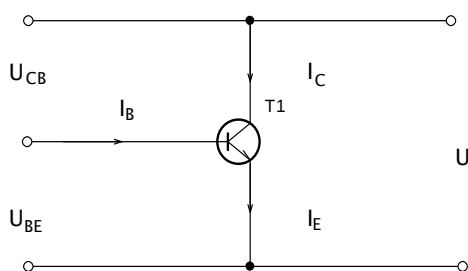
fotodiody využívají toho, že polovodičový přechod je citlivý na osvětlení;

luminiscenční diody (LED) při průchodu proudu svítí, protože při rekombinaci páru díra – elektron se uvolňuje energie ve formě fotonů.

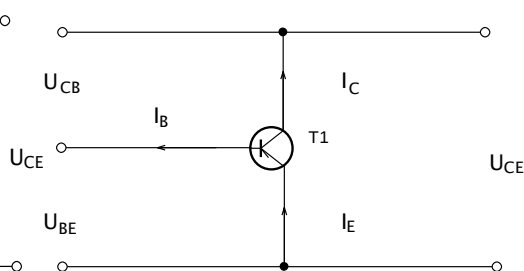
1.3.2 Bipolární tranzistor

Bipolární tranzistor je polovodičová součástka se dvěma přechody PN umožňující malým proudem báze řídit mnohem větší proud mezi kolektorem a emitorem. Tranzistor může pracovat

jako zesilovač i jako spínač. Jako základní materiál je použit většinou křemík. Mohou být typu NPN nebo PNP. Tranzistor má tři elektrody, které se nazývají báze (B), emitor (E) a kolektor (C). Proudů označujeme písmenem I a příslušným indexem I_B , I_C a I_E . Emitorem protéká proud báze i kolektoru ($I_E = I_B + I_C$). Napětí označujeme písmenem U s označením elektrod, uvádějícím elektrody tranzistoru U_{BE} , U_{CB} , U_{CE} . Přechod mezi bází a emitorem je polován v propustném směru a jeho napětí je menší než 1 V ($U_{BE} \approx 0,7$ V).

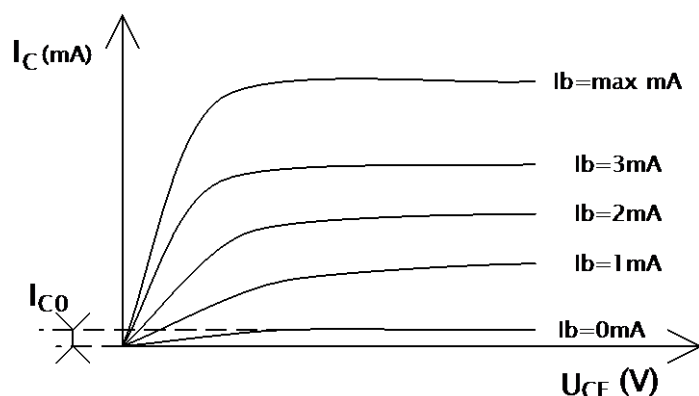


Obr. 1.04: Tranzistor NPN



Obr. 1.05: Tranzistor PNP

Výstupní část voltampérové charakteristiky znázorňuje závislost kolektorového proudu na napětí kolektor-emitor při konstantním proudu do báze.



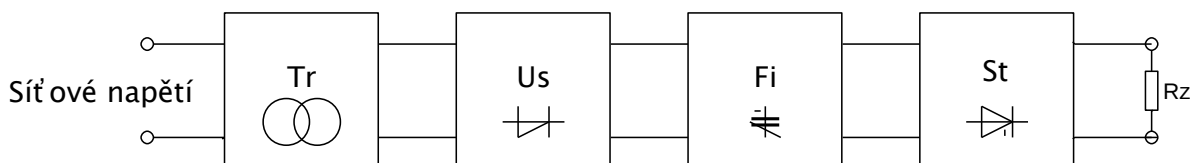
Obr. 1.06: Výstupní charakteristika tranzistoru

1.3.2.1 Proudový zesilovací činitel h_{21E}

Je základním parametrem, který charakterizuje vlastnosti tranzistoru. Vyjadřuje proudové zesílení tranzistoru, tj. kolikrát větší proud protéká kolektorem než bází. Je to bezrozměrné číslo, jeho velikost závisí na typu tranzistoru. Skutečnou hodnotu zjistíme měřením.

1.4 Zdroje

Elektrická zařízení, která nemohou být napájena síťovým napětím, to je 230 V/50 Hz z důvodů technických či bezpečnostních se napájí tzv. napájecími zdroji, které mají výstupní napětí snížené, popř. usměrněné a vyhlazené a stabilizované.



Obr. 1.07: Blokové schéma stabilizovaného zdroje

Tr - transformátor mění velikost napětí na požadovanou velikost

Us - usměrňovač mění střídavé napětí na stejnosměrné pulzní

Fi - filtrace odstraňuje zbytky střídavého napětí a snižuje zvlnění v závislosti na provedení

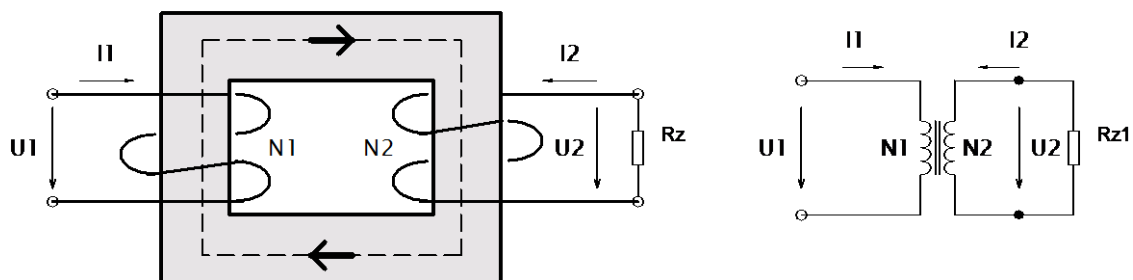
St - stabilizátor odstraňuje zbylé zvlnění a stabilizuje výstupní napětí na požadovanou úroveň

1.4.1 Transformátory

Transformátor je elektrický netočivý stroj, který pomocí elektromagnetické indukce mění elektrické napětí a proud při nezměněném kmitočtu.

1.4.1.1 Princip transformátoru

Po připojení vstupního vinutí na střídavé napětí začne vstupním vinutím protékat střídavý elektrický proud, který vytvoří střídavé magnetické pole. Magnetickým obvodem prochází střídavý magnetický tok a protíná vodiče výstupního vinutí. Ve vodičích výstupní cívky se indukuje střídavé elektrické napětí a po připojení zátěže začne výstupním vinutím protékat elektrický proud. První cívka N_1 je připojena na napětí U_1 , které transformuje. Na sekundární cívce N_2 je transformované napětí U_2 a je k ní rovněž připojen spotřebič.



Obr. 1.08: Zapojení transformátoru

1.4.1.2 Hlavní části transformátoru

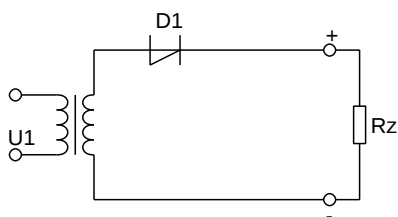
Magnetický obvod - cívky vinutí (izolovaný Cu vodič na kostřičce z izolačního materiálu), ocelokřemíkové plechy, vzájemně izolované, stahovací konstrukce, svorkovnice, štítek s označením údajů.

1.4.2 Usměrňovače

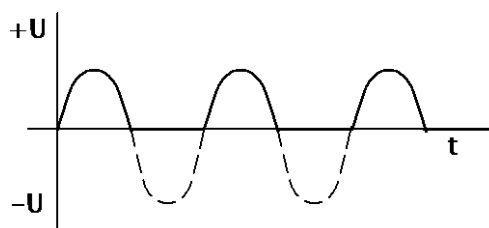
Usměrňovač je zařízení, které ze střídavého napětí vytváří napětí stejnosměrné pulzující.

1.4.2.1 Jednocestný usměrňovač

Do cesty je vložena dioda, která propouští proud jen jedním směrem. V zapojení je jen jedna dioda, usměrněné napětí je pulzní, tepavé. Záporná půlvlna zůstává nevyužita.

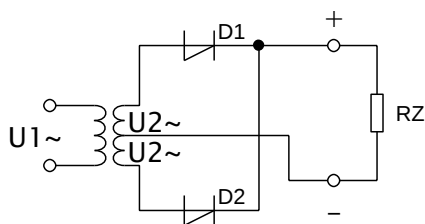


Obr. 1.09: Jednocestný usměrňovač

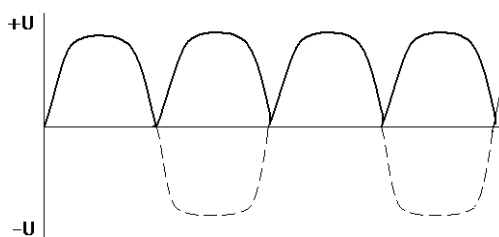


Obr. 1.10: Průběh výstupního napětí

1.4.2.2 Dvoucestný usměrňovač



Obr. 1.11: Dvoucestný usměrňovač



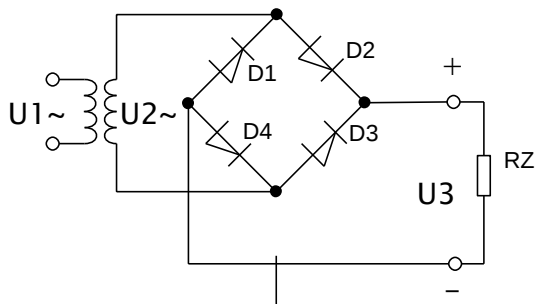
Obr. 1.12: Průběh výstupního napětí

Vyžaduje dvě usměrňovací diody a transformátor s dvojitým sekundárním vinutím, kterým se získávají dvě střídavá napětí, vzájemně fázově obrácená. Na obrázku zapojení jsou vyznačeny i průběhy vstupního střídavého napětí a výstupního usměrněného napětí.

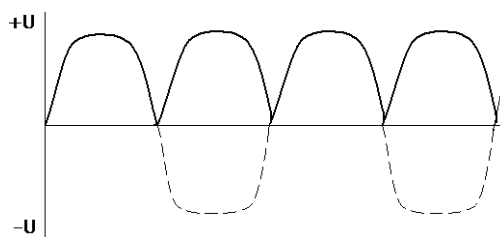
Protože každá z diod usměrňuje jednu půlvlnu střídavého napětí a usměrněné půlvlny z obou diod se střídají, je na výstupních svorkách stále napětí, s výjimkou okamžiku, kdy střídavé vstupní napětí prochází nulou.

Zvlnění usměrněného napětí je menší než u jednocestného usměrňovače. Usměrňovač využívá obou půlvln napájecího napětí

1.4.2.3 Můstkové zapojení usměrňovače – Graetzův můstek



Obr. 1.13: Můstkový usměrňovač



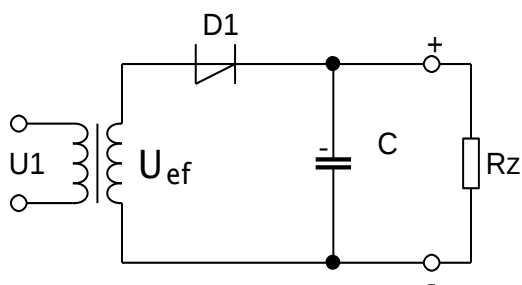
Obr. 1.14: Průběh výstupního napětí

Usměrnění obou půlvln střídavého proudu transformátoru s primárním a sekundárním vinutím umožňuje můstkový usměrňovač. V první půlperiodě střídavého napětí je na horní svorce usměrňovacího můstku kladné napětí. Diody D2 a D4 jsou otevřené, takže může procházet proud. Polarita napětí na výstupních svorkách je patrná z obrázku. Diody D1 a D3 jsou zavřené.

Při následující půlperiodě střídavého proudu je na horní svorce usměrňovacího můstku záporné napětí, na dolní svorce je však kladné napětí. Nyní jsou otevřeny diody D1 a D3, které na horní výstupní svorku přivádějí kladné napětí a na dolní svorku záporné napětí.

Na horní svorce je tedy trvale kladné napětí, na dolní svorce záporné napětí. Jeho průběh odpovídá průběhu napětí při dvoucestném usměrnění.

1.4.2.4 Jednocestný usměrňovač s kondenzátorem

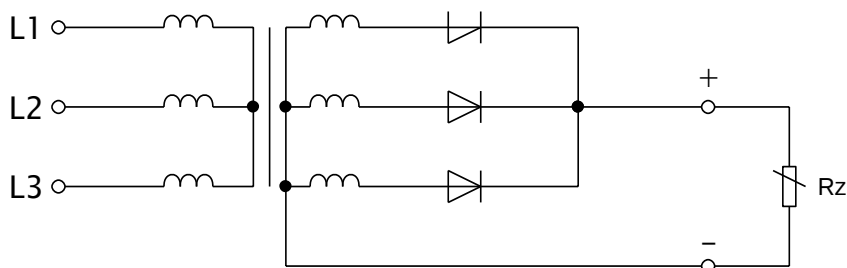


Obr. 1.15: Jednocestný usměrňovač s kondenzátorem

Je-li horní konec sekundárního vinutí síťového transformátoru kladný (je-li na něm kladná půlvlna střídavého napětí), vede dioda D1 a nabíjí se kondenzátor. Napětí na kondenzátoru se rovná špičkovému sekundárnímu napětí transformátoru sníženému o úbytek napětí na diodě D1. Je-li na horním konci sekundárního vinutí záporná půlvlna, dioda nevede. Je-li to například křemíková dioda, bude úbytek napětí asi 0,7V. Pro výše uvedený případ ($U_{ef} = 7V$) bude tedy napětí na kondenzátoru $(1,4U_{ef}) - 0,7V = 9,1V$.

1.4.2.5 Trojfázové usměrňovače

Napájecí trojfázový transformátor je zapojen do hvězdy. Usměrňovač si můžeme představit jako tři samostatné jednocestné usměrňovače, které postupně usměrňují fázově posunutá napětí z výstupních vinutí transformátoru. Zvlněné usměrněné napětí má trojnásobnou frekvenci než má napájecí napětí.



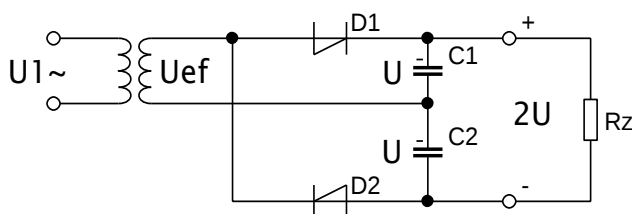
Obr. 1.16: Trojfázový jednocestný usměrňovač

1.4.2.6 Zdvojovače napětí

Používají se, když potřebujeme získat vyšší stejnosměrné napětí než je hodnota výstupního napětí transformátoru

1.4.2.7 Celovlnný zdvojovač

V jedné půlvlně střídavého napětí se nabíjí kondenzátor C1 na napětí U . Během půlvlny opačné polarity se nabíjí na napětí U kondenzátor C2. Protože jako výstupní svorky slouží kladný pól kondenzátoru C1 a záporný pól kondenzátoru C2, napětí na C1 a C2 se sčítají a výstupní napětí je $2U$.



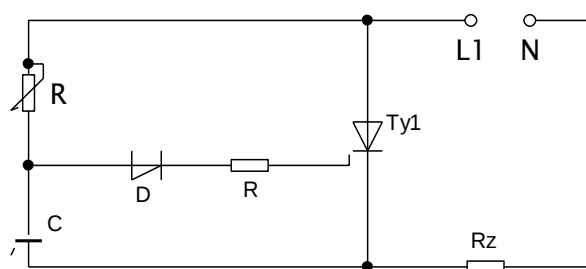
Obr. 1.17: Celovlnný zdvojovač

1.4.2.8 Řízené usměrňovače

Tyto součástky se liší od předchozích tím, že se používají zejména ke spínání střídavých napětí. Pro jejich uvedení do sepnutého stavu stačí přivést krátký proudový impuls do jejich řídicí elektrody (tyristor, triak). Diak spíná samovolně při překročení určité hodnoty napětí na jeho elektrodách. Vypínání tyristoru a triaku se děje samovolně při poklesu procházejícího proudu pod přídržnou hodnotu, u diaku při poklesu napětí. Je zřejmé, že při použití v stejnosměrných obvodech nevypnou samovolně, potřebují na to zvláštní obvody.

Tyristor

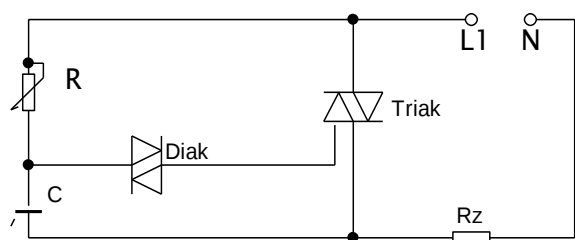
Je to čtyřvrstvá struktura PNPN nebo NPNP, která se chová jako dioda, u které je možno stanovit okamžik sepnutí v rozsahu půlperrody. Řídicí signál má kladnou polaritu. Záporná půlperioda tyristorem neprochází. Jednoduché řídicí obvody a průběhy napětí jsou na obrázku. Dnes jsou řídicí obvody provedeny v integrované podobě.



Obr. 1.18: Zapojení tyristoru s řídícím obvodem

Triak

Triaku řídí okamžik sepnutí během celé periody řídícími signály obou polarit. Tyto požadavky splňuje pětivrstvá struktura NPNPN. V náhradním schématu se vlastně jedná o dva antiparalelně zapojené tyristory. Na obrázku je opět uvedeno schéma řídícího obvodu, kde vidíme i využití diaku.

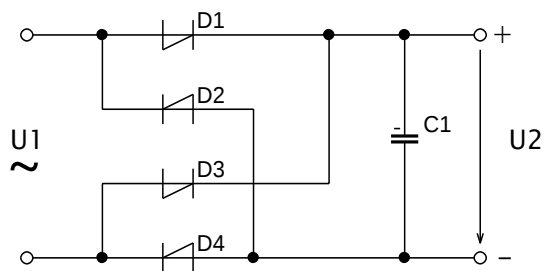


Obr. 1.19: Zapojení triaku s řídícím obvodem

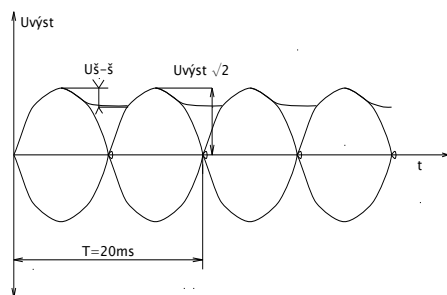
Tyto obvodové prvky umožňují téměř bezetrátovou regulaci výkonu připojené zátěže. Jejich nevýhodou je vysoká úroveň rušivých signálů, zejména při indukční zátěži. Problém je možno řešit použitím odrušovacích LC členů. Další možností je spínání při průchodu signálu nulou, výkon je pak regulován poměrem sepnutých a nesepnutých period.

1.4.3 Filtrační obvody

Filtrací rozumíme odstranění (nebo zmenšení) střídavé složky z usměrněného napájecího napětí. Nejčastěji se pro tento účel využívá kondenzátorů, rezistorů a tlumivek. Typickým příkladem je filtrační kondenzátor připojený k výstupu usměrňovače.

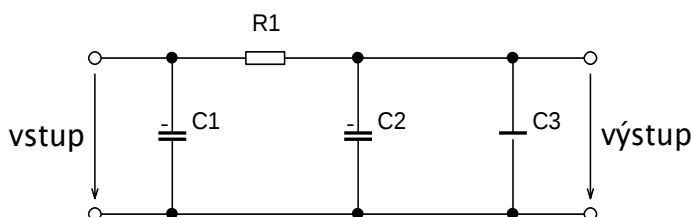


Obr. 1.20: Zapojení usměrňovače společně s filtračním obvodem



Obr. 1.21: Průběh výstupního napětí na výstupu filtračního obvodu

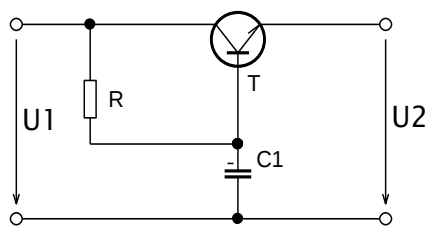
Filtrační řetězce se používají pro dosažení nižších hodnot zvlnění. V sérii je vložen rezistor. Je třeba počítat s úbytkem napětí na tomto prvku. Takové zapojení není vhodné do obvodů s velkým proudovým odběrem. Filtrační řetězce bývají často součástí zapojení daného obvodu. Tlumivka je využíváno zejména k odfiltrování vyšších frekvencí, například ve vysokofrekvenčních zesilovačích nebo spínaných zdrojích. Tlumivka klade střídavé složce signálu daleko vyšší odpor (daný její reaktancí) než stejnosměrnému proudu. Keramický kondenzátor zlepšuje vlastnosti filtru na vysokých kmitočtech.



Obr. 1.22: Zapojení filtračního obvodu s kombinací R a C

1.4.3.1 Aktivní vyhlazovací filtry

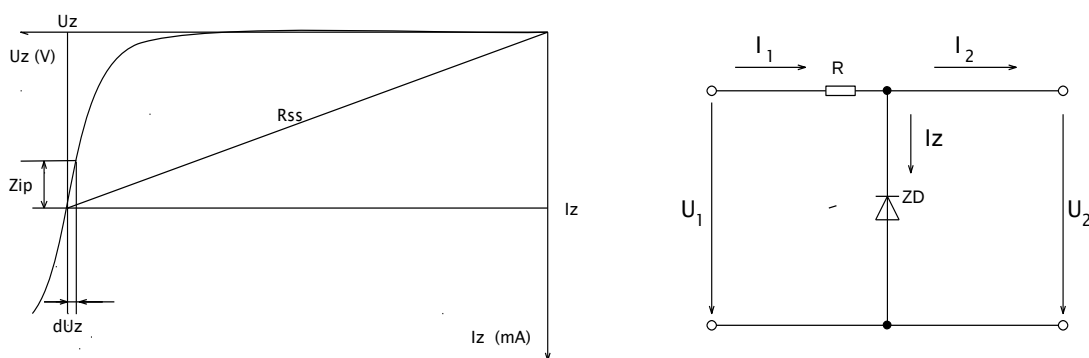
Tyto filtry jsou složitější, protože obsahuje výkonový tranzistor s velkým proudovým zesílením a pasivní RC člen. Někdy jsou také nazývány násobiči kapacity. Kapacita kondenzátoru zapojeného v bázi tranzistoru je násobena proudovým zesilovacím činitelem tranzistoru. Uvedeného zapojení není odolné proti zkratu Na výstupu mohou být zapojeny stabilizátory napětí, jejichž druhotnou funkcí je také filtrace napájecího napětí a odolnost proti zkratu.



Obr. 1.23: Zapojení elektronického filtru

1.4.4 Stabilizátory napětí

Podle připojení regulačního prvku rozdělujeme stabilizátory na paralelní a sériové. Nejjednodušším paralelním stabilizátorem je zapojení se Zenerovou diodou. Zenerova dioda je součástka, u níž dochází v závěrném směru k rychlému nárůstu proudu při pomalém nárůstu napětí. Tato hodnota se nazývá Zenerovo napětí. Používají se do 0,1 A.



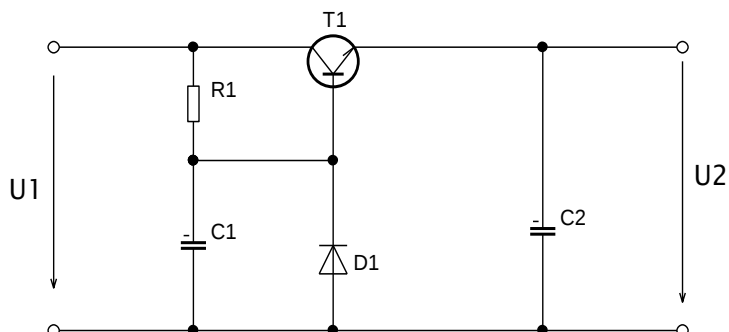
Obr. 1.24: VA Charakteristika a zapojení paralelního stabilizátoru se zenerovou diodou

Velikost odporu R určíme ze vztahu
$$R = \frac{(U_1 - U_2)}{I_1} \quad [\Omega; V, A]$$

Obvykle volíme $I_F > I_2$ tak, aby při $I_2 = 0$ nebyla dioda přetížena proudem I_F , a naopak, aby při maximální hodnotě I_2 kdy začíná napětí na diodě prudce klesat, neklesl proud diodou pod minimální hodnotu.

1.4.4.1 Stabilizátory napětí s diskretními součástkami

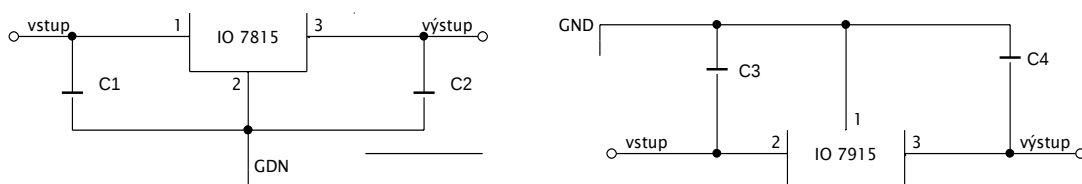
Jednoduchým doplněním stabilizátoru se Zenerovou diodou vznikne sériový stabilizátor. Regulačním prvkem je v tomto případě tranzistor, který je zapojen v sérii se zátěží.



Obr. 1.25: Zapojení sériového stabilizátoru s tranzistorem a Zenerovu diodou

1.4.4.2 Stabilizátory pevného napětí s integrovanými obvody

Pro dnešní realizaci zdrojů používají specializované integrované obvody určené jako stabilizátory stejnosměrného kladného i záporného napětí v rozsahu od 2 do 24 V s výstupním proudem 0,1 až 3 A (tzv. monolitické). Podle výstupního proudu jsou dodávány v pouzdrech se třemi vývody TO92 nebo v pouzdrech TO220 s možností připevnění na chladič. Podle doporučení výrobce je obvody nutné doplnit kondenzátory zajišťujícími stabilitu. Vstupní napětí je 35 až 40 V, respektive -35 až -40 V. Integrované obvody obsahují proudovou ochranu a jsou tedy odolné proti zkratu na výstupu. Běžné stabilizátory řady 78xx a 79xx vyžadují pro správnou činnost, aby rozdíl mezi vstupním a výstupním napětím byl minimálně 3 V.

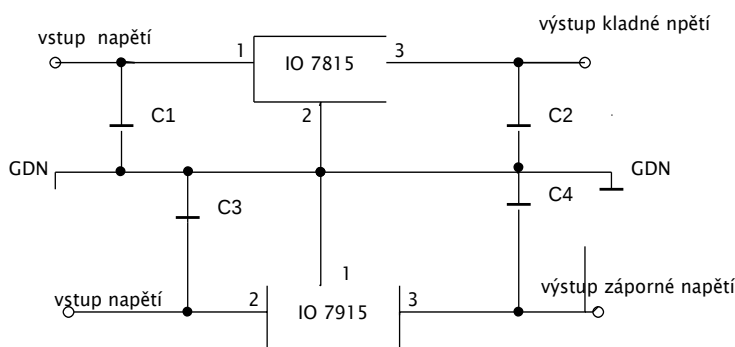


Obr. 1.26: Zapojení stabilizátorů s integrovaným obvodem pro kladné a záporné napětí

Pro stabilizátory nastavitelným napětím používáme integrované obvody například LM317T pro kladné napětí a LM337T pro záporné napětí. Jsou vyráběny v pouzdrech TO220. Výstupní napětí lze regulovat přibližně v rozsahu 1,2 až 37 V a maximální výstupní proud být až 1,5 A. rozdíl mezi vstupním a výstupním napětím musí být minimálně 3V.

1.4.4.3 Zdroj souměrného napětí

K napájení operačních zesilovačů se používá souměrného napětí. Znamená to, že potřebujeme kladné i záporné napětí, jehož velikost je shodná. Jedná se o dva zdroje zapojené v sérii s vyvedeným středem, který je uzemněn.



Obr. 1.27: Zapojení symetrického zdroje integrovanými obvody

1.4.5 Spínané zdroje

Vstupní síťové napětí je usměrněno a vyfiltrováno. Poté je toto stejnosměrné napětí spínáno pomocí výkonových tranzistorů a přiváděno na primární vinutí transformátoru. Kmitočet dosahuje desítek kHz a je řízen oscilátorem (OSC). Velikost dodávané energie je v závislosti na momentální zátěži řízena modulátorem (PWM). Řídící napětí se získává porovnáním (COMP) výstupního napětí s referenčním (REF). Sekundární napětí transformátoru je opět usměrněno a filtrováno.

V praxi se můžeme setkat se spínanými zdroji, které jsou jednodušší a neobsahují všechny popsané části. Na podobném principu pracují také zdroje sloužící k získání různých pomocných napětí jiné velikosti nebo polarity.

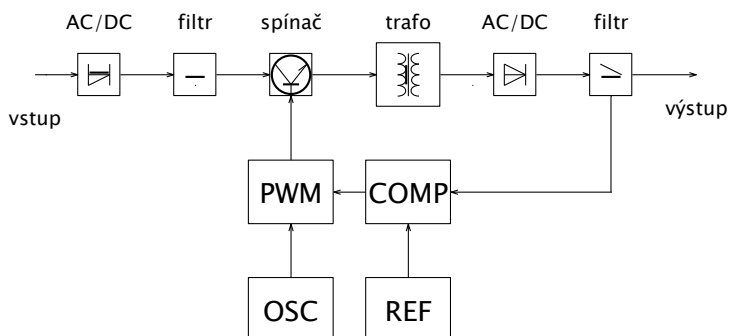
Zmenšení velikosti a hmotnosti je umožněno tím, že použitý transformátor pracuje na mnohem vyšším kmitočtu, než je kmitočet síťový. Má zpravidla feritové jádro a je mnohonásobně menší než běžný síťový transformátor stejného výkonu.

Pokud není vyžadováno galvanické oddělení vstupního a výstupního napětí, využívá se místo transformátoru pouze tlumivky. Popsaná činnost klade značné nároky na použité součástky a vyžaduje použití speciálních rychlých polovodičových prvků s vysokým závěrným napětím. Ke galvanickému oddělení obvodů se používá optočlenů a k řízení specializovaných integrovaných obvodů.

Používají se ve spotřební elektronice nebo výpočetní technice.

Jejich hlavními výhodami je vysoká účinnost, malé rozměry a hmotnost.

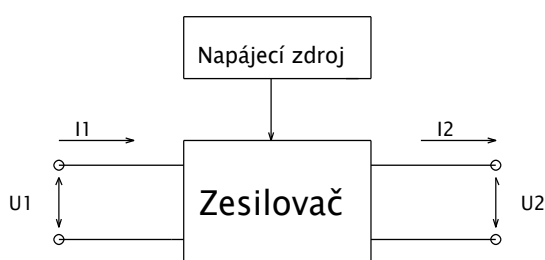
Nevýhodou je jejich složitost, komplikovanější návrh a vyšší počet součástek.



Obr. 1.28: Blokové schéma spínaného zdroje

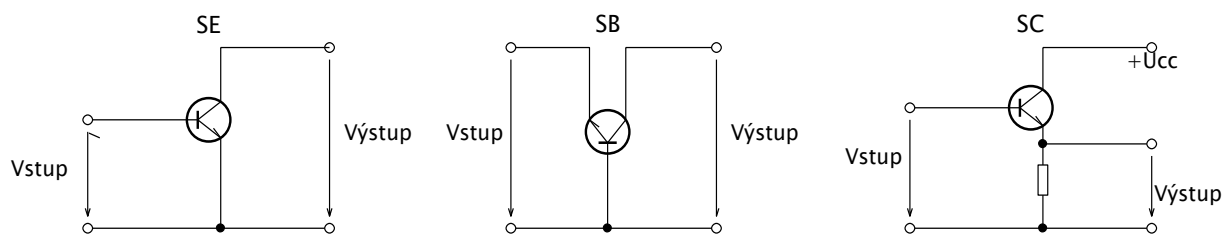
1.5 Nízkofrekvenční zesilovače

Zesilovač je elektronické zařízení, které zesiluje elektrický signál. Má vstup a výstup, je to tedy čtyřpól, na jehož vstupní svorky přivádíme signál U_1 , který chceme zesílit. Zesilovač ke své činnosti vyžaduje napájecí napětí. Část přivedené elektrické energie se mění ve ztrátové teplo a mluvíme tedy o jeho účinnosti. Tato vlastnost je podstatná zejména u výkonových zesilovačů. Vstupní i zesílený výstupní signál mají určitou velikost napětí a mohou dodávat určitý proud. Součin výstupního napětí a proudu udává výkon, který je zesilovač schopen odevzdat do zátěže.



Obr. 1.29: Základní blokové zapojení zesilovače

Jako základní aktivní prvek se v zesilovačích používá bipolární tranzistor. Protože tranzistor má pouze tři vývody, je jeden z nich využíván společně pro vstup i výstup a hovoříme o třech základních zapojeních - se společným emitorem (SE), společnou bází (SB) a společným kolektorem (SC).



Obr. 1.30: Základní typy zapojení tranzistorových zesilovačů

Nejčastěji se využívá zapojení se společným emitorem (SE), které má velké proudové, napěťové i výkonové zesílení. Zapojení se společným kolektorem (SK) nalezneme u zesilovačů, kde je požadován nízký výstupní odpor. Zapojení se společnou bází (SB) se pro svůj malý vstupní odpor používalo ve vstupních obvodech vysokofrekvenčních zesilovačů.

Přehled vlastností jednotlivých zapojení			
veličina	SE	SB	SC
vstupní odpor (R _{vst})	malý až střední	malý	velmi velký
výstupní odpor (R _{výst})	velký	velmi velký	malý
napěťové zesílení	velké	velké	menší než 1
proudové zesílení	velké	menší než 1	velké
výkonové zesílení	velké	malé až střední	malé až střední

Tab. 1.

1.5.1 Vlastnosti zesilovače

Zesílení je bezrozměrnou jednotkou a udává, kolikrát se změní příslušný parametr (napětí, proud, výkon).

Napěťové zesílení:
$$A_u = \frac{\Delta U_2}{\Delta U_1}$$

Proudové zesílení:
$$A_I = \frac{\Delta I_2}{\Delta I_1}$$

Výkonové zesílení:
$$A_p = \frac{\Delta P_2}{\Delta P_1}$$

Zesílení se vyjadřuje v dB (decibelech).

Napětíové a proudové zesílení vyjádříme v decibelech takto: $A_{(dB)} = 20\log A_u (A_i)$

Výkonové zesílení vyjádříme v decibelech takto: $A_{(dB)} = 10\log A_P$

Šířka pásma je rozsah kmitočtů, který je zesilovač schopen přenést. V této souvislosti hovoříme o dolním a horním mezním kmitočtu, který odpovídá poklesu zesílení o 3 dB.

Nelineární zkreslení je způsobeno nelinearitou aktivních součástek a způsobuje tvarovou změnu výstupního signálu oproti vstupnímu.

Stabilita je odolnost zesilovače proti rozkmitání. Zesilovač může kmitat na velmi vysokých kmitočtech.

Základní rozdělení

Zesilovače napětí, proudu a výkonu, nízkofrekvenční, vysokofrekvenční a širokopásmové, dále podle nastavení pracovního bodu a principu činnosti do tříd A, AB, B, C, D, G, H, S a T nebo na předzesilovače a výkonové (koncové) zesilovače.

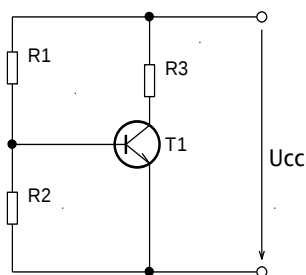
Nízkofrekvenční zesilovače slouží k zesilování slyšitelného pásma kmitočtů, v rozsahu 20 Hz až 20 kHz. Vysokofrekvenční zesilovače se používají pro zesílení kmitočtů vyšších než 20 kHz. Používají se v rozhlasových a televizních přijímačích, vysílačích nebo mobilních telefonech. Širokopásmové zesilovače se používají tam, kde je třeba pracovat se širokým spektrem kmitočtů.

Zesilovače, pracující ve třídě A, mají pracovní bod nastaven na střed lineární části převodní charakteristiky, mohou tedy s minimálním zkreslením zpracovat kladná i záporná napětí, a proto se používají k zesílení střídavého napětí. Výkonové nízkofrekvenční zesilovače pracují ve třídě AB, s malým klidovým proudem, nebo B s nulovým klidovým proudem. Výhodou je zvýšení účinnosti. Aby bylo možno zesilovat obě půlvlny střídavého napětí, používá se komplementárních tranzistorů (viz. dále). V současné době se převážně setkáváte se specializovanými integrovanými obvody a zesilovače s diskrétními součástkami se používají jen v jednoduchých aplikacích.

1.5.2 Nastavení a stabilizace pracovního bodu tranzistoru v zesilovači

Poloha klidového pracovního bodu tranzistoru je neměnná pokud je teplota stálá to však není proto musíme pracovní bod nastavit vhodnou hodnotou stejnosměrného proudu do báze

I_B . Tento proud ovlivňuje proud kolektoru I_C a tím i úbytek napětí na kolektorovém rezistoru a napětí na kolektoru U_{CE} . Abychom dosáhli co největší amplitudy výstupního napětí, mělo by U_{CE} být přibližně polovinou napájecího napětí. Následující obrázek ukazuje nejjednodušší možnosti nastavení pracovního bodu.



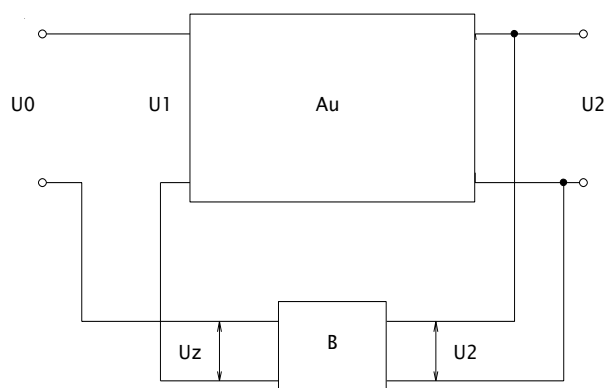
Obr. 1.31: Stabilizace pracovního bodu tranzistoru

Pro nastavení pracovního bodu použito děliče napětí, jehož příčný proud je několikanásobně větší než proud báze (tvrdý dělič napětí). Napětí na bázi je stabilní a nekolísá. Nevýhodou takového řešení je, že rezistor R_2 , který je připojen paralelně ke vstupu, snižuje vstupní odpor zesilovače a zatěžuje zdroj vstupního signálu. Měkký dělič napětí s většími hodnotami R_1 a R_2 volíme u takového zapojení, kde prochází malý I_C a v kolektoru je velká hodnota rezistoru R_3 .

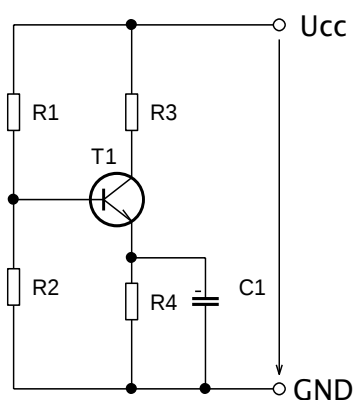
1.5.3 Zpětné vazby

Zpětná vazba vznikne přivedením části zesíleného napětí nebo proudu na vstup zesilovače. Jsou-li tato napětí ve fázi, hovoříme o kladné zpětné vazbě, která zvyšuje zesílení i zkreslení, snižuje vstupní odpor a může způsobit i rozkmitání zesilovače (na výstupu se objeví napětí bez ohledu na přítomnost vstupního signálu). Tento jev je považován za nežádoucí, proto se kladná zpětná vazba v zesilovačích nepoužívá. Čím větší je zesílení, tím větší nebezpečí hrozí. Vstupní a výstupní obvody musí být od sebe co nejvíce vzdáleny, případně odděleny plochou spojenou s nulovým potenciálem (GND). Na přívody se používá stíněných vodičů, jejichž plášť je spojen s nulovým potenciálem.

Záporná zpětná vazba, kdy jsou napětí v protifázi (posunutá o 180°), sice snižuje celkové zesílení, ale příznivě se projevuje snížením zkreslení, zvýšením vstupního odporu a stability. V zesilovačích je často používána mimo jiné také k teplotní stabilizaci pracovního bodu tranzistoru.



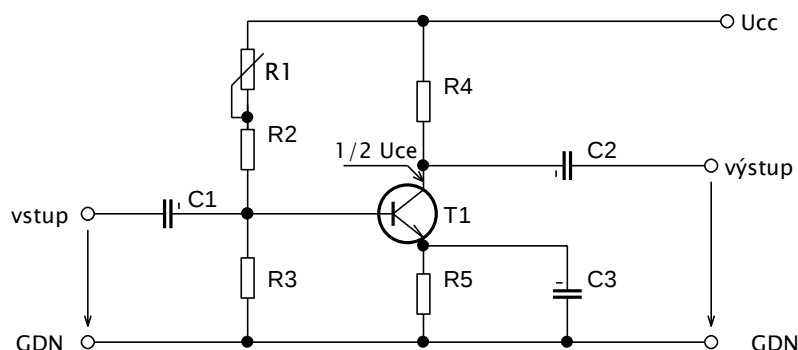
Obr. 1.32: Záporná zpětná vazba



Obr. 1.33: Zapojení záporné střídavé zpětné vazby C_1

V zesilovač se záporná zpětná vazba realizuje pouze pro střídavý signál, případně pro určité pásmo kmitočtů pomocí C_1 . Jeho kapacita je výrazně větší (50 až 500 μF). V přenášeném kmitočtovém pásmu se tedy záporná zpětná vazba neprojevuje a R_4 plní funkci stejnosměrné stabilizace pracovního bodu.

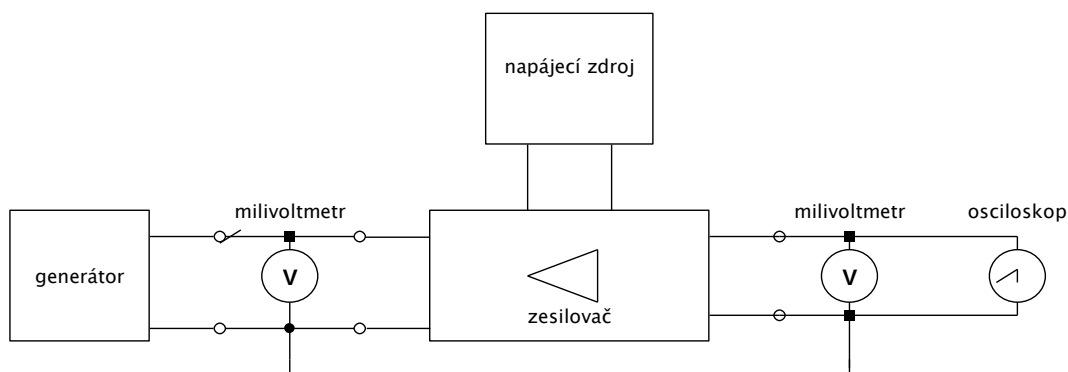
1.5.4 Jednostupňový střídavý zesilovač



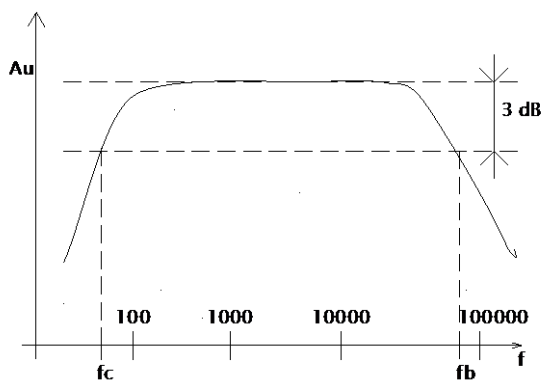
Obr. 1.34: Jednostupňový střídavý zesilovač s tranzistorem NPN v zapojení SE

Na obrázku je zapojení jednostupňového střídavého zesilovače ve třídě A v zapojení se společným emitorem. Klidový pracovní bod je nastaven odporovým trimrem R1. Ochranný rezistor R2 omezuje maximální hodnotu proudu I_B a zabraňuje překročení proudu I_B tranzistoru. Abychom dosáhli co největší amplitudy výstupního napětí, nastavíme trimrem napětí v kolektoru na $\frac{1}{2} U_{cc}$. Kondenzátory C1 a C2 se nazývají vazební a oddělují stejnosměrné napětí v bázi, respektive v kolektoru tranzistoru. Protože kondenzátory „propouštějí“ pouze střídavý proud, nenaměříme na vstupu ani výstupu žádné stejnosměrné napětí. Rezistor R4 nazýváme pracovní a vlivem změn I_C v rytmu vstupního signálu se na něm vytváří úbytek napětí. R5 společně s C3 stabilizují pracovní bod tranzistoru.

Připojíme-li na vstup zesilovače generátor sinusového signálu, na výstup osciloskop a jemně otáčíme běžcem trimru R1, můžeme na obrazovce sledovat změny výstupního signálu. Správné nastavení je takové, kdy jsou obě půlvlny zesíleny bez viditelného zkreslení.



Obr. 1.35: Schéma měření zesílení zesilovače



Obr. 1.36: Frekvenční charakteristika nízkofrekvenčního zesilovače

Chceme-li dosáhnout vyššího zesílení, než je schopen poskytnout jednostupňový zesilovač, musíme použít více tranzistorů. Nejjednodušší by bylo zapojit dva jednostupňové zesilovače za sebou (výstup prvního se spojí se vstupem druhého). Celkové zesílení je pak násobkem zesílení jednotlivých stupňů. V praxi se však častěji používá stejnosměrně vázaný dvou nebo více stupňový zesilovač.

1.5.5 Korekční zesilovače

Korekční zesilovače se používají k úpravě kmitočtové charakteristiky. Korekce jsou buď nastaveny pevně a zesilovače vyrovnávají kmitočtové charakteristiky, nebo mají potenciometry a umožňují odděleně nastavit zesílení nízkých a vysokých kmitočtů.

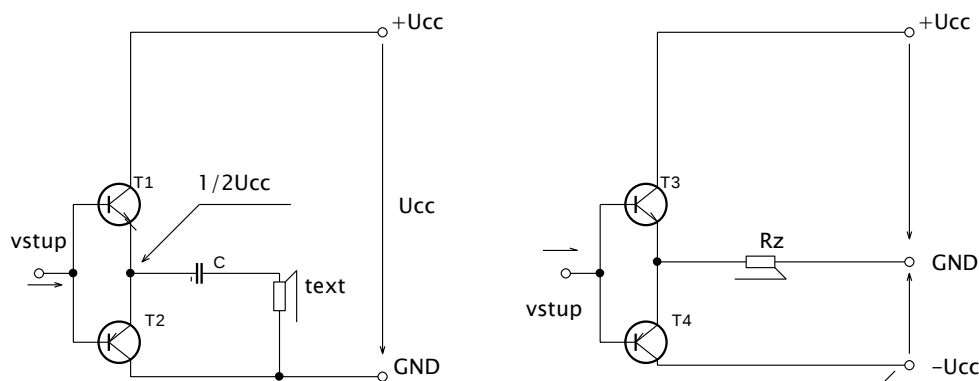
V korekčních zesilovačích se dnes používá speciálních integrovaných obvodů.

1.5.6 Výkonový zesilovač s komplementárními tranzistory

Výkonové (koncové) zesilovače musí dodat dostatečný výkon pro reproduktory, které elektrický signál přemění na akustický. Impedance reproduktorů bývá 4 až 16 Ω , proto zesilovač musí dodat poměrně velký výstupní proud.

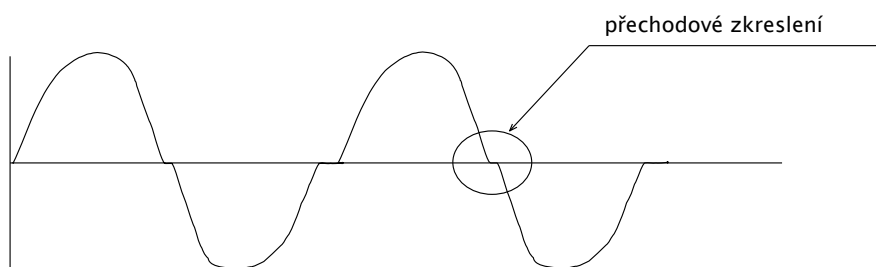
$$\text{Výstupní výkon zesilovače } P = \frac{U_{\text{výst.}}^2}{R_z}$$

Koncové tranzistory bývají komplementární, tzn. typu NPN a PNP. Využívá se toho, že vzhledem ke své opačné vodivosti reagují na přivedený signál odlišně. Při kladné půlvlně se tranzistor typu NPN otevírá, zatím co tranzistor typu PNP se zavírá. Amplituda výstupního napětí tedy může dosáhnout téměř hodnoty napájecího napětí



Obr. 1.37: Základní zapojení komplementárních koncových stupňů

Pracují-li tranzistory s nulovým klidovým proudem I_C (třída B), jako na předchozích obrázcích, objevuje se ve výstupním signálu tzv. přechodové zkreslení. Proto se častěji používá zapojení ve třídě AB, kdy v klidu prochází koncovými tranzistory malý klidový proud (cca 5 až 20 mA).



Obr. 1.38: Přechodové zkreslení koncového stupně ve třídě B

1.5.7 Integrované obvody pro koncové zesilovače

V současnosti jsou koncové stupně řešeny celou řadou integrovaných obvodů, určených pro koncové zesilovače. Liší také svými parametry a oblastí použití. Ke své činnosti vyžadují minimum dalších součástek, a proto již podstatě vytlačily „klasická“ zapojení s diskrétními součástkami.

1.5.8 Použití unipolárních tranzistorů v zesilovačích

Unipolární tranzistory mají pouze jeden polovodičový přechod, který tvoří tzv. kanál mezi emitorem a kolektorem. Velikost proudu je závislá pouze na napětí řídicí elektrody, kterým se mění vodivost kanálu. Na rozdíl od bipolárních tranzistorů protéká proud kolektorem i při nulovém napětí na řídicí elektrodě. Jeho velikost se v závislosti na U_G pouze zvětšuje nebo zmenšuje.

U tranzistorů J-FET je řídicí elektroda oddělena polovodičovým přechodem, který je vždy polován v závěrném směru a neprochází jím tedy téměř žádný proud. Technologie MOS-FET využívá kovové řídicí elektrody, oddělené slabou izolační vrstvou oxidu křemičitého (SiO_2). V obou případech hovoříme o tranzistorech řízených elektrickým polem a jejich vstupní odpor je velmi vysoký. To s sebou však přináší riziko zničení tranzistoru elektrostatickým polem.

Tranzistor JFET Tranzistor MOSFET



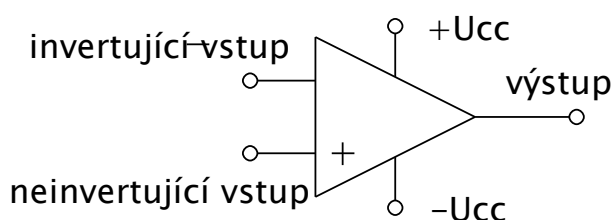
Obr. 1.39: Polem řízené tranzistory

1.6 Operační zesilovače

Základní informace

Název operační zesilovače (OZ) pochází z původní funkce, tedy provádění různých matematických operací v analogových počítačích. Od té doby se konstrukce OZ velmi změnila, ale základní požadavky zůstávají stejné. Ideální OZ by měl mít následující parametry: nekonečně velké zesílení, nekonečně velkou vstupní impedanci, nulovou výstupní impedanci a nekonečně vysokou rychlost přeběhu. V praxi nelze samozřejmě těchto parametrů dosáhnout, proto se můžeme setkat např. s nízkošumovými OZ, které se používají ve vstupních obvodech, vysokofrekvenčními OZ nebo výkonovými OZ. V jednom pouzdře mohou být až čtyři.

Protože jedním ze základních požadavků je zpracování stejnosměrných signálů, musí být OZ napájeny symetrickým napájecím napětím (např. $\pm 15\text{V}$).



Obr. 1.40: Vývodové zapojení operačního zesilovače

Každý OZ má dva vstupy invertující (-) a neinvertující (+), jeden výstup, vývody pro kladné ($+U_{cc}$) a záporné ($-U_{cc}$) napájecí napětí a případně další pomocné vývody pro nastavení

symetrie nebo pro kmitočtovou kompenzaci. Číslování i zapojení vývodů se u jednotlivých typů OZ může odlišovat. Konkrétní údaje jsou uvedeny v katalogových listech.

1.6.1 Vlastnosti

Při použití OZ musíme vzít v úvahu reálné vlastnosti, které nemohou dosahovat výše uvedených hodnot ideálního OZ.

1.6.1.1 Vstupní napěťová nesymetrie

Protože vstupní obvody nejsou nikdy naprosto stejné, objevuje se na výstupu napětí, i když je na obou vstupech napětí shodné. Kromě toho dochází ke změnám vstupní napěťové nesymetrie i v závislosti na teplotě

1.6.1.2 Vstupní klidový proud

Je způsoben nestejně velkými vstupními proudy. Na výstupu se projevuje nenulovým napětím a je teplotně závislý, podobně jako v předchozím případě.

1.6.1.2 Kmitočtová kompenzace

Při zesilování střídavého napětí se směrem k vyšším kmitočtům snižuje zesílení a mění fáze signálu. To bývá příčinou nestability. Proto se zavádí kmitočtová kompenzace. Některé OZ mají kmitočtovou kompenzaci vnitřní.

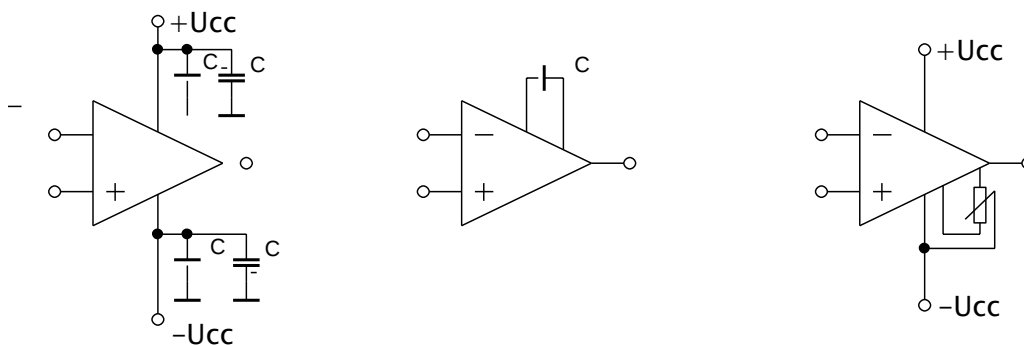
1.6.1.3 Rychlost přeběhu

Rychlost přeběhu znamená rychlost změny výstupního napětí, na skokovou změnu vstupního napětí. Udává se ve voltech za mikrosekundu ($V/\mu s$)

1.6.1.4 Obvodové požadavky

Napájecí napětí (kladné i záporné) je vhodné blokovat kondenzátory kapacity 100 nF, případně i elektrolytickými kondenzátory kapacity 10 až 100 μF . Některé typy OZ umožňují pomocí vnějších pasivních součástek kompenzovat napěťovou a proudovou nesymetrii.

Pokud OZ neobsahuje prvky kmitočtové kompenzace uvnitř pouzdra, je třeba připojit rezistory a kondenzátory.



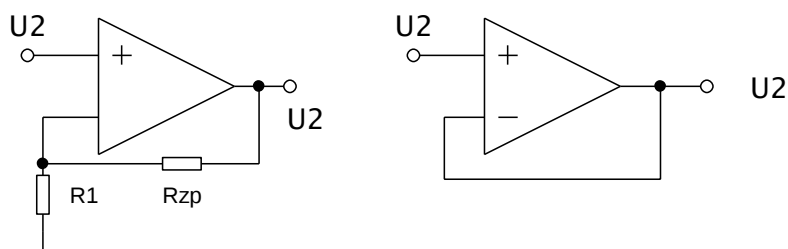
Obr. 1.41: Jednotlivé obvodové řešení vnějších prvků OZ – blokování napájecího napětí, kmitočtová kompenzace, kompenzace vstupní nesymetrie

1.6.2 Neinvertující zesilovač

Vstupní napětí se přivádí do neinvertujícího vstupu (+) a invertující vstup (-) je připojen k odporovému děliči složenému z rezistorů R1 a Rzp. U neinvertujícího zesilovače je vstupní signál ve fázi s výstupním a zesílení je vždy větší než jedna.

$$U_2 = U_1 \cdot \left(1 + \frac{R_{zp}}{R1}\right) \quad A_v = 1 + \frac{R_{zp}}{R1}$$

Významnou vlastností tohoto zapojení je vysoký vstupní odpor (nedochází k zatěžování zdroje signálu připojeného na vstup).

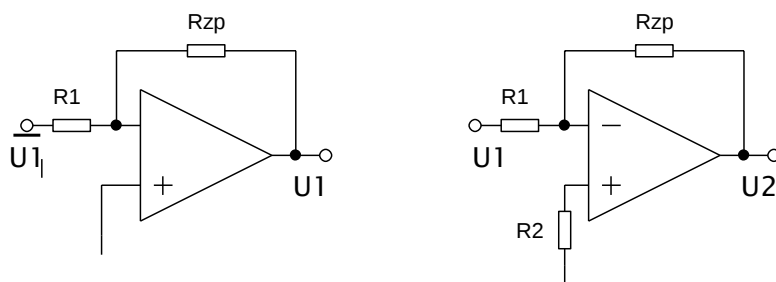


Obr. 1.42: Zapojení neinvertujícího zesilovače

Krajní variantou je zapojení bez R1 i Rzp, kde je invertující vstup přímo spojen s výstupem. Má zesílení 1 a používá se jako impedanční převodník.

1.6.3 Invertující zesilovač

Vstup je připojen přes R1 k invertujícímu vstupu, zatímco druhý vstup je spojen s nulovým potenciálem. K výstupu je připojena zátěž Rz a zpětnovazební rezistor Rzp, který určuje velikost záporné zpětné vazby. Vstupní impedance závisí na velikosti rezistoru R1.



Obr. 1.43: Zapojení invertujícího zesilovače

Pro jednotlivé proudy platí vztahy:

Kompensace vstupní proudové nesymetrie pomocí rezistoru R2, jehož velikost je rovná paralelní kombinaci R1 a Rzp:

$$R_2 = \frac{R_1 \cdot R_{zp}}{R_1 + R_{zp}}$$

Výpočet zesílení:

$$U_2 = -\frac{R_{zp}}{R_1} \cdot U_1$$

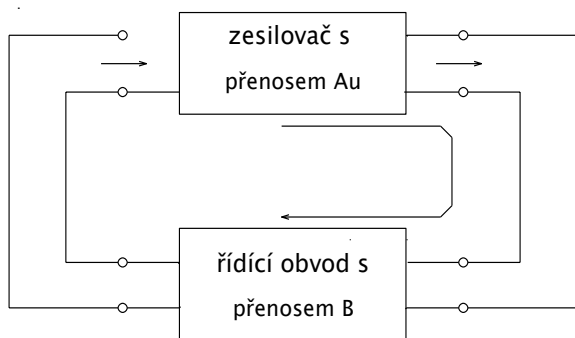
$$A_U = -\frac{R_{zp}}{R_1}$$

1.7. Oscilátory

1.7.1 Nízkofrekvenční oscilátory

Oscilátor je zdroj střídavého napětí jehož frekvence je určena použitými součástkami. Nezpracovává žádný signál, ale je sám zdrojem signálu na rozdíl od zesilovače je oscilátor dvojpól

Základem je zesilovač, z jehož výstupu je zavedena na vstup kladná zpětná vazba. Praktické užití t oscilátorů je rozmanité v závislosti na použitých obvodových prvcích. Díky tomu je možné realizovat zpětnovazební obvody s přesně určenými vlastnostmi.



Obr. 1.44: Blokové zapojení zpětnovazebního oscilátoru

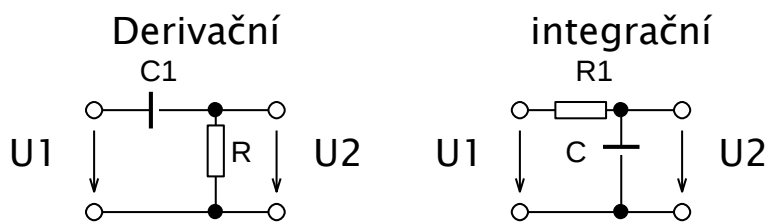
1.7.1.1 Dělení nízkofrekvenčních oscilátorů

- podle tvaru kmitů na sinusové a nesinusové
- podle prvků určujících kmitočet na RC, LC, krystalové a další
- podle druhu aktivní části oscilátoru na tranzistorové a další
- podle přeladitelnosti na pevné a přeladitelné

Oscilátory RC obsahují zesilovač a členy RC, určující kmitočet a potřebný fázový posuv ve zpětnovazební větvi. Oscilátory LC obsahují zesilovač a rezonanční obvod LC, kterým je určen kmitočet. V sinusových oscilátorech je zavedena slabá kladná zpětná vazba, která předává energii od zesilovače k pasivním členům. Silná kladná zpětná vazba v obvodu oscilátoru je podmínkou pro vznik nesinusového signálu. Příčina deformace sinusového průběhu je dána přítomností vyšších harmonických kmitočtů. Amplituda těchto složek bývá nesterajně velká.

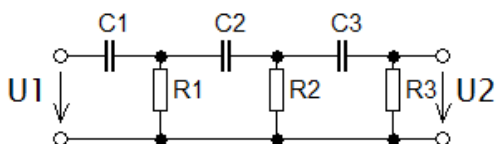
1.7.1.2 Nízkofrekvenční oscilátory RC

V oscilátoru dochází k oscilacím při splnění amplitudové i fázové podmínky. Tranzistor, v zapojení se společným emitorem, posouvá fázi signálu o 180° , takže ke splnění fázové podmínky oscilací musí být fázový posuv smyčky kladné zpětné vazby také 180° , aby celkový výsledný posuv byl 360° . Jednotlivý článek posouvá fázi o 60° , proto je nutné, aby pro posuv o 180° byl použit řetězec složený minimálně ze tří fázovacích článků.



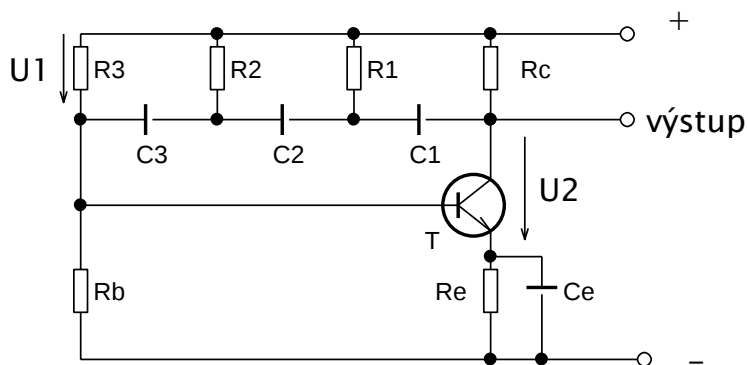
Obr. 1.45: Články používané v oscilátorech

Z řetězce derivačních článků vznikne horní propust a z řetězce integračních článků vznikne dolní propust. Používá se tříčlánková horní i dolní propust. Pro splnění amplitudové části oscilační podmínky musí být útlum signálu v pasivní části oscilátoru vyrovnán zesílením v aktivní části. Pro oscilace musí být celkový přenos $A_u \beta > 1$.



Obr. 1.46: Zapojení horní propusti

Praktické užití je převážně pro oscilátory s pevnou frekvencí. Příklad jednoduchého tříčlánkového oscilátoru RC s horní propustí vidíme obrázku. Řídící obvod tvoří tři derivační články z nichž každý posunuje fázi o 60° celkem o 180° .



Obr. 1.47: RC oscilátor s derivačními články ve zpětné vazbě

1.7.2 Vysokofrekvenční oscilátory

Vysokofrekvenční oscilátory slouží jako zdroje nosných vln různých vysílačů a generátorů. Jedná se převážně o LC oscilátory. Základem tohoto typu oscilátoru je zesilovač, z jehož výstupu je zavedena na vstup kladná zpětná vazba.

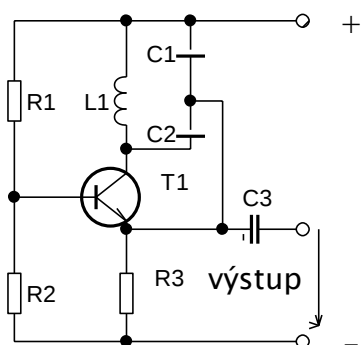
1.7.2.1 Vznik oscilací

Vysokofrekvenční oscilátory využívají klasický rezonanční obvod a jejich kmitočet je určen Thomsonovým vztahem:

$$f = \frac{1}{2\pi\sqrt{LC}}$$

1.7.2.2 Colpittsův oscilátor

Zpětná vazba u Colpittsova oscilátoru je odvozována z paralelního rezonančního obvodu s uzemněnou kapacitní odbočkou. Použitý tranzistor v zapojení se společným emitorem obrací fázi procházejícího signálu. Zpětnovazební napětí, přiváděné na vstup tranzistoru z jednoho konce rezonančního obvodu L-C1-C2 s uzemněnou kapacitní odbočkou, je však rovněž v protifázi s napětím na druhém konci tohoto obvodu, tj. na výstupu z tranzistoru. Zpětná vazba je tedy kladná a při dostatečném zesílení tranzistoru se obvod rozkmitá.



Obr. 1.48: Zapojení Colpittsova oscilátoru

Výše uvedené zapojení označujeme jako trojbodové (ta je u vf oscilátorů často používána).

Tyto oscilátory se vyznačují poměrně malou stabilitou kmitočtu (řádu 10^{-3} až 10^{-4} za 1 hodinu). Podstatně lepší jsou oscilátory stabilizované krystalovým rezonátorem.

1.7.2.3 Parametry vysokofrekvenčních oscilátorů

Stabilita kmitočtu

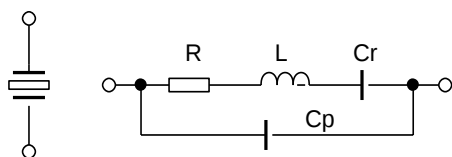
Stabilita kmitočtu je míra jeho změny s časem. Číselně se vyjadřuje poměrem maximální změny kmitočtu Δf proti jmenovité hodnotě kmitočtu f , uvažované za určitý časový interval Δt , například za 1 sekundu, 1 hodinu, 1 den atd.

Přesnost kmitočtu

Přesnost kmitočtu se vyjadřuje poměrnou odchylkou $\Delta f / f$ průměrné hodnoty kmitočtu f , uvažovanou opět za určitý časový interval Δt od referenční hodnoty f .

1.7.3 Krystalové oscilátory

Požadavky na vysokou stabilitu kmitočtu oscilátoru splňují oscilátory řízené krystalem. Používají se pro kmitočtové standardy jsou využívány v čítačích kmitočtu, v digitálních systémech, mikroprocesorových zařízeních, časových etalonech, a při měření fyzikálních veličin. Existuje velké množství druhů krystalových oscilátorů (KO)



Obr. 1.49: Náhradní schéma krystalu

1.7.3.1 Rozdělení podle kmitočtového rozsahu

Nízkofrekvenční (10 kHz až 1000 kHz), vysokofrekvenční (1 MHz a výše)

1.7.3.2 Rozdělení podle stability kmitočtu

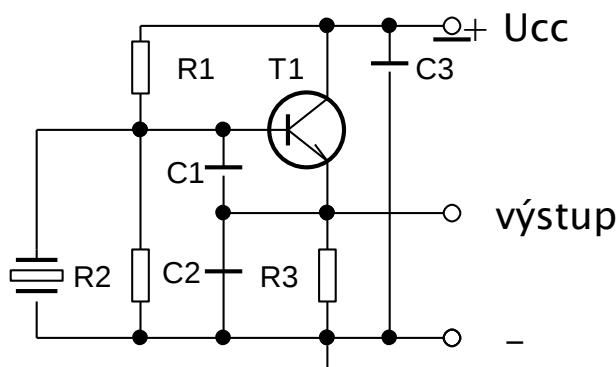
Jedná se o dlouhodobou stabilitu kmitočtu. Velmi stabilní, $\Delta f / f < 1 \cdot 10^{-9} / \text{den}$, se zlepšenou stabilitou, $\Delta f / f < \pm 1 \cdot 10^{-7} / \text{rok}$ standardní, $\Delta f / f < \pm (1-2) \cdot 10^{-6} / \text{rok}$.

1.7.3.3 Hlavní součásti krystalových oscilátorů

Krystalový oscilátor je obvod, ve kterém je použita krystalová jednotka jako velmi jakostní selektivní člen. Zapojení využívá piezoelektrických vlastností výbrusu krystalu křemene. Oscilátory, v nichž je krystalová jednotka, která nahrazuje LC rezonanční obvod. Tato zapojení nazýváme třibodová a krystal v nich kmitá na kmitočtu mezi sériovou a paralelní rezonancí. Charakter jeho impedance je vždy indukční.

1.7.3.4 Oscilátor Clappův

Příklad zapojení krystalového oscilátoru je na obrázku. Střídavě uzemněný je kolektor a krystalová jednotka kmitá mezi sériovou a paralelní rezonancí podle velikosti zatěžovací kapacity. U Clappova oscilátoru je více zatížena krystalová jednotka bázovým děličem. Kapacita kondenzátorů C1 a C2 se volí opět pokud možno co největší, aby se potlačil vliv parazitních reaktancí tranzistoru.



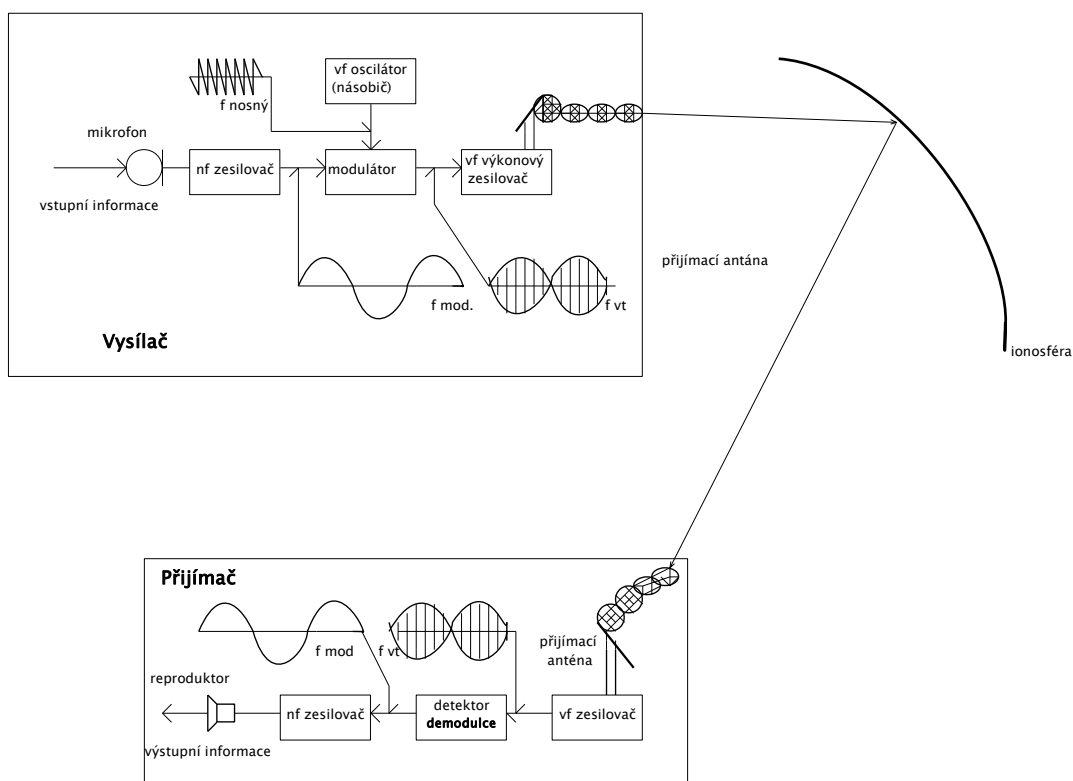
Obr.1.50: Zapojení Clappova krystalového oscilátoru

1.8 Modulace

Vysílač vysílá do prostoru elektromagnetické vlny, které se šíří prostorem rychlostí 300 000 km za sekundu, kde je zachytí anténa rádiového nebo televizního přijímače.

Samotné elektromagnetické vlny potřebnou informaci nepřenášejí. Pomocí přijímače bychom mohli nejvýše zjistit, že vysílač elektromagnetické vlny buď vysílá nebo nevysílá. Samostatné elektromagnetické vlny totiž představují pouze jakousi spojovací linku mezi vysílačem a přijímačem. Stejně jako drátovým vedením, tak i elektromagnetickým vlněním se zpráva, informace (řeč, hudba, obraz) musí teprve poslat. Informaci je nutné na vysílané elektromagnetické vlně určitým způsobem vtisknout - tomu se říká modulace.

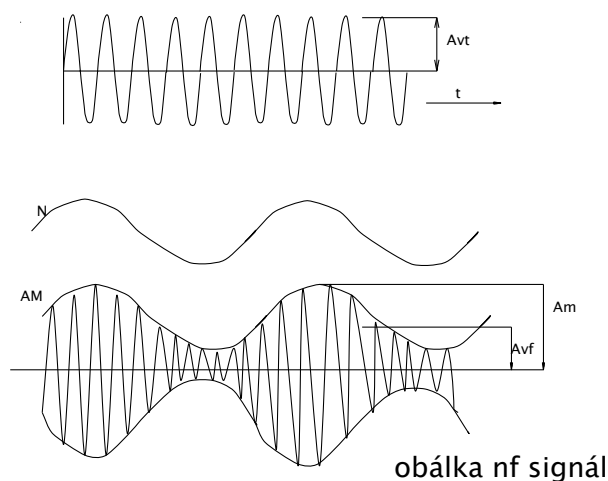
Modulace je základem bezdrátového přenosu informací. Dělíme na dvě základní a to na amplitudovou a frekvenční modulaci.



Obr. 1.51: Princip přenosu informací pomocí elektromagnetických vln

1.8.1 Amplitudová modulace

Základní princip amplitudové modulace je celkem jednoduchý: amplituda nosné vlny se mění podle okamžité hodnoty amplitudy modulující vlny (obrázek a). Křivka N znázorňuje časový průběh nízkofrekvenční modulující vlny. Křivka vf znázorňuje průběh vysokofrekvenční nosné vlny. Časový průběh modulované vlny je znázorněn křivkou AM. Čárkované čáry představují obalovou křivku signálu. Amplituda původní vysokofrekvenční vlny se tedy zmenšuje a zvětšuje v porovnání se svou původní hodnotou přesně v rytmu modulující vlny.

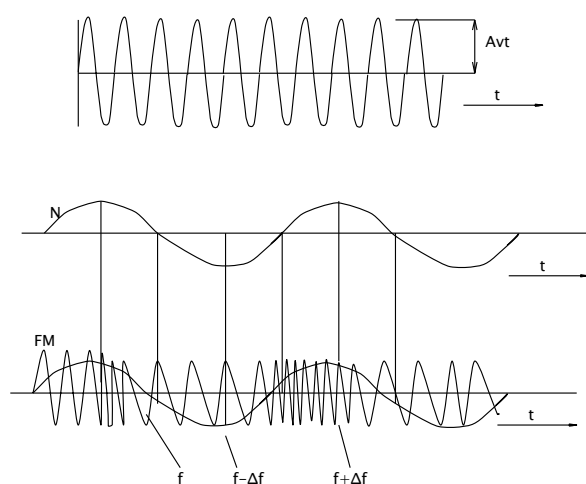


Obr. 1.52: Amplitudově modulovaná nosná vlna

Přenášená informace (v našem případě sinusový průběh s kmitočtem $n f$) je v modulované vlně obsažena v podobě pomyslné obálky naznačené na obrázku. Při amplitudové modulaci je důležitý údaj o hloubce amplitudové modulace α . Modulované vlny mohou mít různou hloubku modulace 25%, 75%, 100%. Je-li hloubka modulace větší než 100% dochází ke zkreslení signálu a proto např. v rozhlasovém a televizním vysílání nelze hloubku modulace nad 100% povolit.

1.8.2 Frekvenční modulace

Jestliže budeme modulující vlnou působit místo na amplitudové nosné vlny na její kmitočet (frekvenci), dostaneme kmitočtově neboli frekvenčně modulovanou vlnu. Kmitočet nosné vlny se mění přesně podle průběhu modulující vlny (obrázek).



Obr. 1.53: Frekvenčně modulovaná nosná vlna

Při frekvenční modulaci se však dosahuje kvalitnějšího přenosu. Pro velkou potřebnou šířku pásma nemohou frekvenčně modulované vysílače vysílat na dlouhých, středních ani krátkých vlnách. Vysílají v pásmu velmi krátkých vln, kde je místa tolik, že se frekvenční rozsah modulujícího signálu nemusí omezovat. Vysílá se celé nízkofrekvenční spektrum akustického signálu.

Pro velmi krátké vlny (rozhlas) kmitočtová modulace 50 kHz ve špičce. Pro neskreslený přenos zapotřebí, aby byla šířka pásma 200 kHz pro mono a stereo 300 kHz

1.9 Optoelektronika

Optoelektronika je oblast elektroniky která se zabývá interakci světla s elektronickými prvky.

Optoelektronické prvky umožňují přeměnu elektrické energie na světelnou nebo naopak. Přenos informací pomocí světelné energie, to je elektromagnetických vln ležících v optické části spektra, umožňuje aplikace optoelektronických zařízení v mnoha dalších oborech – automatizace průmyslu, počítače, medicína, optická sdělovací technika využívající optoelektronických prvků a vláknové optiky přenosu velkého množství informací.

Základní prvky optoelektroniky tvoří polovodičové zdroje záření (elektroluminiscenční dioda, laserová dioda), které umožňují přímou konverzi elektrické energie na optickou, a detektory optického záření. Detektory převádějí energii optického záření na jinou měřitelnou veličinu.

1.9.1 Optoelektronika – prvky

Optoelektronické zdroje světla: LED diody, polovodičový laser, displeje LED a LCD, osvětlení a signalizace.

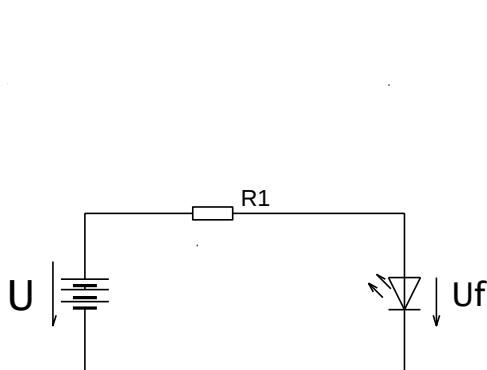
Optoelektronické senzory: fotorezistory, infradiody, fototranzistory.

Optické spínací prvky: optočleny, optotriaky a další prvky a jejich kombinace pro různé aplikace v elektronice.

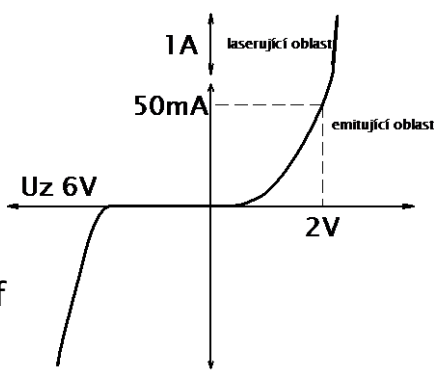
Přenos informací: optická vlákna případně přenos vzduchem.

1.9.1.1 LED dioda

LED (Light Emitting Diode = světlo emitující dioda) dioda přeměňuje elektrickou energii na elektrickou a využívá se hlavně k signalizaci. Na rozdíl od žárovky má delší životnost, ale "vysílá" jen barevné světlo a je "slabší" než svit žárovky a při provozu postupně klesá. Vyrobí se v různých provedeních a v různých barvách – červená, žlutá, bílá, modrá jak o základní materiál se používá GaAs. Led dioda emituje světlo v propustném směru a při U_f 1,6V až 4V toto je dáno barvou emitovaného světla



Obr. 1.54: Zapojení LED diody

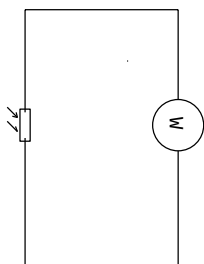


Obr. 1.55: VA charakteristika LED

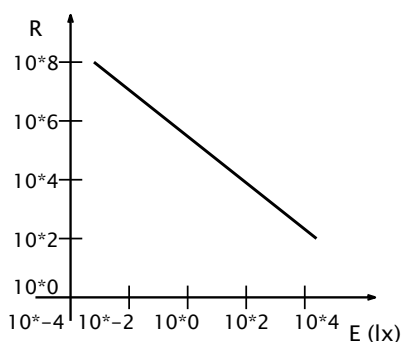
1.9.1.2 Fotorezistor

Jsou to polovodičové součástky, které mění svůj odpor s osvětlením. Ke své činnosti využívají fotoefekt, což je vlastnost některých látek např. selenu, že při osvětlení uvolňují elektrony, které neemitují, ale zůstávají uvnitř hmoty. Čím více světla, tím více uvolněných elektronů a menší odpor součástky.

Přibližně má fotorezistor ve tmě odpor $1 M\Omega$ a na světle $1 k\Omega$



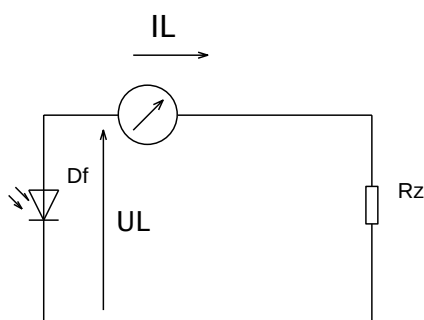
Obr. 1.56: Zapojení fotorezistoru



Obr. 1.57: Jeho charakteristika

1.9.1.3 Fotodioda

Využívá fotoelektrický jev k přeměně světelné energie na elektrickou. Základem je polovodičový přechod, na který dopadá světlo. V obou polovodičích se dopadem fotonů uvolňují dvojice díra-elektron a proudí přechodem. Náboje se na obou stranách hromadí, čímž vzniká rozdíl potenciálů - fotoelektrické napětí U_L (elektromotorické napětí). Na anodě se nachází kladný potenciál, na katodě záporný potenciál. Fotoelektrické napětí dosahuje až $U_L = 0.5 \text{ V}$. Připojíme-li k vývodům zátěž R_z , obvodem bude procházet proud.

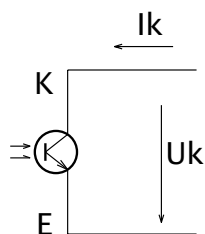


Obr. 1.58: Zapojení pro změření vlastností fotoelektrického jevu

Fotoelektrické články větších rozměrů se propojují ve sluneční baterie, takže fungují jako alternativní zdroj energie.

1.9.1.4 Fototranzistor

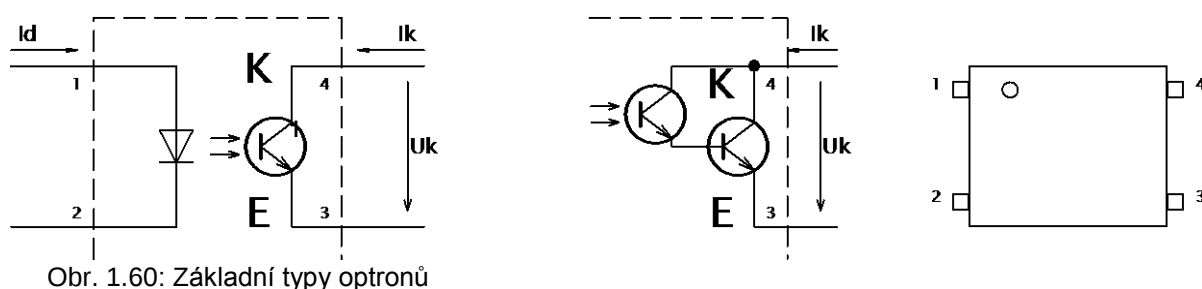
Fototranzistor je optoelektronický prvek, v němž je jako detektor použit přechod báze emitru tranzistoru. Proud vzniklý absorpcí dopadajících částic je zesílen tranzistorovým jevem. Je konstruován tak aby co nejvíce záření bylo absorbováno oblastí báze. Proudová citlivost fototranzistoru je o jeden řád vyšší než u fotodiod má však horší spínací časy.



Obr. 1.59: Fototranzistor

1.9.1.5 Optron – optoelektrický vazební člen

Optron se skládá z luminiscenční diody, která emituje světlo, a fototranzistoru nebo fotodiody, které světlo detekují. Oba prvky jsou v jednom pouzdře, jsou však elektricky izolovány. Protéká-li fotodiodou proud, vybudí světelný tok, který je snímán a zesílen fototranzistorem. Toto umožňuje optronu přenést signál mezi dvěma galvanicky oddělenými obvody. Optron se používá zejména ke galvanickému oddělení elektronických obvodů. Účinně chrání obvody proti rušení a poškození vstupu cizím napětím.



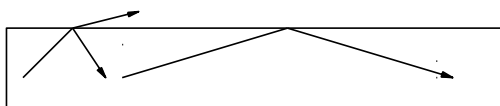
1.9.2 Optická vlákna a kabely

Základním rozdílem mezi metalickými a optickými kabely je, že u metalických kabelů jsou data přenášena za využití elektrických signálů, zatímco v optických kabelech je signál přenášen světelnými impulzy.

1.9.2.1 Optické vlákno

Přenos dat v optickém vlákne se děje šířením světla v optickém vlnovodu. Princip vedení světla je jednoduchý - světelný paprsek dopadá na rozhraní dvou prostředí s rozdílnou optickou hustotou a tedy s rozdílným indexem lomu, kde se **zčásti láme** a prostupuje z jednoho prostředí do druhého, a z **části se odráží** a vrací se zpět do prostředí, ze kterého přichází. Nakolik se paprsek odrazí zpět do prostředí, ze kterého pochází, záleží na úhlu, ve kterém paprsek do vlnovodu přichází. Pro každé optické rozhraní však existuje **mezní úhel odrazu**. Pokud světlo dopadá pod tímto (nebo menším úhlem) dochází k tzv. **totálnímu odrazu**, kdy se 100% světla odráží a neopouští prostředí, ze kterého přichází. Právě tento princip "vnitřních odrazů" využívají optická vlákna.

Odraz a lom světla



Obr. 1.61: Funkce optického vlákna

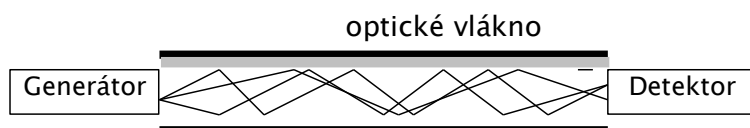
1.9.2.2 Typy optických vláken

1.9.2.3 Mnohovidová vlákna (MM, multimode)

Pro mnohovidová vlákna se používají optické kabely s průměrem jádra 50 a 62,5 μm

V současné době je typickou vlnovou délkou těchto vláken 850nm pro gigabitové aktivní prvky a 1310 nm pro 100 Mbitové aktivní prvky.

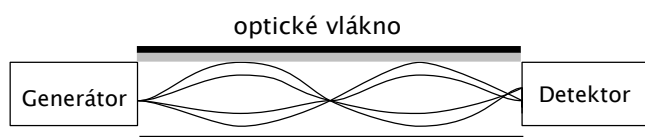
■



Obr. 1.62: Mnohovidová vlákna

Mnohovidová vlákna s plynulou změnou indexu lomu jsou tvořena z tenkých vrstev, které se liší indexem lomu. Čím je paprsek dál od osy jádra, tím je index lomu menší a postupně se paprsek "vrací zpět" do středu jádra.

■



Obr. 1.63: Funkce přenosu v mnohovidovém vlákně

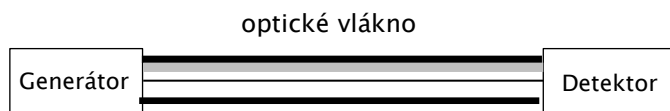
Tyto vlákna tedy netrpí vidovou disperzí a vyslané paprsky dorazí ke konci vlákna zhruba ve stejné časové okamžiky. Typický útlum těchto vláken je 2 až 10 db/km při vlnové délce 850nm. Mnohovidová vlákna se nejčastěji používají u LAN sítí. Vzhledem k tlustšímu jádru optického kabelu se lépe na vlákno připevňují konektory.

1.9.2.4 Jednovidová vlákna (SM, singlemode)

Pro jednovidová vlákna se používají optické kabely s průměrem jádra pouhých 9 μm (průměr vlasu 70 μm). Typickými vlnovými délkami jsou v současné době 1310nm a 1550nm.

Signál je tvořen **jedním videm** (paprskem), který se šíří podél osy jádra a odráží se při ohybu.

■



Obr. 1.64: Přenos signálu jednovidovým vláknem

Pro generování a detekci světla u jednovidových vláken je zapotřebí kvalitnějších a tedy i dražších zařízení. Jako generátory jsou zde používány zejména polovodičové lasery.

1.9.2.5 Opakovací otázky:

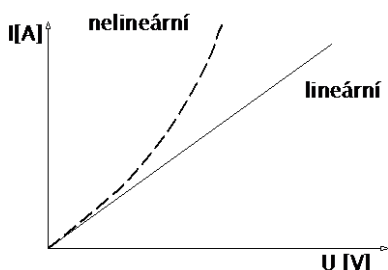
1. Jaký je rozdíl mezi lineárními a nelineárními prvky?
2. Popište funkci polovodičové diody.
3. Popište bipolární tranzistor.
4. Popište činnost můstkového usměrňovače.
5. K čemu slouží filtrační obvody ve zdroji?
6. Popište a nakreslete stabilizátor pevného kladného napětí.
7. Popište princip transformátoru.
8. Popište princip spínaného zdroje.
9. Popište jednostupňový nízkofrekvenční zesilovač ve třídě A. Uveďte základní třídy nízkofrekvenčních zesilovačů.
10. Jaké jsou základní parametry operačního zesilovače.
11. Popište nízkofrekvenční oscilátor RC a uveďte podmínky kmitání.
12. Jaké jsou základní optoelektronické prvky.

1.9.2.6 Odpovědi na opakovací otázky:

1. Jaký je rozdíl mezi lineárními a nelineárními prvky?

Lineární prvky mají voltampérovou charakteristiku přímkovou.

Nelineární prvky mají V-A charakteristiku zakřivenou (např. dioda).



2. Popište funkci polovodičová diody.

Polovodičová dioda je polovodičová součástka s jedním přechodem PN, která propouští proud pouze v jednom směru. Aby vůbec došlo k otevření diody, musíme nejprve překonat tzv. prahové napětí (asi 0,6V). Zbytkový proud v závěrném směru lze většinou zanedbat. S rostoucí teplotou závěrný proud narůstá. Reálnou diodu lze napětově namáhat jen do určitého napětí a potom může nastat průraz (zda bude destruktivní nebo nedestruktivní, záleží na velikosti výkonu, který se na přechodu změní v teplo – především tedy na velikosti procházejícího proudu diodou).

3. Popište bipolární tranzistor.

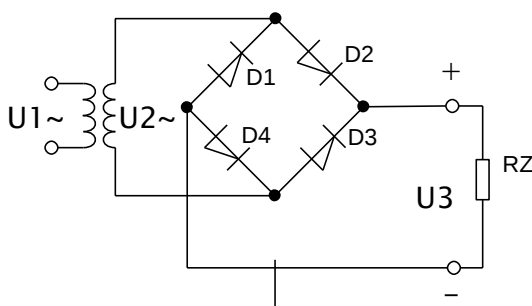
Bipolární tranzistor je polovodičová součástka se dvěma přechody PN umožňující malým proudem báze řídit mnohem větší proud mezi kolektorem a emitorem. Tranzistor může pracovat jako zesilovač i jako spínač. Jako základní materiál je použit většinou křemík. Mohou být typu NPN nebo PNP. Tranzistor má tři elektrody, které se nazývají báze (B), emitor (E) a kolektor (C). Proudy označujeme písmenem I a příslušným indexem I_B , I_C a I_E . Emitorem protéká proud báze i kolektoru ($I_E = I_B + I_C$). Napětí označujeme písmenem U s označením elektrod, uvádějícím elektrody tranzistoru U_{BE} , U_{CB} , U_{CE} . Přechod mezi bází a emitorem je polován v propustném směru a jeho napětí je menší než 1 V ($U_{BE} \approx 0,7$ V).

4. Jak pracuje můstkový usměrňovač?

Můstkový usměrňovač umožňuje usměrnění obou půlvln střídavého proudu transformátoru. V první půlperiodě střídavého napětí je na horní svorce usměrňovacího můstku kladné napětí. Diody D2 a D4 jsou otevřené, takže může procházet proud. Polarita napětí na výstupních svorkách je patrná z obrázku. Diody D1 a D3 jsou zavřené.

Při následující půlperiodě střídavého proudu je na horní svorce usměrňovacího můstku záporné napětí, na dolní svorce je však kladné napětí. Nyní jsou otevřeny diody D1 a D3, které na horní výstupní svorku přivádějí kladné napětí a na dolní svorku záporné napětí.

Na horní svorce je tedy trvale kladné napětí, na dolní svorce záporné napětí. Jeho průběh odpovídá průběhu napětí při dvoucestném usměrnění.

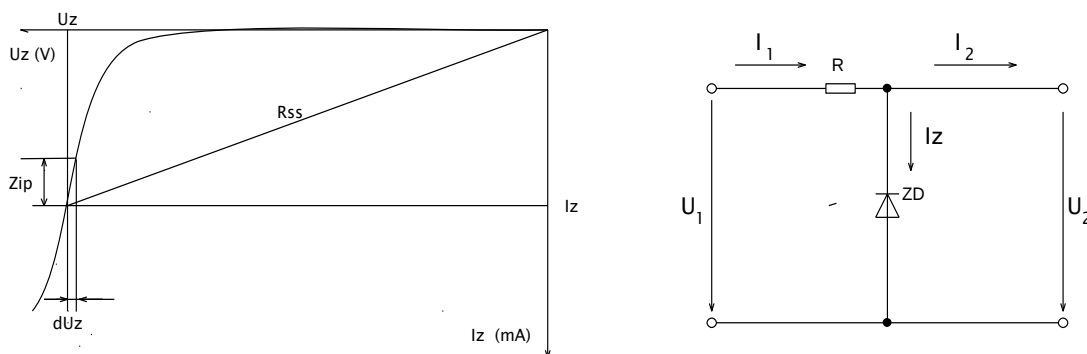


5. K čemu slouží filtrační obvody ve zdroji?

Filtrační obvody slouží k odstranění (nebo zmenšení) střídavé složky z usměrněného napájecího napětí. Nejčastěji se pro tento účel využívá kondenzátorů, rezistorů a tlumivek. Typickým příkladem je filtrační kondenzátor připojený k výstupu usměrňovače.

6. Popište a nakreslete stabilizátor pevného kladného napětí.

Podle připojení regulačního prvku rozdělujeme stabilizátory na paralelní a sériové. Nejjednodušším paralelním stabilizátorem je zapojení se Zenerovou diodou. Zenerova dioda je součástka, u níž dochází v závěrném směru k rychlému nárůstu proudu při pomalém nárůstu napětí. Tato hodnota se nazývá Zenerovo napětí. Používají se do 0,1 A.



VA Charakteristika a zapojení paralelního stabilizátoru se zenerovou diodou

Velikost odporu R určíme ze vztahu
$$R = \frac{(U_1 - U_2)}{I_1} \quad [\Omega; V, A]$$

Obvykle volíme $I_F > I_2$ tak, aby při $I_2 = 0$ nebyla dioda přetížena proudem I_F , a naopak, aby při maximální hodnotě I_2 kdy začíná napětí na diodě prudce klesat, neklesl proud diodou pod minimální hodnotu.

7. Popište princip transformátoru.

Transformátor je elektrický netočivý stroj, který pomocí elektromagnetické indukce mění elektrické napětí a proud při nezměněném kmitočtu.

Po připojení vstupního vinutí na střídavé napětí začne vstupním vinutím protékat střídavý elektrický proud, který vytvoří střídavé magnetické pole. Magnetickým obvodem prochází střídavý magnetický tok a protíná vodiče výstupního vinutí. Ve vodičích výstupní cívky se indukuje střídavé elektrické napětí a po připojení zátěže začne výstupním vinutím protékat elektrický proud. První cívka N_1 je připojena na napětí U_1 , které transformuje. Na sekundární cívce N_2 je transformované napětí U_2 a je k ní rovněž připojen spotřebič.

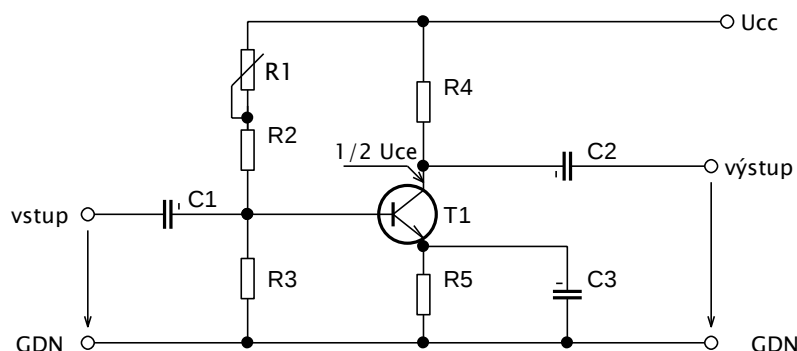
8. Popište princip spínaného zdroje.

Vstupní síťové napětí je usměrněno a vyfiltrováno. Poté je toto stejnosměrné napětí spínáno pomocí výkonových tranzistorů a přiváděno na primární vinutí transformátoru. Kmitočet dosahuje desítek kHz a je řízen oscilátorem (OSC). Velikost dodávané energie je v závislosti na momentální zátěži řízena modulátorem (PWM). Řídící napětí se získává porovnáním (COMP) výstupního napětí s referenčním (REF). Sekundární napětí transformátoru je opět usměrněno a filtrováno.

Zmenšení velikosti a hmotnosti je umožněno tím, že použitý transformátor pracuje na mnohem vyšším kmitočtu, než je kmitočet síťový. Má zpravidla feritové jádro a je mnohonásobně menší než běžný síťový transformátor stejného výkonu.

9. Popište jednostupňový nízkofrekvenční zesilovač ve třídě A. Uveďte základní třídy nízkofrekvenčních zesilovačů.

Na uvedeném schématu je zapojení jednostupňového střídavého zesilovače ve třídě A v zapojení se společným emitorem. Klidový pracovní bod je nastaven odporovým trimrem R1. Ochranný rezistor R2 omezuje maximální hodnotu proudu I_B a zabraňuje překročení proudu I_B tranzistoru. Abychom dosáhli co největší amplitudy výstupního napětí, nastavíme trimrem napětí v kolektoru na $\frac{1}{2} U_{CC}$. Kondenzátory C1 a C2 se nazývají vazební a oddělují stejnosměrné napětí v bázi, respektive v kolektoru tranzistoru. Protože kondenzátory „propouštějí“ pouze střídavý proud, nenaměříme na vstupu ani výstupu žádné stejnosměrné napětí. Rezistor R4 nazýváme pracovní a vlivem změn I_C v rytmu vstupního signálu se na něm vytváří úbytek napětí. R5 společně s C3 stabilizují pracovní bod tranzistoru.

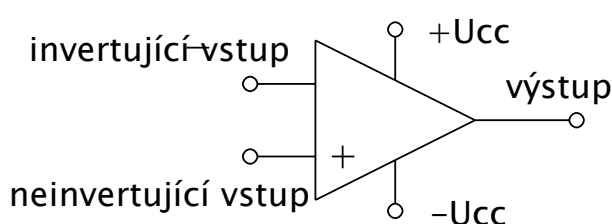


Nízkofrekvenční zesilovače slouží k zesilování slyšitelného pásma kmitočtů, v rozsahu 20 Hz až 20 kHz. Zesilovače, pracující ve třídě A, mají pracovní bod nastaven na střed lineární části převodní charakteristiky, mohou tedy s minimálním zkreslením zpracovat kladná i záporná napětí, a proto se používají k zesílení střídavého napětí. Výkonové nízkofrekvenční zesilovače pracují ve třídě AB, s malým klidovým proudem, nebo B s nulovým klidovým proudem. Výhodou je zvýšení účinnosti. Aby bylo možno zesilovat obě půlvlny střídavého napětí, používá se komplementárních tranzistorů.

10. Jaké jsou základní parametry operačního zesilovače?

OZ by měl mít následující parametry: nekonečně velké zesílení, nekonečně velkou vstupní impedanci, nulovou výstupní impedanci a nekonečně vysokou rychlost přeběhu. V praxi nelze samozřejmě těchto parametrů dosáhnout, proto se můžeme setkat např. s nízkošumovými OZ, které se používají ve vstupních obvodech, vysokofrekvenčními OZ nebo výkonovými OZ. V jednom pouzdře mohou být až čtyři.

Protože jedním ze základních požadavků je zpracování stejnosměrných signálů, musí být OZ napájeny symetrickým napájecím napětím (např. $\pm 15\text{V}$).



Obr. 6.01: Vývodové zapojení operačního zesilovače

Každý OZ má dva vstupy invertující (-) a neinvertující (+), jeden výstup, vývody pro kladné ($+U_{cc}$) a záporné ($-U_{cc}$) napájecí napětí a případně další pomocné vývody pro nastavení symetrie nebo pro kmitočtovou kompenzaci. Číslování i zapojení vývodů se u jednotlivých typů OZ může odlišovat. Konkrétní údaje jsou uvedeny v katalogových listech. Základní vlastnosti ideálního OZ jsou vstupní napěťová nesymetrie, vstupní klidový proud, kmitočtová kompenzace, rychlost přeběhu ($\text{V}/\mu\text{s}$).

11. Popište nízkofrekvenční oscilátor RC, a uveďte podmínky kmitání.

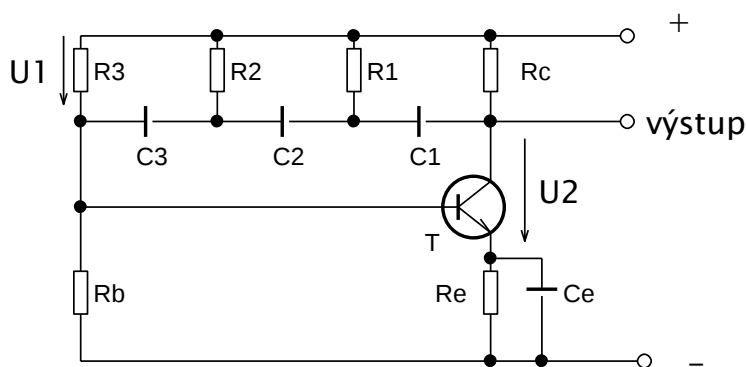
Oscilátor je zdroj střídavého napětí, jehož frekvence je určena použitými součástkami. Základem je zesilovač, z jehož výstupu je zavedena na vstup kladná zpětná vazba. Díky tomu je možné realizovat zpětnovazební obvody s přesně určenými vlastnostmi.

Oscilátory RC obsahují zesilovač a členy RC, určující kmitočet a potřebný fázový posuv ve zpětnovazební větvi, kterým je určen kmitočet. V sinusových oscilátorech je zavedena slabá kladná zpětná vazba, která předává energii od zesilovače k pasivním členům.

V oscilátoru dochází k oscilacím při splnění amplitudové i fázové podmínky. Tranzistor, v zapojení se společným emitorem, posouvá fázi signálu o 180° , takže ke splnění fázové

podmínky oscilací musí být fázový posuv smyčky kladné zpětné vazby také 180° , aby celkový výsledný posuv byl 360° . Jednotlivý článek posouvá fázi o 60° , proto je nutné, aby pro posuv o 180° byl použit řetězec složený minimálně ze tří fázovacích článků.

Z řetězce derivačních článků vznikne horní propust a z řetězce integračních článků vznikne dolní propust. Používá se tříčlávková horní i dolní propust. Pro splnění amplitudové části oscilační podmínky musí být útlum signálu v pasivní části oscilátoru vyrovnán zesílením v aktivní části. Pro oscilace musí být celkový přenos $A_u \beta > 1$. Praktické užití je převážně pro oscilátory s pevnou frekvencí. Příklad jednoduchého tříčlávkového oscilátoru RC s horní propustí vidíme obrázku. Řídící obvod tvoří tři derivační články, z nichž každý posunuje fázi o 60° celkem o 180° .



12. Jaké jsou základní optoelektronické prvky?

Základní prvky optoelektroniky tvoří polovodičové zdroje záření (elektroluminiscenční dioda, laserová dioda), které umožňují přímou konverzi elektrické energie na optickou, a detektory optického záření. Detektory převádějí energii optického záření na jinou měřitelnou veličinu.

Optoelektronické zdroje světla: LED diody, polovodičový laser, displeje led a LCD, osvětlení a signalizace.

Optoelektronické senzory: fotorezistory, infradiody, fototranzistory.

Optické spínací prvky: optočleny, optotriaky a další prvky a jejich kombinace pro různé aplikace v elektronice.

Přenos informací: optická vlákna případně přenos vzduchem.

2 ČÍSLICOVÁ TECHNIKA

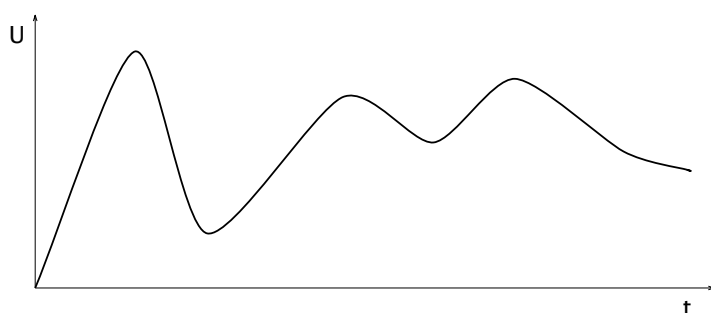
2.1 Základní pojmy

Číslicová technika je poměrně mladý obor, avšak nemůžeme opomenout, že prvopočátek oboru sahá několik století nazpět. Opravdový rozmach však nastal až s rozvojem elektroniky. Nejznámější využití číslicové (digitální) techniky je ve spojení s počítači. Není to však využití jediné. Přesto si však mnoho lidí pod pojmem číslicová technika představuje spíše výpočetní techniku s programováním.

Co si tedy představit pod pojmem číslicová technika? Číslicová technika je obor, který se zabývá zpracováním a přenosem číslicově kódovaných informací. Nositel informace v elektronických obvodech je elektrický signál, tedy např. určitá velikost napětí.

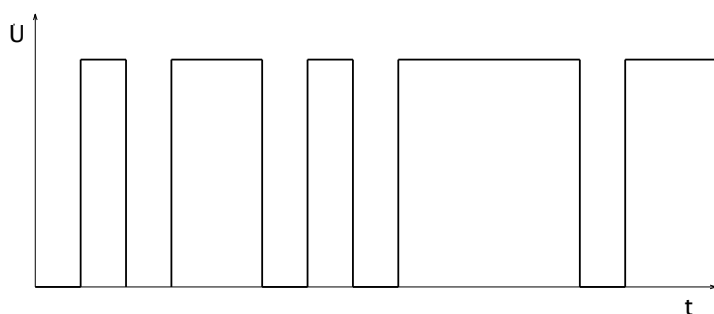
2.1.1 Analogový a číslicový signál

V předešlé kapitole jste se setkali s pojmem analogový signál. Analogový signál (obr. 2.1) může nabývat jakékoliv hodnoty od minimální hodnoty až po hodnotu maximální. Navíc v celém intervalu je možné jeho hodnotu měnit plynule (spojitě).



Obr. 2.1: Analogový signál

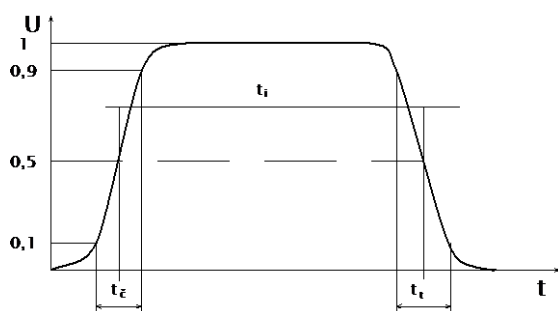
Číslicový (digitální) signál (obr. 2.2) je na rozdíl od signálu analogového nespojitý, tzn., že k jeho změně dochází skokem. Nabývá dvou hodnot.



Obr. 2.2: Číslicový signál

2.1.2 Impuls

Protože vstupy a výstupy číslicových obvodů tvoří impulsy, nebo jejich sledy, je nutno tento pojem nejdříve definovat: Impuls (obr. 3) je definován jako jednorázový jev konečné délky. Impuls je jedinečný svým tvarem (pravoúhlý, jehlový, trojúhelníkový, atd.), amplitudou, dobou trvání a okamžikem vzniku.



Obr. 2.3: Impuls

Impuls (obr. 2.3) můžeme popsat následujícím souborem údajů:

t_i – aktivní délka impulsu

t_ϵ – čelo impulsu

t_τ – tyl impulsu

Vydeme-li z reálného tvaru impulsu, lze stanovit následující důležité pojmy:

Velikost impulsu – můžeme definovat hodnotu střední, efektivní, okamžitou. Zpravidla však udáváme hodnotu maximální, kterou nazýváme amplitudou.

Doba trvání impulsu - (aktivní délka) - je rozpětí mezi 50 % hodnotou přední a zadní hrany amplitudy impulsu (čela a týlu). Náběh a doběh je časové rozpětí mezi 10% a 90% amplitudy.

Sled impulsů - pravidelně se opakující impulsy.

Frekvence – počet impulsů za sekundu.

Perioda – doba daná součtem doby trvání impulsu a mezery mezi nimi.

Střída – poměr doby trvání impulsu k délce periody.

Impulsové průběhy většinou sledujeme osciloskopem, který na stínítku obrazovky znázorní časový průběh připojeného signálu. V mnoha případech se nám může zobrazení na osciloskopu jevit jako dokonalý pravoúhlý impuls, avšak přesný obraz tvaru měřeného impulsu můžeme posoudit pouze s použitím vhodného časového měřítka.

2.1.3 Základní pojmy užívané v číslicové technice

Logické operace – operace s logickými proměnnými podle zákonů určité logické algebry

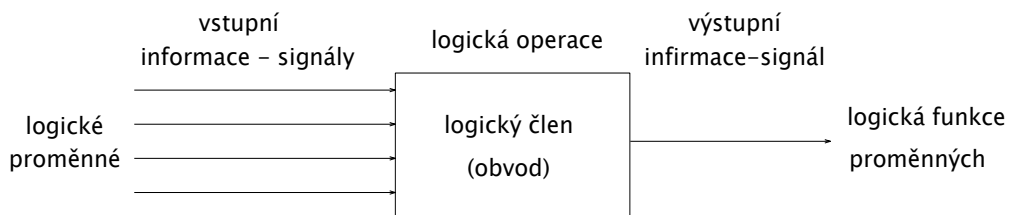
Dvojková logická operace – operace může nabývat pouze dvou diskrétních hodnot, které se označují jako logické úrovně a popisují se symboly log 1 (nebo jen 1) a log 0 (nebo jen 0). Pro označení log 0 je také používáno L (Low), pro log 1 H (High). Vztahy mezi logickými proměnnými se řídí zákony dvouhodnotové logické algebry – v praxi je využívána Booleova algebra.

Logická funkce – předpis, který jedné nebo kombinaci více vstupních proměnných jednoznačně přiřazuje hodnoty jediné závislé logické proměnné. Základními logickými funkcemi jsou logický součet AND, logický součin OR a negace NOT.

Hradlo – také logický člen, elektronické zapojení, které je uspořádáno tak, aby plnilo určitou logickou funkci.

Logický obvod – integrovaný obvod, pracující s číslicovým signálem (úrovněmi 0/1), vzniká spojením několika logických členů (hradel) do skupiny.

Činnost základních stavebních částí číslicových zařízení je schematicky shrnuje na obr. 2.4.



Obr. 2.4: Logický člen

2.1.4 Číselné soustavy

Každé číslicové zařízení pracuje s čísly, která jsou pro nás symbolem určitého množství – kvantity. Nám je vlastní desítková číselná soustava, kterou využíváme bez problémů v běžném životě a při aritmetických operacích dostáváme vždy jednoznačný a nám srozumitelný výsledek. Desítková číselná soustava používá číslice 0, 1, 2, 3, 4, 5, 6, 7, 8 a 9.

Desítková číselná soustava není však vhodná pro použití v počítačích či číslicových systémech, protože právě číslicové zařízení by muselo rozlišovat deset úrovní. Proto v těchto zařízeních využíváme soustavy o jiných základech, nejčastěji o základu 2, kdy zařízení pracuje s číslicemi 0 a 1.

Libovolné číslo o základu $F_z \geq 0$ lze vyjádřit v soustavě o základu Z mnohočlenem

$$F_z = a_m z^m + a_{m-1} z^{m-1} + \dots + a_0 z^0 = \sum_{i=0}^m a_i z^i$$

kde a_i jsou číselné hodnoty na jednotlivých řádových místech čísla F_z .

Tomuto zápisu se také říká **Hornerovo schéma**.

Např. číslo $F_{10} = 190$ rozložené dle Hornerova schématu vypadá následovně:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 190$$

V logických systémech se nejčastěji využívají tyto číselné soustavy:

- dvojková (binární) – soustava o základu $Z=2$ (hodnoty 0,1);
- osmičková (oktalová) – soustava o základu $Z=8$ (0, 1, 2, 3, 4, 5, 6, 7);

- šestnáctková (hexadecimální) – soustava o základu $Z=16$ (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

V převodní tabulce 1 uvádíme zobrazení čísel v nejčastěji používaných soustavách pro čísla 0 – 15.

Dekadická soustava $Z=10$	Dvojková soustava $Z=2$	Osmičková soustava $Z=8$	Šestnáctková soustava $Z=16$
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Tab. 1: Zobrazení čísel v nejčastěji používaných soustavách

2.2 Logické funkce

Ve všech oborech lidské činnosti se setkáváme s případy, kdy předměty nebo jejich vlastnosti nabývají jen dvou užitečných stavů. Tyto stavy jsou často vzájemně opačné. Stav předmětu nebo jeho vlastnosti pak můžeme jednoznačně vyjádřit výrokem ANO (spínač je sepnut – žárovka svítí) nebo opačným výrokem NE (spínač je rozepnut – žárovka nesvítí). Informaci o určitém stavu, kterou vnímáme svými smysly, můžeme využít k rozhodnutí o nějaké další účelové činnosti. Jednou z takových činností může být negace zjištěného stavu, kdy daný stav změníme ve stav opačný. Negací převádíme stav ANO na stav NE a naopak.

Rozhodnutí o další účelové činnosti jsou často založena na informacích o stavech (ANO – NE) většího počtu předmětů nebo jejich vlastností. Taková rozhodování označujeme jako *rozhodování logická*. Předpis, podle něhož se rozhodujeme k žádané další činnosti, označujeme jako *logickou funkci proměnných*. Vlastní proces rozhodování, tj. zpracování proměnných podle předepsané logické funkce, nazýváme *logickou operací*. Proměnné, z nichž vychází logická operace, pak označujeme jako *logické proměnné*. Vlastní mechanismus, který zpracovává signál podle určené logické funkce a realizuje tak předepsanou logickou operaci, označujeme jako logický člen nebo **logický obvod**.

Podle druhu realizované funkce dělíme logické obvody na:

- *kombinační logické obvody* - výstup je vždy určen kombinací vstupních veličin;
- *sekvenční logické obvody* - výstup je určen nejen kombinací vstupních veličin, ale i hodnotou předchozího stavu logického obvodu. Z toho tedy plyne, že sekvenční logický obvod má paměť.

Přítomnost logické proměnné vyjádřené výrokem ANO označujeme symbolem 1 příp. H. Nepřítomnost logické proměnné vyjádřené výrokem NE označujeme symbolem 0 příp. L.

2.2.1 Základní logické funkce - logický součet, součin, negace

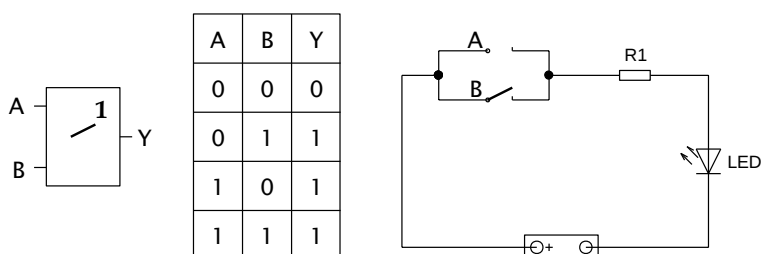
V polovině 19. století zpracoval irský matematik G. Boole matematickou logiku, tj. souhrn pravidel a zákonů, které umožňují pracovat s logickými výroky jako s logickými proměnnými a funkcemi, a to formou algebraických operací.

K vyjádření libovolné logické funkce se v Booleově algebře používají jen tři základní funkce:

- logický součet (OR);
- logický součin (AND);
- logická negace (NOT).

2.2.1.1 Logický součet - OR

Logický součet OR (česky NEBO) je logická funkce ve tvaru $Y = A + B$, kde na levé straně pravdivostní tabulky jsou hodnoty vstupních veličin označených A a B a na pravé straně pravdivostní tabulky je hodnota výstupní veličiny Y. Schematická značka (obr. 2.5) platí pro dvouvstupový logický člen OR, počet vstupů však může být libovolně větší.



Obr. 2.5: OR – schematická značka, pravdivostní tabulka a realizace

Logickou funkci OR popisuje pravdivostní tabulka, která je popisem všech možných stavů, které mohou nastat. Pro dvě vstupní proměnné A a B můžeme mít $2^2 = 4$ stavů (řádků v tabulce), pro tři vstupní proměnné $2^3 = 8$ stavů, pro čtyři vstupní proměnné $2^4 = 16$ stavů atd.

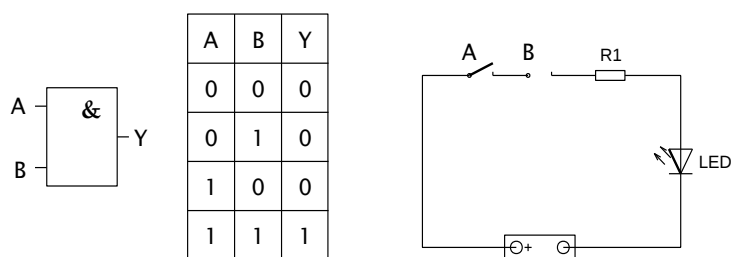
Při použití matematické funkce $Y = A + B$ dostaneme pro všechny vstupní proměnné hodnoty A a B následující hodnoty výstupu Y: $0 + 0 = 0$; $0 + 1 = 1$; $1 + 0 = 1$ a $1 + 1 = 1$.

Uvedené údaje tvoří pravdivostní tabulku logické funkce OR na obr. 5. U logické funkce OR si můžeme všimnout, že pokud je kdekoli na vstupu úroveň log. 1, je na výstupu také hodnota log. 1.

Pro jednodušší pochopení uvádíme u zapojení (obr. 2.5) logického členu OR odpovídající zapojení spínačů. Pro tento způsob znázornění platí, že spínače fungují jako vstupy, přičemž sepnutý spínač znázorňuje úroveň log. 1, rozepnutý log. úroveň 0. Prochází-li obvodem el. proud, pak svítící LED dioda představuje výstup s úrovní log. 1, nesvítí-li LED dioda, pak je na výstupu úroveň log. 0.

2.2.1.2 Logický součin - AND

Logický součin AND (čti "end") je logická funkce ve tvaru $Y = A \cdot B$, kde na levé straně pravdivostní tabulky jsou hodnoty vstupních veličin označených A a B a pravdivostní tabulky na pravé straně je hodnota výstupní veličiny Y. Schematická značka (obr. 2.6) platí pro dvouvstupový logický člen AND, počet vstupů však může být opět libovolně větší.



Obr. 2.6: AND – schematická značka, pravdivostní tabulka a realizace

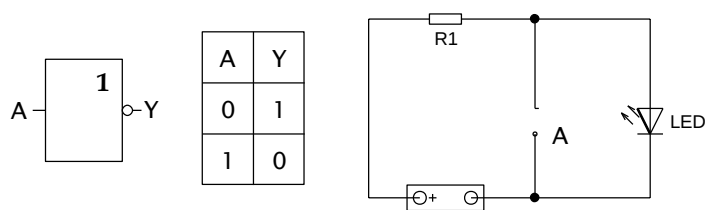
Logickou funkci AND opět můžeme popsat pravdivostní tabulkou (obr. 2.6) a pro jednodušší pochopení opět přidáváme zapojení s tlačítky a LED diodou.

U logické funkce AND si můžeme všimnout, že pokud je kdekoliv na vstupu úroveň log. 0, je na výstupu také hodnota log. 0. Úroveň log. 1 je na vstupu pouze tehdy, když všechny vstupy mají úroveň log. 1.

2.2.1.3 Logický operátor NOT - invertor

Funkce logické negace se algebraicky vyjádří ve tvaru: $Y = \overline{A}$, což také čteme "A non". Log. funkce NOT má hodnotu log. 1 tehdy, je-li hodnota vstupní proměnné 0 a naopak. Jinak řečeno, logická negace převrací hodnotu vstupního signálu.

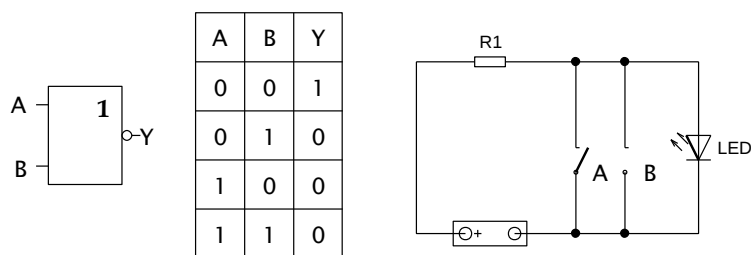
Logický člen, který realizuje funkci negace se nazývá INVERTOR (NOT). Na obr. 2.7 je schematická značka, pravdivostní tabulka a realizace pomocí LED diody a tlačítka.



Obr. 2.7: NOT – schematická značka, pravdivostní tabulka a realizace

2.2.1.4 Negace logického součtu - NOR

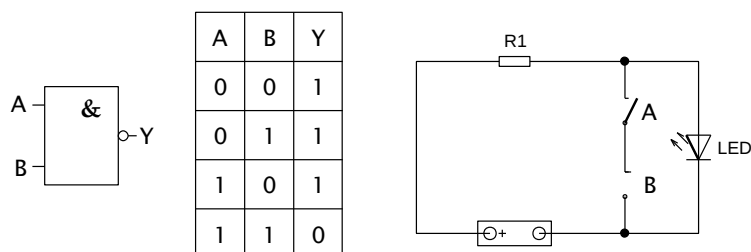
Negace logického součtu NOR je logická funkce ve tvaru $Y = \overline{A+B}$, kde tato log. funkce vznikne negací, tedy připojením NOT na výstup OR. Schematická značka, pravdivostní tabulka a realizace je na obr. 2.8.



Obr. 2.8: NOR – schematická značka, pravdivostní tabulka a realizace

2.2.1.5 Negace logického součinu - NAND

Negace logického součinu NAND je logická funkce ve tvaru $Y = \overline{A \cdot B}$, kde tato log. funkce vznikne negací, tedy připojením NOT na výstup AND. Schematická značka, pravdivostní tabulka a realizace je na obr. 2.9.



Obr. 2.9: NAND – schematická značka, pravdivostní tabulka a realizace

2.2.2 Booleova algebra

Logické členy, ať již jsou prakticky projedeny jakkoli, mají vždy vestavěnou pevnou logickou funkci, podle níž pak mohou rozhodovat. Jediný logický člen plní obvykle jen velmi jednoduché úkoly. Možnosti praktického nasazení takového členu jsou zřejmé na první pohled z jeho logické funkce. Existuje-li však mechanismus (zde logický člen), který je schopen automaticky rozhodovat podle jednoduché logické funkce, je zřejmě možno sestavou takových členů řešit zařízení, která rozhodují podle složitějších logických zákonitostí. K tomu je nutno, aby celé rozhodování ve složité situaci bylo možno rozložit na jednotlivá dílčí rozhodnutí, která lze realizovat jednoduchými logickými členy. Tento přístup vede k účinnému řešení automatizace rozhodování ve složitých podmínkách, jakými jsou např. situace při řízení výrobních pochodů a obecně pochodů, při nichž vstupuje řada vzájemně závislých proměnných. Řešit sestavy logických členů k dosažení logické funkce větší složitosti je možné na základě dobré znalosti logické funkce použitých členů. Často však docházíme k řešení, která nejsou optimální, tj. která vedou k použití většího počtu logických členů, než je nezbytně nutné. Dobrou pomůckou při řešení složitějších logických situací je pak znalost teoretických zákonitostí veličin se dvěma stavy. Takové zákonitosti nalezneme v Booleově algebře, která, ač starého data, je velmi užitečná při řešení moderních technických problémů.

Booleova algebra je speciálním druhem algebry, která se zabývá jen funkcemi se dvěma stavy. Jako v běžné algebře jsou i Booleovy algebraické vztahy složeny z proměnných, z konstant a ze značek operací. V Booleově algebře může mít proměnná pouze dva stavy, tj. 1 nebo 0.

Základní vztahy Booleovy algebry

Zákon asociativní:

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Zákon asociativní:

$$A + B + C = A + (B + C) = (A + B) + C \quad A \cdot B \cdot C = A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

Zákon distributivní:

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

Pravidlo dvojité negace:

$$\overline{\overline{A}} = A$$

Pravidlo o neutrálnosti a agresivnosti konstant 0, 1:

$$A + 1 = 1$$

$$A + 0 = A$$

$$A \cdot 1 = A$$

$$A \cdot 0 = 0$$

Pravidlo negace:

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

Pravidlo absorpce:

$$A + A = A$$

$$A \cdot A = A$$

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

Pravidlo o absorpci negace:

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

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

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

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

Teorém De Morganův:

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

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

Vztahy konstant:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

Použití Booleovy algebry

Chceme-li k řešení daného problému použít matematický způsob řešení, musíme nejprve úlohu formulovat v symbolické formě. Každou proměnnou nejprve pojmenujeme a označíme. Podobně pojmenujeme a označíme výsledek hledané operace. Poté sestavíme vztah, který stanoví podmínky proměnných pro dosažení žádaného výsledku. Postup ukážeme na následujícím jednoduchém příkladu.

Mějme stroj, který má pracovat po splnění podmínky, že musí být v činnosti 2 z 3 mazacích čerpadel (třetí je náhradní). Pojmenujme a označme proměnné: činnost jednotlivých mazacích čerpadel označme jako proměnné A, B, C. Výsledek operace je start stroje, který označíme Y. Start stroje nastane, bude-li v činnosti čerpadlo A a B nebo A a C nebo B a C". Tomu odpovídá algebraický tvar: $Y = A.B + A.C + B.C$.

Při řešení složitějších úloh získáváme obvykle značně složité a nepřehledné vztahy proměnných k výsledku. S využitím pravidel Booleovy algebry se snažíme tyto vztahy pokud možno co nejvíce zjednodušit. Teprve nejjednodušší dosažený tvar funkce pak realizujeme logickými členy. Mějme např. funkci:

$$Y = A.B.C.\bar{D} + A.\bar{B}.C.D + A.B.\bar{C}.\bar{D} + A.\bar{B}.\bar{C}.D + A.B.D + A.B.C$$

Pro zjednodušení nejprve vytkneme společné členy před závorku:

$$Y = A.B(C.\bar{D} + \bar{C}.D + D + C) + A.\bar{B}.(C.D + \bar{C}.D)$$

Vztah dále upravujeme postupně po závorkách. Nejprve upravíme výraz v první závorce.

$$\text{Bude: } C.\bar{D} + \bar{C}.D + D + C = \bar{D}(C + \bar{C}) + D + C$$

Nyní využijeme pravidel Booleovy algebry, podle níž je:

$$C + \bar{C} = 1 \quad \bar{D}.1 = \bar{D} \quad \bar{D} + D = 1 \quad 1 + C = 1$$

Po dosazení bude výraz z první závorky roven jednotce. Nyní upravíme výraz v druhé závorce. Bude: $C.D + \bar{C}.D = D(C + \bar{C}) = D$

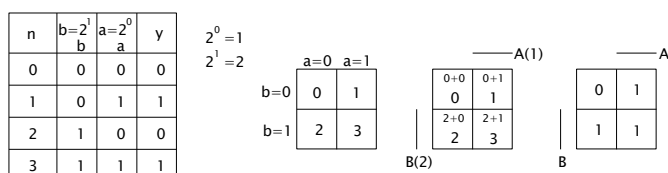
Úpravami se celá funkce zjednodušila do tvaru: $Y = A.B + A.\bar{B}.D$ tento tvar dále upravíme: $Y = A(B + \bar{B}.D)$ a po poslední úpravě: $Y = A.(B + D)$

Jiná metoda minimalizace spočívá v technice mapování např. s použitím Karnaughovy mapy.

2.2.3 Minimalizace s užitím Karnaughovy mapy

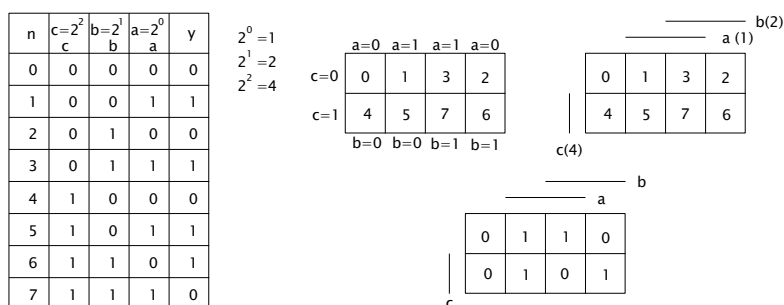
Minimalizací logické funkce získáme nejjednodušší možný zápis pro danou funkci, což obvykle vede k výraznému zjednodušení praktické realizace dané funkce.

Každé políčko Karnaughovy mapy, odpovídá určitému vstupnímu stavu logické funkce.



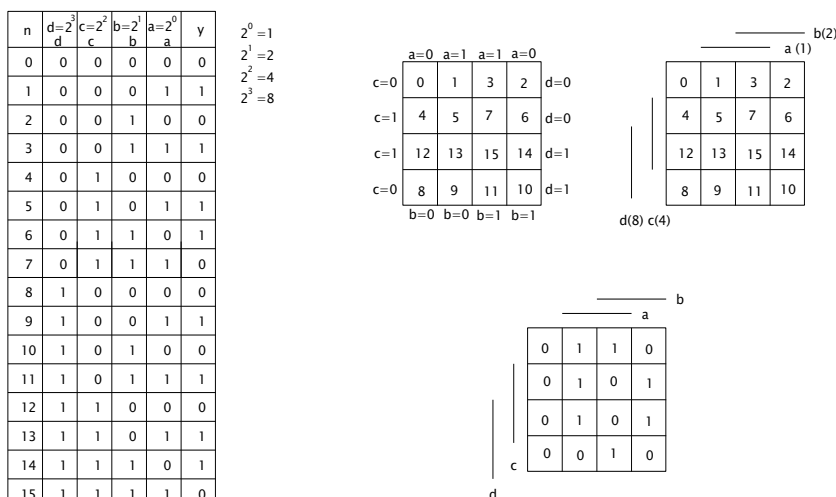
Obr. 2.10: Karnaughova mapa pro 2 vstupní proměnné

Na obrázku č. 2.10 je uvedena pravdivostní tabulka a Karnaughova mapa pro dvě vstupní proměnné, na obrázku č. 2.11 pro tři vstupní proměnné a na obrázku č. 2.12 pro čtyři vstupní proměnné. Ve sloupci n jsou uvedeny čísla řádků pravdivostní tabulky.



Obr. 2.11: Karnaughova mapa pro 3 vstupní proměnné

Nyní si např. všimněme Karnaughovy mapy pro 3 vstupní proměnné (obrázek č. 2.11). Ve sloupcích c, b, a jsou uvedeny vstupní hodnoty (za povšimnutí stojí, že kombinace hodnot c, b, a reprezentují odpovídající číslo n ve dvojkové soustavě, tedy např. pro $n = 2$ je kombinace c, b, a = 010). Při zápisu do mapy např. řádku $n = 0$ ($a=0$, $b=0$, $c=0$) odpovídá políčko v levém horním rohu, kombinaci proměnných $a=0$, $b=1$, $c=0$ ($n=2$) náleží políčko v pravém horním rohu. Tabulku s čísly řádků lze získat jednoduše součtem hodnot jednotlivých parametrů, např. řádek 0 odpovídá součtu $0.2^2 + 0.2^1 + 0.2^0 = 0$, nebo řádek 2 odpovídá součtu $0.2^2 + 1.2^1 + 0.2^0 = 2$. Nejnižšímu bitu, který má hodnotu 2^0 , tedy 1 pak odpovídá obvykle vstupní parametr $a(1)$, $2^1=2$ - parametr b, $2^2=4$ parametr c,...



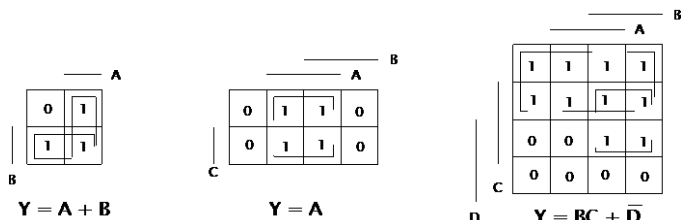
Obr. 2.12: Karnaughova mapa pro 4 vstupní proměnné

Minimalizace spočívá v rozkladu mapy na soubor čtyřúhelníků, obsahujících vždy maximální, avšak právě mocnině čísla dva rovný počet políček, odpovídajících vstupním stavům, pro které logická funkce nabývá výhradně hodnoty 1 (H).

Utváření čtyřúhelníků, sdružujících právě sousední políčka mapy, jímž odpovídající vstupní stavy se liší vzájemně pouze v jediné proměnné, vyžaduje vytvoření představy o zobrazení mapy na plášti toroidu. Sousedními jsou pak i políčka, zakreslená na protilehlých okrajích mapy. Každý čtyřúhelník lze popsat součinem, obsahujícím ty vstupní logické proměnné, které ve všech políčkách čtyřúhelníka nabývají ekvivalentní logické hodnoty.

Pokud příslušná vstupní proměnná nabývá hodnoty 1 je obsažena v součinu přímo, pokud příslušná vstupní proměnná nabývá hodnoty 0, je v součinu negována.

Příklady minimalizace funkcí pro dvě, tři a čtyři vstupní proměnné pomocí Karnaughovy mapy jsou uvedeny na obrázku č. 2.13.



Obr. 2.13: Příklady minimalizace funkcí

Logická funkce je zapsána jako součet součinů, odvozených z jednotlivých čtyřúhelníků za předpokladu, že tyto pokrývají všechny jedničkové stavy logické funkce.

Pro každou výstupní proměnnou se musí sestavit samostatná mapa.

2.3 Logické členy

Nyní se podíváme na logické členy. Logický člen (také hradlo) je elektronické zapojení, které je uspořádáno tak, aby plnilo určitou logickou funkci. Jedná se tedy o elektronický obvod, který má alespoň jeden vstup a jeden výstup a který transformuje vstupní dvouhodnotový logický signál podle určité funkce.

2.3.1 Rozdělení logických členů

Logické členy můžeme dělit podle různých hledisek:

- podle realizované funkce

- kombinační
- sekvenční

- podle druhu signálu, který je nositelem logické informace

- Elektromechanické (signál je U nebo I a následně síla - relé)
- elektrické (U nebo I)
- optoelektrické (světelný tok $\rightarrow$ U nebo I)
- pneumatické (mechanický tlak)

- podle použité součástkové základny

- elektromechanické (logická funkce je realizována pomocí relé)
- diodové (diody)
- tranzistorové (tranzistory, diody, rezistory)
- integrované (integrované obvody)

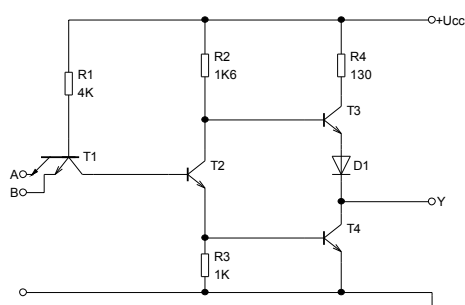
V názvech logických integrovaných obvodů, složených obvykle ze tří písmen, jsou obsaženy použité součástky (první dvě písmena), třetí písmeno L vždy znamená začátek slova logika. Některé technologie logických integrovaných obvodů si připomeneme:

- RDL – obsahuje nejjednodušší logické členy sestavené z rezistorů a diod (diodová logika)

- DTL – logické obvody s diodami na vstupu a tranzistory na výstupu (diodově tranzistorová logika)
- TTL – nejrozšířenější technologická řada, vstupní i výstupní členy tvoří tranzistory (tranzistorově-tranzistorová logika)
- ECL – tento typ logiky se používá pro velmi rychlé číslicové obvody, větší příkon, dvě napájecí napětí (emitorově vázaná logika)
- CMOS – tato logika využívá unipolární tranzistory, tj. tranzistory řízené elektrickým polem, velmi rozšířená (komplementární MOS technologie)

2.3.2 Základní logický člen integrovaných obvodů TTL

Základním stavebním prvkem všech obvodů TTL je logický člen AND (viz obr. 2.14). Totéž zapojení, příp. s jiným počtem emitorů vstupního tranzistoru se používá u celé řady dalších kombinačních obvodů (a též u obvodů sekvenčních).



Obr. 2.14: Vnitřní zapojení hradla NAND se dvěma vstupy A, B

Elektrickou funkci obvodu je důležité sledovat ze tří hledisek. První hledisko se týká způsobu, jakým dochází ke změně stavu obvodu. Vlastnosti obvodu tohoto druhu můžeme popsat přenosovou charakteristikou, která udává závislost výstupního napětí obvodu na vstupním napětí obvodu. Druhé hledisko se týká elektrických vlastností vstupu obvodu. Můžeme je popsat vstupní charakteristikou, která udává závislost vstupního proudu na vstupním napětí. Třetí hledisko se pak týká elektrických vlastností výstupu obvodu. Lze je popsat výstupní charakteristikou, která udává závislost výstupního napětí na výstupním proudu obvodu.

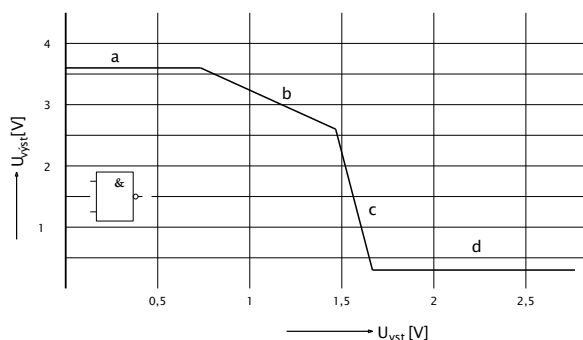
U elektrických obvodů logické stavy 0 a 1 nepřísluší určitým diskretním napětím, ale pásmům napětí. V symbolice elektronických logických obvodů se proto k odlišení nepoužívají symboly 0 a 1 ale symboly L a H. L znamená malé napětí, tj. stav 0, H znamená velké napětí, tj. stav 1. Úrovně pásma TTL jsou zobrazena v následující tabulce.

Logická úroveň	Vstup	Výstup
Log. 0	0 – 0,8 V	0 – 0,4 V
Log. 1	2 – 5 V	2,4 – 5 V

Tab. 2: Logické úrovně TTL

Přenosová charakteristika

Činnost obvodu z obr. 2.14 lze sledovat na jeho přenosové charakteristice, která je naznačená na obr. 2.15.



Obr. 2.15: Přenosová charakteristika log. členu NAND

Přivedeme-li na jeden (nebo oba) vstup obvodu signál o úrovni L, bude příslušný přechod (přechody) báze - emitor transistoru T1 polarizován v propustném směru, neboť báze bude kladnější než emitor. Tranzistor T1 bude v nasyceném stavu - bude mít malý odpor mezi kolektorem a emitorem. Tranzistorem T1 se Odvádějí nosiče proudu z báze tranzistoru T2, kde byly nahromaděny při předchozím pracovním cyklu. Tranzistory T2 a T4 jsou účinkem tranzistoru T1 uzavřeny. Tranzistor T3 pracuje jako emitorový sledovač. Udržuje výstupní úroveň obvodu ve stavu H, tj. napětí, které je asi o 1,5V menší, než je napětí U_{cc} (část a přenosové charakteristiky).

Zvětší-li se napětí vstupu (úroveň L) nad asi 0,7 V, začne se tranzistor T2 otevírat, avšak tranzistor T4 zůstane uzavřen. Při zvětšování napětí na vstupu se bude výstupní napětí (stav H) zmenšovat (část b charakteristiky).

Jakmile napětí na vstupu dosáhne asi 1,4 V, otevře se i transistor T4. Zvětší celkové zesílení obvodu a při malé změně napětí na vstupu dojde k velké změně napětí na výstupu (část c charakteristiky).

Při dalším zvětšování napětí na vstupu se bude tranzistor T2 dokonaleji otevírat a napětí na jeho kolektoru se bude zmenšovat. Tím se bude uzavírat transistor T3, jemuž je účinkem tranzistoru T2 odnímán proud báze. K tomu přispívá dioda D1, která zabezpečuje, že napětí na emitoru tranzistoru T3 bude při sepnutých tranzistorech T2 a T4 prakticky stejné jako napětí na bázi tranzistoru T3. Transistor T4 je, nyní nasyceném stavu, a výstupní napětí obvodu je rovné saturačnímu napětí tohoto tranzistoru, tj. asi 0,4 V. Vstupní napětí má úroveň H (část d charakteristiky).

Budeme-li měnit nyní vstupní úroveň H na úroveň L, probíhá celý pochod obráceně. Nosiče proudu nahromaděné na bázi tranzistoru T4 jsou odváděny rezistorem R3, nosiče proudu na bázi tranzistoru T2 jsou odváděny tranzistorem T4, který je nasycen. Tranzistor T3 se otevírá a pracuje jako emitorový sledovač.

Základní parametry logických obvodů

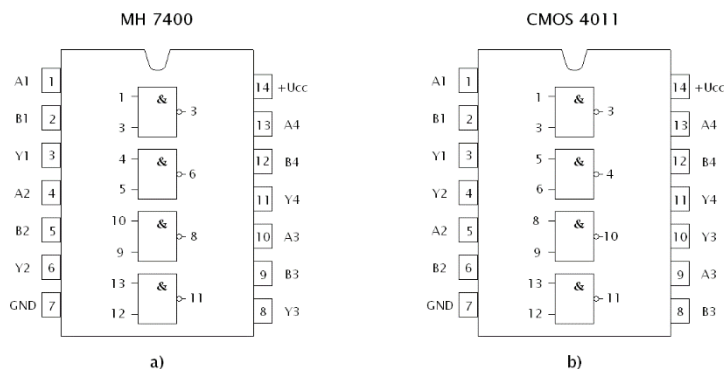
Chceme-li správně používat logické obvody, je potřeba mít představu o základních parametrech, které je charakterizují. K základním parametrům patří:

- *šumová odolnost* neboli odolnost proti šumu představuje odolnost dané technologie proti parazitnímu rušení;
- *logický zisk* udává maximální počet vstupů stejné technologie, které můžeme zapojit na výstup logického členu při zachování logických úrovní;
- *napájecí napětí*
- *vstupní a výstupní napěťové úrovně a proudy*
- *doba zpoždění signálu mezi vstupem a výstupem*

Podrobnější údaje najdeme v katalogu součástek.

Pouzdra logických obvodů

Na obrázku č. 2.16 je jeden z běžně používaných logických obvodů, kde najdeme čtveřici hradel typu NAND. Všechna čtyři hradla mají stejnou funkci, a tudíž jsou záměnná.



Obr. 2.16: Pouzdro log. obvodu NAND

Na obrázku č. 2.16a je provedení TTL, na obrázku č. 2.16 b je provedení CMOS.

2.3.3 Porovnání vlastností jednotlivých rodin logických obvodů

V dnešní době je vyráběno mnoho typů logických obvodů. Prvně byly vyráběny obvody označované TTL. V současné době jsou nahrazeny dokonalejšími nástupci. Sem patří například známý obvod 7400 (čtveřice 2vstupových hradel NAND).

Přímým nástupcem řady TTL je řada TTL LS (Low power & Shottky). Ty jsou navrženy vnitřně tak, že vstupy mají nižší spotřebu (viz tab.). Dále je paralelně k přechodům B-C všech tranzistorů připojena Schottkyho dioda. Protože je prahové napětí Schottkyho diody nižší, než prahové napětí prostého PN přechodu, brání dioda saturaci přechodu B-C. Výsledkem je rychlejší vypínání tranzistoru, protože ten při přechodu mezi stavem zapnuto a vypnuto neprochází saturační oblastí. Patří sem například 74LS00 (obdoba 7400). Řada CMOS 4000 je vyrobena technologií CMOS (pracuje s unipolárními tranzistory s indukovaným kanálem P a N). Vnitřní odpor těchto tranzistorů ve stavu sepnuto je poměrně vysoký, také je vysoká parazitní kapacita mezi hradlem a emitorem. To způsobuje poměrně špatné dynamické parametry (nízký mezní kmitočet a dlouhé doby náběhu a poklesu, viz tab.). Zdánlivou výhodou je možnost práce v širokém rozsahu napájecích napětí (3 až 15 V). Tato skutečnost je zajištěna optimalizací parametrů pro maximální napájecí napětí. Při běžném napětí 5 V jsou obvody velmi pomalé. Příkladem je 4011 (dvojice 4vstupových hradel NAND). Řada HC (High speed Cmos) je rovněž vyráběna technologií CMOS. Vlastnosti unipolárních tranzistorů jsou

optimalizovány na napětí 5 V. V této řadě jsou vyráběny nejenom obvody se stejnou funkcí jako v řadách TTL a TTL LS, ale i CMOS 4000. Tak si můžete koupit obvody 74HC00, ale i 74HC4011.

Přímým nástupcem řady HC je řada HCT (High speed Cmos compatible with TTL). Vlastnosti jsou upraveny tak, aby byly tyto obvody slučitelné (kompatibilní) s obvody TTL nebo TTL LS. Cena je srovnatelná s obvody HC, a podobně lze koupit například obvody 74HCT00 nebo 74HCT4011.

Parametr	TTL	TTL LS	HC	HCT	CMOS 4000
I _{vst log. 0} [mA]	-1,6	-0,36	$-6 \cdot 10^{-3}$	-10^{-3}	-10^{-3}
I _{vst log. 1} [mA]	0,04	0,1	$6 \cdot 10^{-3}$	10^{-3}	10^{-3}
I _{výst log. 0} [mA]	20	4	4	6	0,4
I _{výst log. 1} [mA]	-0,4	-0,4	-4	-6	-0,4
napájecí napětí [V]	4,75 až 5,25	4,5 až 5	2 až 7	4,5 až 5	3 až 15
U _{vst log. 0} [V]	0 až 0,8	0 až 0,8	0 až 1	0 až 0,8	0 až 1,5
U _{vst log. 1} [V]	2 až 5	2 až 5	3,5 až 5	2 až 5	3,5 až 5
U _{výst log. 0} [V]	0 až 0,4	0 až 0,4	0 až 0,5	0 až 0,3	0 až 0,5
U _{výst log. 1} [V]	2,4 až 5	2,7 až 5	4,5 až 5	4,5 až 5	4,5 až 5
N [ks]	10	20	600	6000	400
T _n [ns]	20	2 až 5	15	12	60
T _p [ns]	15	2 až 5	15	12	60

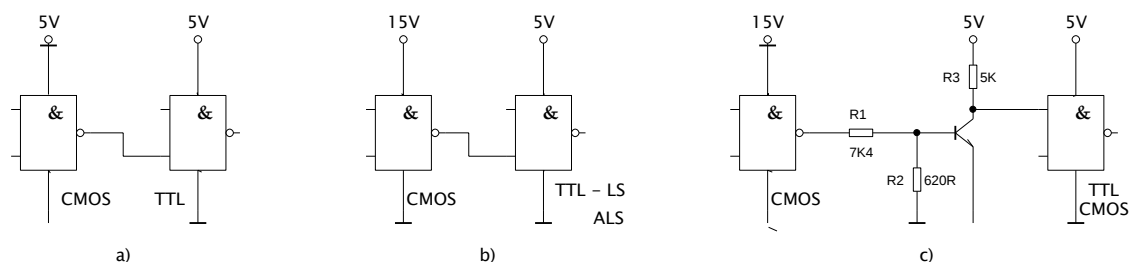
Tab. 3: Porovnání vlastností logických obvodů

2.3.4 Vzájemné spojování obvodů CMOS a TTL

Při vzájemném spojování obvodů CMOS s různým napájecím napětím nebo při spojení s TTL obvody musí být přizpůsobeny napěťové a proudové úrovně.

Spojení CMOS → TTL

Mohou nastat tři případy, které jsou znázorněny na obrázku 2.17.



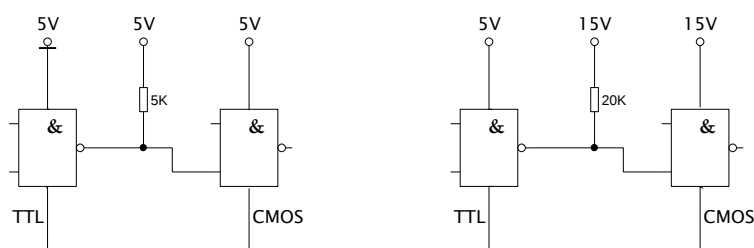
Obr. 2.17: Připojení CMOS → TTL

Vstupy obvodů standardní řady TTL74xx je možno přímo připojit k CMOS výstupům při napájení obvodů CMOS $U_{cc} = 5V$ (obr. 17a). Totéž platí i pro připojení CMOS $U_{cc} = 15V$ ke vstupům hradel 74LSxx a 74ALSxx, protože tyto obvody mají na vstupech omezovací Schottkyho diody (obr. 17b).

Jestliže budíme z výstupu CMOS obvodu napájeného vyšším napájecím napětím $U_{cc} = 15V$, obvody standardního TTL nebo obvody CMOS s nižším napájecím napětím, je nutné vysoké výstupní úrovně transformovat tranzistorovým stupněm (obr. 17c).

Spojení TTL → CMOS

Vstupy CMOS obvodů mohou být přímo připojeny na výstupy TTL, jak je patrné z obrázku č. 2.18. Výstupní úroveň hradla TTL je upravena pomocí tzv. Pull-up rezistoru.



Obr. 2.18: Připojení TTL → CMOS

2.4 Kombinační logické obvody

Kombinační logický obvod je realizován spojením základních logických členů tak, aby splňoval požadovanou logickou funkci. Výstupní hodnota kombinačních logických obvodů je určena výhradně okamžitou kombinací vstupních veličin, tj. neobsahuje žádné paměťové prvky.

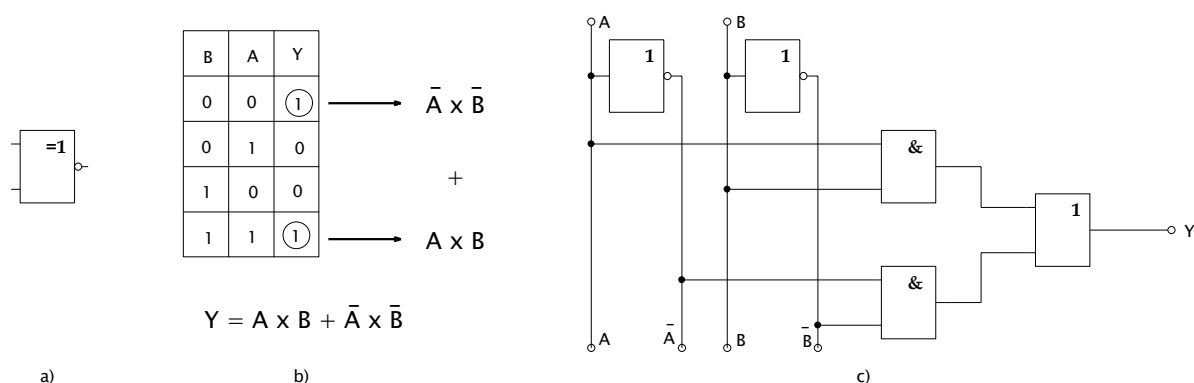
Mezi základní kombinační logické obvody patří:

- XOR neboli Exklusiv OR;
- Kodéry a dekodéry;
- Binární sčítačka;
- Generátor parity;
- Binární komparátor;
- Multiplexery a demultiplexery.

2.4.1 Návrh kombinačních logických obvodů

Na obvodu NOXOR si nyní ukážeme postup návrhu jednoduchého kombinačního logického obvodu.

Máme tedy navrhnout dvouvstupový obvod NOXOR (ekvivalence), jehož funkci jednoznačně popisuje pravdivostní tabulka na obr. 2.19.



Obr. 2.19: Návrh obvodu NOXOR

Schematická značka obvodu NOXOR je uvedena na obrázku 2.19a. Z pravdivostní tabulky (obr. 19b) je zřejmá funkce obvodu. Jsou-li na vstupu dvě stejné hodnoty (dvě nuly či jedničky),

bude na výstupu log. 1, v opačném případě log. 0 (při vstupní kombinaci nula a jedna nebo jedna a nula).

Při návrhu kombinačních logických obvodů nejprve označíme řádky (všechny s logickou jedničkou na výstupu – na obr. 2.19b zakroužkovanou) se kterými budeme pracovat. Vedle každého označeného řádku vypíšeme odpovídající funkci. Jak ji zjistíme? Řádky jsou jednoduše tvořeny součinem (logická funkce AND) vstupních parametrů s výsledkem jedna a to u logického součinu platí jen tehdy, když v našem případě oba dva vstupy mají hodnotu 1 (u AND $1 \times 1 = 1$). Není-li u vstupního stavu log. 1, musíme tyto parametry negovat (první řádek, kde $\overline{A}\overline{B}$ je opět 1). Výsledná funkce má potom podobu součtu jednotlivých řádků:

$$Y = AB + \overline{A}\overline{B}$$

Na obrázku 2.19c je zapojení celého obvodu. Pro praktické provedení je dále možné u složitějších funkcí minimalizovat, popřípadě upravit s využitím pravidel Booleovy algebry.

2.4.2 Základní kombinační logické obvody

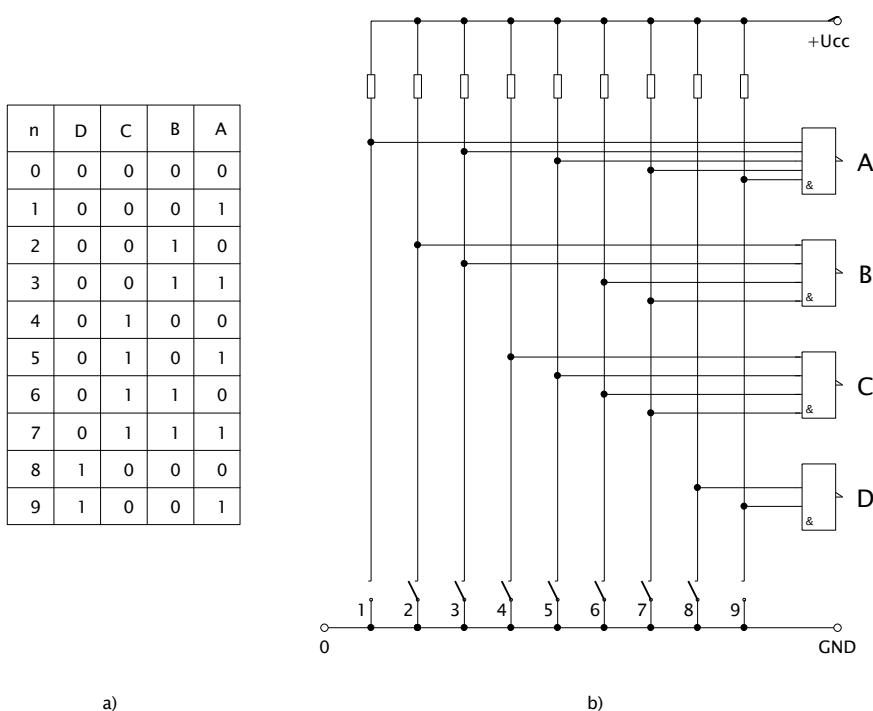
2.4.2.1 Kodéry a dekodéry

Jedním z důležitých oborů aplikace kombinačních logických členů je technika kódování a dekódování číslicových signálů. Elektrické obvody a zařízení, které pracují se signálem o dvou možných stavech, mohou využívat dvojkové číselné soustavy. V této soustavě je možno realizovat všechny potřebné aritmetické operace. Velikost dvojkových čísel je možno též jednoduše indikovat např. světelnými indikátory. Desítková číselná soustava je ovšem známější a je obecně používána. Proto je vhodné mít možnost vzájemně převádět čísla dvojkové a desítkové soustavy. K převodu mezi dvojkovými a desítkovými čísly používáme kódy. Kódem se rozumí pravidlo, podle něhož určité kombinaci nul a jednotek (nebo stavů L a H) přiřazujeme určité desítkové číslo. Byla vypracována celá řada takových kódů, z nichž každý má své výhody a nevýhody. Některé používané kódy jsou v tab. na obr. 2.20.

Kód des. číslo	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA	DCBA
	Dvojkový	BCD	N+3	Aiken	Grayův	7421	5421
0	0000	0000	0011	0000	0000	0000	0000
1	0001	0001	0100	0001	0001	0111	0001
2	0010	0010	0101	0010	0011	0110	0010
3	0011	0011	0110	0011	0010	0101	0011
4	0100	0100	0111	0100	0110	0100	0100
5	0101	0101	1000	1011	0111	1010	1000
6	0110	0110	1001	1100	0101	1001	1001
7	0111	0111	1010	1101	0100	1000	1010
8	1000	1000	1011	1110	1100	1111	1011
9	1001	1001	1100	1111	1101	1110	1100
10	1010						
11	1011						
12	1100						
13	1101						
14	1110						
15	1111						

Obr. 2.20: Tabulka kódování

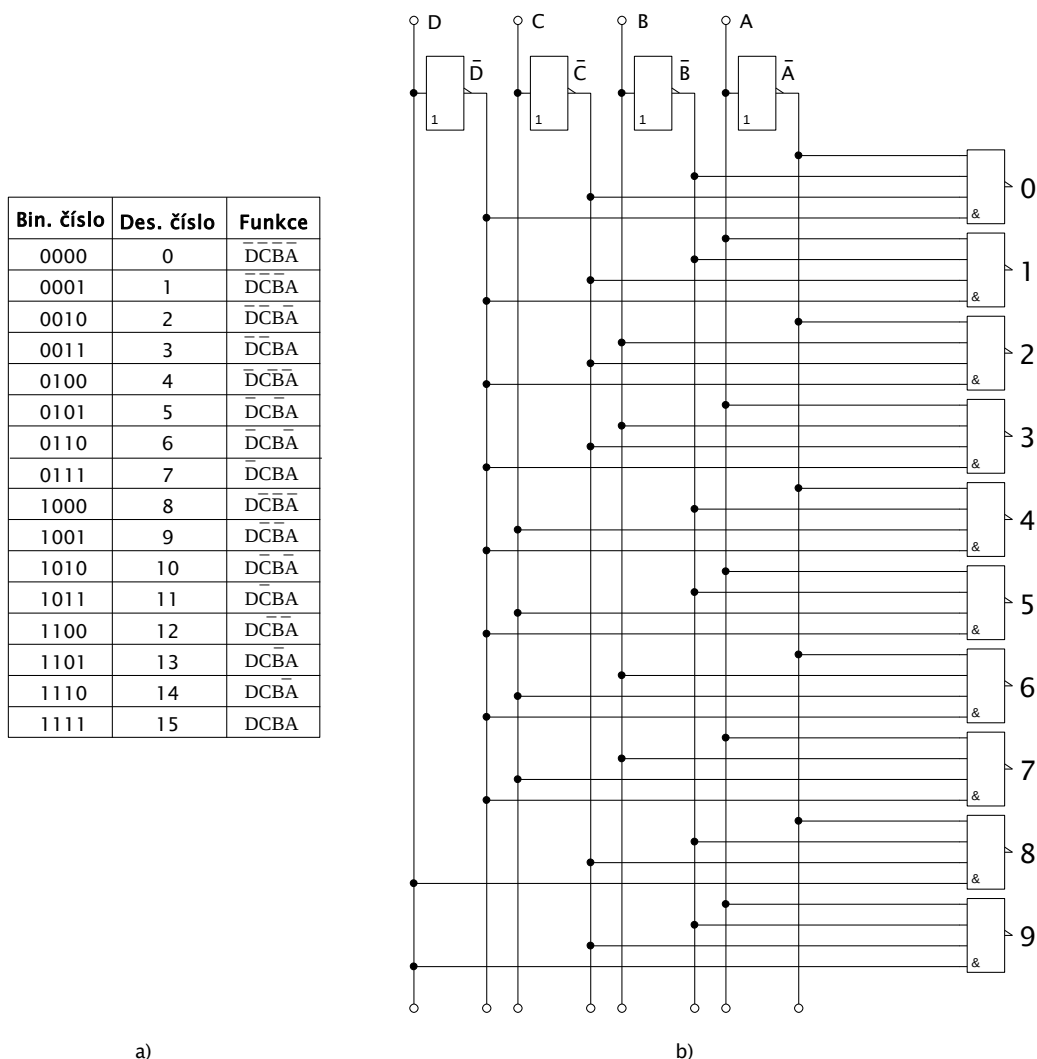
Jako příklad kódování uvažujme převod desítkových čísel 0 až 9 do kódu BCD 1248, tj. do dvojkového čísla o čtyřech bitech. Převodník můžeme realizovat čtyřmi logickými členy NAND, přičemž výstup každého z nich přísluší jednomu bitu dvojkového čísla. Vstupní obvod převodníku může být tvořen odpory, jimiž se jednotlivé vstupy logických členů udržují ve stavu H. Jednotlivá desítková čísla, která chceme převádět, můžeme volit spínači 0 až 9, jak je naznačeno na obr. 2.21b.



Obr. 2.21: Kodér 0 až 9 do kódu BCD

Sepnutím kteréhokoliv spínače se příslušná úroveň H změní v úroveň L. Logické členy NAND, jejichž vstupy takto nabyly úrovně L, budou mít na výstupu stav H. Podle téhož pravidla je možno realizovat převodníky pro libovolný kód. Místo spínačů lze použít obvody TTL, nebo jiné obvody s ekvivalentními logickými úrovněmi. Pravdivostní tabulka je na obrázku 2.21a.

Opačnou úlohou je převod dvojkového čísla v určitém kódu na číslo desítkové. Velmi často přichází v úvahu převod čísel v kódu BCD na čísla desítková. V podstatě se jedná o převod čísla o čtyřech bitech na jedno desítkové číslo z deseti možných.



Obr. 2.22: Dekodér jeden z deseti

Při sestavování odpovídajících logických vztahů vycházíme z pravdivostní tabulky (obr. 2.22a) a přihlédneme k tomu, že čtyřbitové číslo dává šestnáct možných kombinací, z nichž využíváme jen deseti. Pracujeme s kombinacemi nejvýše do čísla 9 = 1001. Z toho plynou

některá zjednodušení logických vztahů i bez řešení minimalizace. Číslo 9 můžeme např. vyjádřit jen součinem $A.D$, neboť kombinace $A.\bar{B}.C.D$, $A.B.\bar{C}.D$, $A.B.C.\bar{D}$ nepřicházejí v úvahu. Dostaneme pak tyto vztahy:

$$0 = \bar{A}.\bar{B}.\bar{C}.\bar{D} \quad 3 = A.B.\bar{C} \quad 6 = \bar{A}.B.C$$

$$1 = A.\bar{B}.\bar{C}.\bar{D} \quad 4 = \bar{A}.\bar{B}.C \quad 7 = A.B.C$$

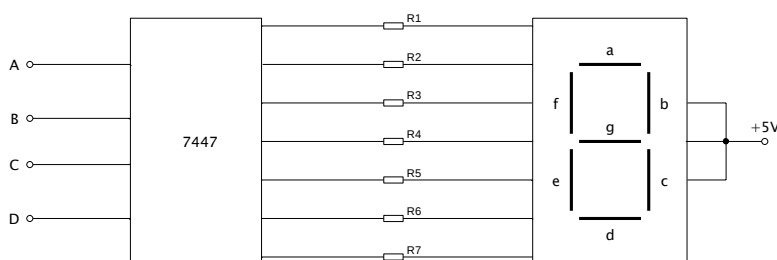
$$2 = \bar{A}.B.\bar{C} \quad 5 = A.\bar{B}.C \quad 8 = \bar{A}.D \quad 9 = A.D$$

Vztahy v této formě jsou logickými součiny, tj. odpovídají funkci AND. Často je však žádán aktivní výstup ve stavu L, pak musíme logické součiny negovat, tj. použít logické členy NAND. Zapojení dekodéru podle uvedených vztahů je na obr. 2.22b.

Protože jsme při návrhu dekodéru vyloučili kombinace proměnných odpovídající číslům větším než 9, nesmíme tato čísla na dekodér přivádět (přivedeme-li je, dá dekodér chybný výsledek).

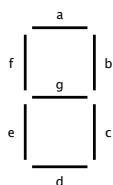
2.4.2.2 Dekodéry pro sedmissegmentové displeje

V některých případech je potřeba provádět vzájemné převody kódů. Na obr. 2.23 je uveden příklad převodu kódu BCD 1248 na desítkové číslo zobrazované sedmissegmentovým zobrazovačem pomocí integrovaného dekodéru SN 7447.



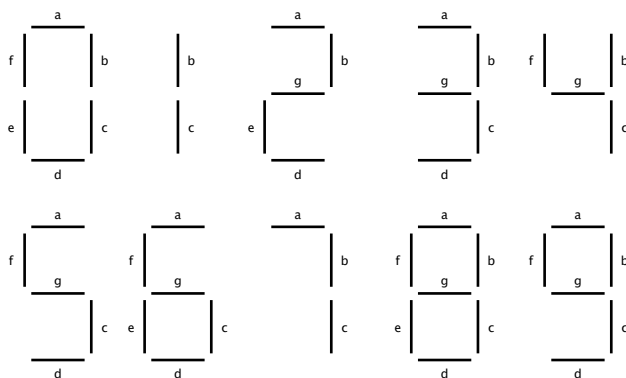
Obr. 2.23: Ovládání sedmissegmentového zobrazovače dekodérem SN 7446

Dekodér je určen pro buzení zobrazovacích jednotek tvořených sedmissegmentovými displeji. Jeho výstupní stavy rozsvěcují některé ze sedmi LED diod tak, aby kombinace rozsvícených LED diod tvořily dekadické číslice od 0 až 9. Jednotlivé segmenty jsou označeny písmeny a, b, c, d, e, f, g, h a g (obr. 2.24). Stejně jsou označeny i výstupy dekodéru.



Obr. 2.24: Sedmisegmentový zobrazovač

Požadovaná čísla rozsvítíme na sedmisegmentovém zobrazovači vhodnou kombinací výstupních hodnot dekodéru. Pokud nevybudíme segment g, bude na displeji svítit číslice 0., atd. Na obrázku 2.25 jsou znázorněny svítící segmenty pro jednotlivé stavy.



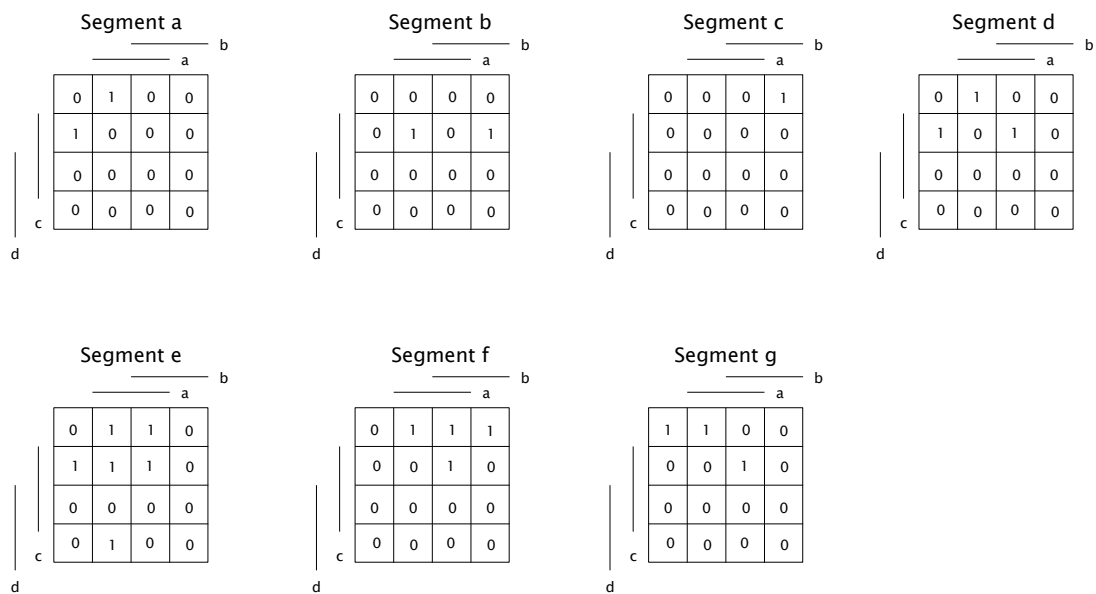
Obr. 2.25: Zobrazení čísel na sedmisegmentovém zobrazovači

Při návrhu dekodéru vycházíme z odpovídající pravdivostní tabulky, která je uvedena na obrázku č. 2.26. Jedná se o dekodér displeje se společnou anodou (k výstupu dekodéru jsou připojeny jednotlivé katody displeje), proto jsou svítící segmenty označeny logickou 0.

Dekadická hodnota	Vstupy				Značení segmentů						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

Obr. 2.26: Pravdivostní tabulka sedmisegmentového zobrazovače

Karnaughovy minimalizační mapy, které jsou zde uvedeny pro každý segment, slouží pro zjednodušení výsledné funkce. Pro každý segment sestavíme jednu Karnaughovu mapu, jak je patrné z obrázku č. 2.27.



Obr. 2.27: Karnaughovy mapy pro návrh sedmisegmentového zobrazovače

Po minimalizaci pro jednotlivé segmenty dostáváme:

$$Q_a = A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D}$$

$$Q_b = A\bar{B}C\bar{D} + \bar{A}B.C\bar{D}$$

$$Q_c = \bar{A}B.\bar{C}\bar{D}$$

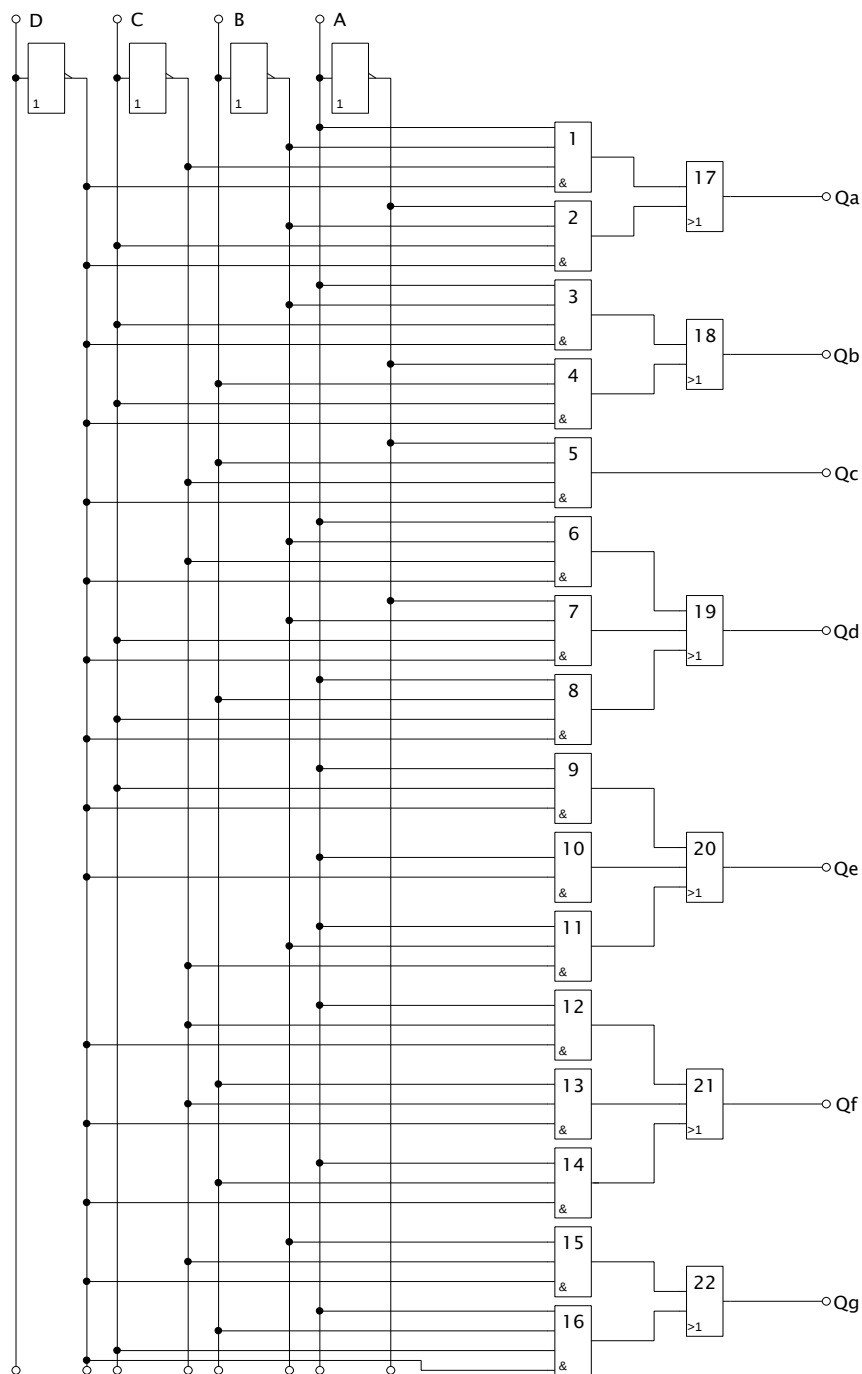
$$Q_d = \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A.B.C\bar{D}$$

$$Q_e = A\bar{D} + A\bar{B}\bar{C} + \bar{B}C.\bar{D}$$

$$Q_f = A\bar{C}.\bar{D} + AB.\bar{D} + B.\bar{C}.\bar{D}$$

$$Q_g = \bar{B}.\bar{C}.\bar{D} + A.B.C.\bar{D}$$

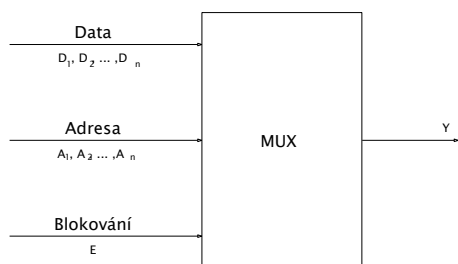
Příklad realizace sestavy z logických členů je na obr. 2.28.



Obr. 2.28: Realizace sedmisegmentového zobrazovače

2.4.2.3 Multiplexer

Multiplexer je kombinační logický obvod, který umožňuje přepínat z několika vstupních proměnných na jeden výstup podle hodnot řídicích signálů (obr. 2.29). Obsahuje n vstupů D_1 až D_n a jeden výstup Y . Volbu jednoho ze vstupů provádíme pomocí řídicích adresových signálů A_1 až A_n , písmenem E (enable) je vyznačen blokovací vstup.

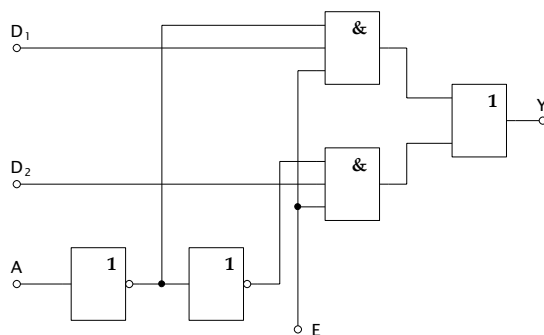


Obr. 2.29: Principiální schéma multiplexeru

Ukázka realizace dvouvstupového multiplexeru je na obrázku 2.30b. Vstupní data jsou součinným hradlem zadržena nebo propuštěna na vstup součtového členu a odtud na výstup. Přenos přes součinný člen je ovládán adresovým vstupem A a blokovacím vstupem E.

E	D ₁	D ₂	A	Y
1	0	X	0	0
1	1	X	0	1
1	X	0	1	0
1	X	1	1	1
0	X	X	X	0

a)



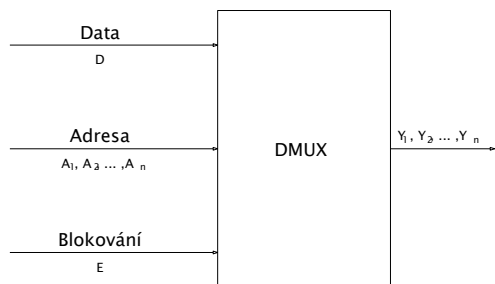
b)

Obr. 2.30: Schéma zapojení a pravdivostní tabulka a dvouvstupového multiplexeru

Pro tento multiplexer můžeme sestavit pravdivostní tabulku na obr. 2.30a. Můžeme ji ale zjednodušit, když na místě logických proměnných, jejichž stav nemá vliv na chování obvodu, napíšeme symbol neurčitého stavu x.

2.4.2.4 Demultiplexer

Demultiplexer plní přesně opačnou funkci než multiplexer. Jedná se tedy o přepínací obvod, který jeden vstup přepíná na několik výstupů dle řídicího signálu (z dekodéru).

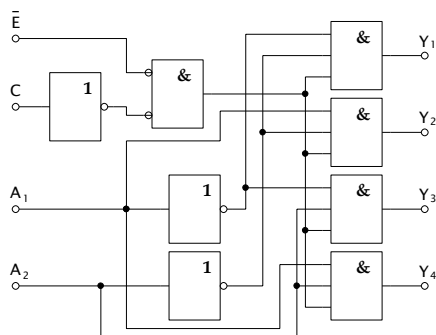


Obr. 2.31: Principiální schéma demultiplexeru

Jako příklad uveďme zapojení na obrázku 2.32b a pravdivostní tabulku 2.32a čtyřvýstupového demultiplexeru, který je dodáván v integrované podobě pod označením SN 74155.

$\bar{E}$	C	A_1	A_2	Y_1	Y_2	Y_3	Y_4
1	X	X	X	1	1	1	1
0	1	0	0	0	1	1	1
0	1	1	0	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0
X	0	X	X	1	1	1	1

a)

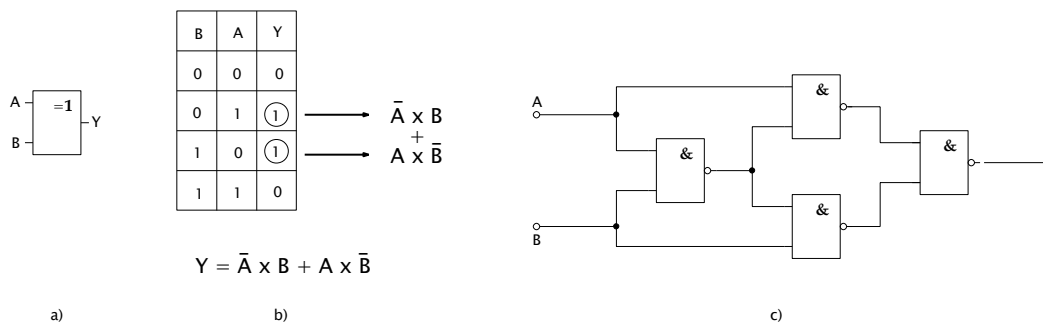


b)

Obr. 2.32: Schéma zapojení a pravdivostní tabulka čtyřvýstupového demultiplexoru

2.4.2.5 Komparátor

Komparátor je kombinační logický obvod, který porovnává dvě binární čísla a zároveň generuje výstupní signál o jejich rovnosti či nerovnosti. Obvod je též označován jako Exclusive OR nebo též XOR.

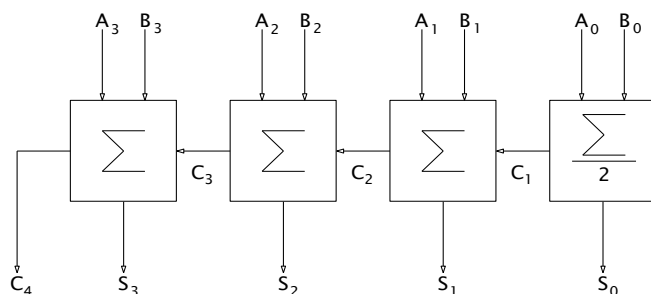


Obr. 2.33: Schematická značka, zapojení a pravdivostní tabulka Exklusivní OR

Na obr. 2.33c je logický obvod sestavený z hradel NAND, na obr. 2.33b je pravdivostní tabulka.

2.4.2.6 Binární sčítačka

Dalším typickým kombinačním obvodem je binární sčítačka (obr. 2.34). Sečtení dvou binárních čísel je založeno na sečtení všech dvojic dvojkových míst, které si vzájemně řádově odpovídají počínaje nejnižšími řády. Přitom je potřeba brát v úvahu i přenosy z nižších do vyšších řádů, které během čítání vznikají.



Obr. 2.34: Schematická značka, zapojení a pravdivostní tabulka Exklusivní OR

Pro ilustraci uvedme součet binárních čísel:

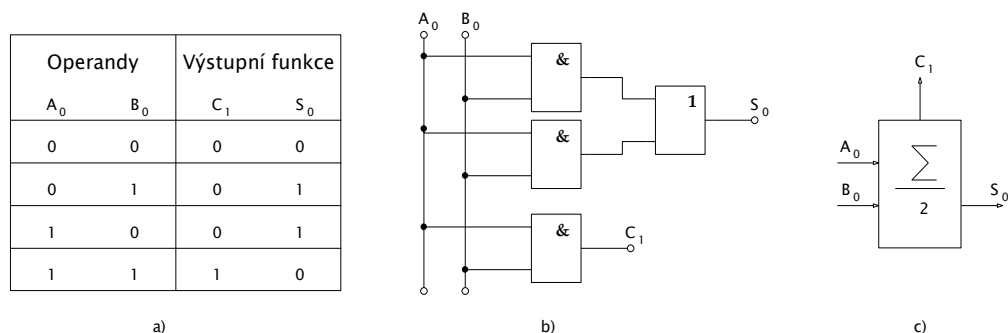
0	0	1	1
0	1	0	1
---	---	---	---
0	1	1	10

V posledním případě vznikl přenos z řádu 2^0 do řádu 2^1 .

Součet dvou čtyřbitových čísel A a B můžeme znázornit pomocí blokového schématu na obr. 2.34. Tři nejvyšší řády jsou sčítány v blocích označených Σ , které se nazývají úplné sčítačky, pro nejnižší řád se používá tzv. poloviční sčítačka se znakem $\Sigma/2$.

Poloviční sčítačka

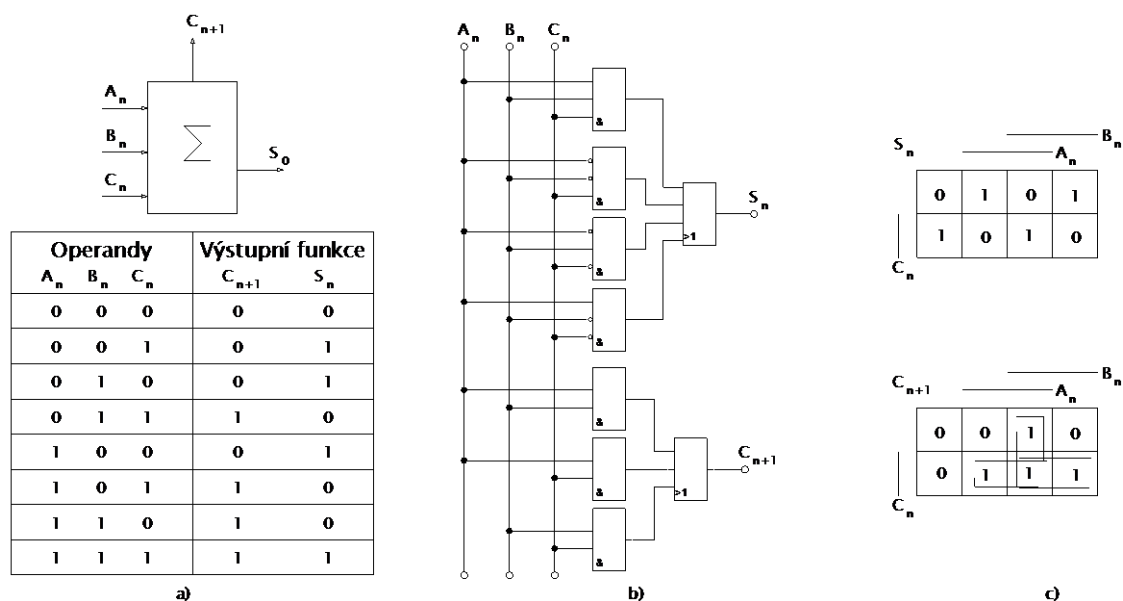
Na obr. 2.35c je logický obvod, který realizuje funkci poloviční sčítačky. Má dva vstupy A_0 a B_0 a dva výstupy. Prvním z výstupů je součet S_0 , druhý výstup znázorňuje přenos do vyššího řádu C_1 (carry).



Obr. 2.35: Pravdivostní tabulka, zapojení a schematická značka poloviční sčítačky

Úplná sčítačka

S použitím poloviční sčítačky můžeme zvládnout sečtení nejnižšího bitu dvou binárních čísel. Pro sečtení vyšších bitů se používá plná sčítačka, která na rozdíl od poloviční sčítačky ke dvěma sčítaným číslicím A_n a B_n přidává další vstup a tím je přenos z nižšího řádového místa C_n viz obr. 2.36. Součet binárních hodnot $A_n + B_n + C_n$ je označen S_n , a přenos do vyššího řádu C_{n+1} může nabývat hodnot dle pravdivostní tabulky na obr. 2.36a.

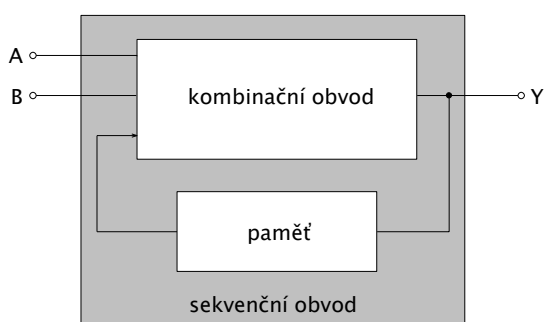


Obr. 2.36: Úplná sčítačka realizovaná minimalizací

V pravdivostní tabulce si můžeme všimnout, že jestliže jedna z proměnných A_n , B_n , C_n nabývá hodnoty 1, potom $S_n = 1$ a $C_{n+1} = 0$. Jsou-li mezi vstupními proměnnými dvě jedničky, dochází k přenosu do vyššího řádu, tedy $C_{n+1} = 1$ a zároveň $S_n = 0$. Pouze v posledním řádku, kdy mají všechny vstupní hodnoty úroveň 1, je součet $S_n = 1$ a zároveň dochází i k přenosu do vyššího řádu.

2.5 Sekvenční logické obvody

Stav výstupu sekvenčních logických členů a obvodů závisí nejen na kombinaci vstupních proměnných (tak jak tomu bylo u kombinačních obvodů), ale i na stavu výstupu při předchozí kombinaci vstupních proměnných. Podstatnou částí sekvenčních členů nebo obvodů je paměťový člen, který zabezpečuje žádoucí uchování informace, jak je patrné z následujícího obrázku č. 2.37.



Obr. 2.37: Schéma sekvenčního logického obvodu

Mezi základní sekvenční logické obvody patří:

- klopné obvody RS, RST a D;
- dvojité klopné obvody RST a JK;
- čítače impulsů a děliče frekvence;
- posuvné registry.

2.5.1 Klopné obvody

2.5.1.1 Klopný obvod R-S

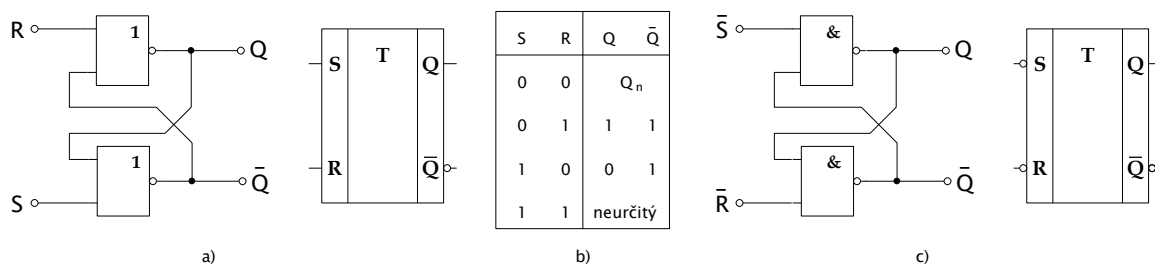
Obvod na obrázku č. 2.38a představuje jednoduchý klopný obvod. Má dva výstupy Q a $\bar{Q}$, jejichž stav je vzájemně opačný. Stav, při němž je na výstupech $Q = H$ a $\bar{Q} = L$, označujeme jako stav H (High) klopného obvodu. Tento stav vyjadřujeme i označením „klopný obvod je nastaven“. Stav výstupů $Q = L$ a $\bar{Q} = H$ označujeme jako stav L (Low) klopného obvodu. Tento stav vyjadřujeme i označením „klopný obvod je nulován“.

Stav klopného obvodu je řízen dvěma vstupy. Ten vstup, na němž působící úroveň H a uvede klopný obvod do stavu H, se označuje vstup S (Set – nastaven). Ten vstup, na němž působící úroveň H a uvede klopný obvod do stavu L, se označuje jako vstup R (Reset).

Byl-li na začátku klopný obvod ve stavu H a přijde-li úroveň H na vstup S, stav obvodu se nezmění. Podobně byl-li na počátku klopný obvod ve stavu L a přijde-li na vstup R úroveň H, zůstane stav obvodu nezměněn. Počáteční stav obvodu zůstane zachován také tehdy, přijde-li na vstupy R a S současně úroveň L. Vidíme tedy, že výsledný stav klopného obvodu záleží nejen na úrovních přivedených na oba vstupy, ale i na předchozím stavu obvodu.

Zvláštní případ nastane, přivedeme-li na vstupy R a S současně úroveň H. Po dobu působení těchto úrovní budou mít výstupy obou logických členů úroveň L. Odstraníme-li úroveň H ze vstupů, přejde klopný obvod náhodně do jednoho z obou stavů. Výsledný stav klopného obvodu bude tedy neurčitý. Stav, při němž na vstupy R a S působí současně úroveň H, nemůžeme pro účely logického rozhodování používat.

Činnost uvedeného klopného obvodu můžeme popsat pravdivostní tabulkou, která je na obr 2.38b. V tabulce znamená Q_n počáteční stav klopného obvodu.



Obr. 2.38: Zapojení klopného obvodu R-S, pravdivostní tabulka a schem. značka

Všimněme si nyní klopného obvodu sestaveného analogicky z logických členů NAND, je znázorněn na obr. 2.38c. Stav obvodu zde nelze řídit úrovní H na vstupech. Na druhý vstup některého z obou členů bude totiž vždy působit úroveň L z jednoho výstupu. Klopný obvod R-S sestavený z logických členů NAND lze řídit jen úrovněmi L na vstupech. Abychom vyhověli definici vstupů R a S, která byla uvedena, musíme opačnou řídící úroveň respektovat v označení vstupů. Klopný obvod toho druhu se označuje jako klopný obvod R-S s negovanými (také s nepřímými) vstupy. Značka negace zde vyjadřuje symbolicky invertory. Kdybychom do obou vstupů zařadili invertory, bylo by možno klopný obvod ovládat úrovní H stejně, jako v případě podle obr. 2.38a.

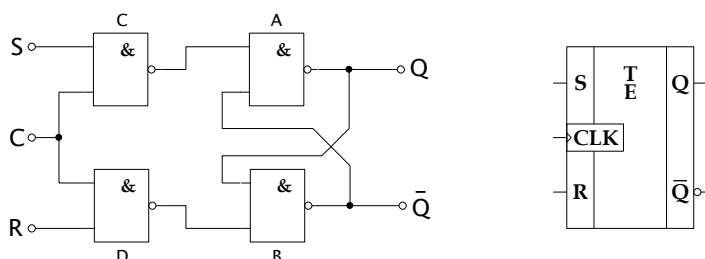
2.5.1.2 Klopný obvod R-S-T

Klopné obvody R-S, které jsme popsali, náleží mezi klopné obvody nesynchronizované. Jejich činnost může být nepříznivě ovlivňována rušivými signály. Dostanou-li se na vstup R a S náhodné špičky napětí, může se snadno nežádoucím způsobem změnit stav klopného obvodu. Větší odolnost vůči rušení tohoto druhu mají klopné obvody synchronizované. U nich se stav klopného obvodu působením vnějšího rušení může změnit jen v době synchronizace. Proto se v praxi častěji využívá synchronizovaných klopných obvodů.

K synchronizaci slouží synchronizační impulsy, označované také jako impulsy taktovací nebo impulsy hodinové. Tyto impulsy se získávají ze zvláštního zdroje impulsů, který je součástí číslicového zařízení. Přivádějí se na synchronizační (hodinový) vstup synchronizovaných obvodů. Na ostatní, tzv. informační vstupy synchronizovaných obvodů se

nejprve přivedou dané logické úrovně. Obvod se pak uvede do žádané činnosti až působením synchronizačního impulsu.

Klopný obvod R-S který je opatřen obvodou umožňujícími jeho synchronizaci hodinovými impulsy, bývá označován jako klopný obvod R-S-T (T od slova takt). Zapojení takového obvodu je na obr. 2.39. Vlastnímu klopnému obvodu R-S je předřazen řídicí obvod sestavený ze dvou logických členů NAND. Synchronizační vstup je označen C (od slova clock - hodiny).



Obr. 2.39: Zapojení klopného obvodu R-S-T a schematická značka

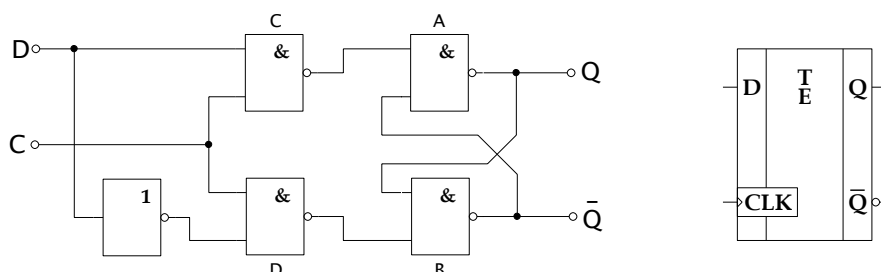
Je-li na vstupu S úroveň H a přijde-li na vstup C úroveň H, bude výstup členu C na úrovni L. Na výstupu Q bude úroveň H a klopný obvod přejde do stavu H. Přejde-li nyní vstup C do úrovně L (ukončí se hodinový impuls), bude na výstupech obou členů C a D stav H. Původní stav klopného obvodu se tím však nezmění. Je-li na vstupu R úroveň H a přijde-li na vstup C úroveň H, bude na výstupu členu D úroveň L.

Výstup $\bar{Q}$ přejde na úroveň H a klopný obvod bude ve stavu L. Po ukončení hodinového impulsu, tj. přijde-li vstup C na úroveň L, budou výstupy členů C a D ve stavu H. Stav klopného obvodu se tím opět nezmění. Stav klopného obvodu se mění v obou případech s příchodem hodinového impulsu, tj. v okamžiku, kdy tento impuls nabude úrovně H.

Bude-li na vstupech R a S současně úroveň L, bude na výstupech členů C a D bez ohledu na hodinový impuls úroveň H a předchozí stav klopného obvodu se s hodinovým impulsem nezmění. Bude-li na obou vstupech R a S současně úroveň H a přejde-li vstup C na úroveň H, bude na obou výstupech Q a $\bar{Q}$ úroveň H. S ukončením hodinového impulsu obvod ustálí v neurčitěm stavu. Stav klopného obvodu se tedy mění úrovní H na vstupu R nebo S s hodinovým impulsem.

2.5.1.3 Klopný obvod D

U tohoto obvodu je spojen vstup R se vstupem S prostřednictvím invertoru. Vstupy R a S mají tedy vždy opačnou úroveň. Vzniká tak jediný vstup (kromě hodinového), označovaný jako vstup D (od slova data). Zapojení včetně schematické značky je na obr. 2.40.

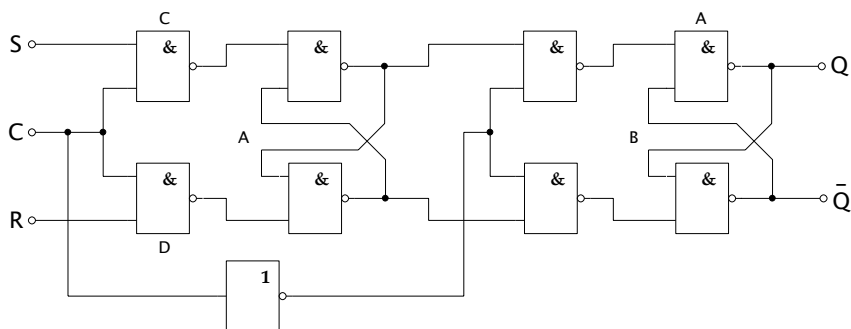


Obr. 2.40: Zapojení klopného obvodu D a schematická značka

Funkci obvodu je možno popsat takto: je-li před příchodem a v době trvání hodinového impulsu vstup D na úrovni H, bude po ukončení hodinového impulsu klopný obvod ve stavu H. Je-li před příchodem a v době trvání hodinového impulsu D na úrovni L, bude klopný obvod po ukončení hodinového impulsu ve stavu L. Informace přítomná na vstupu D se tedy s příchodem hodinového impulsu přesouvá na výstup.

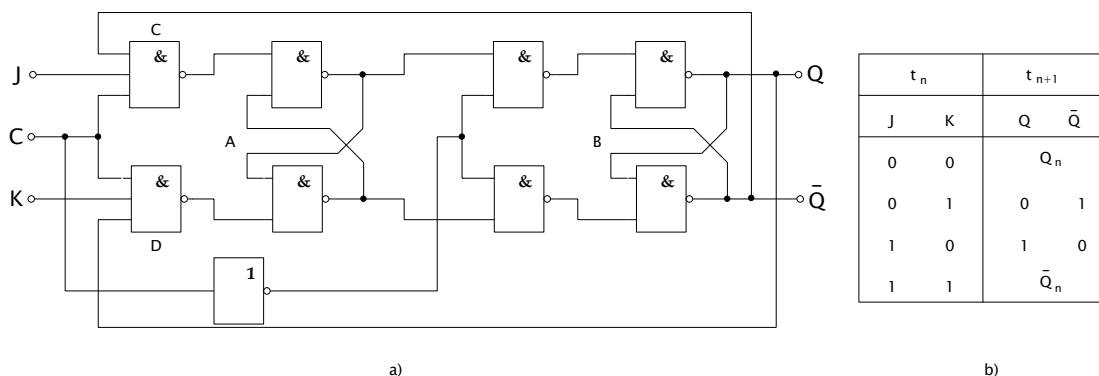
2.5.1.4 Dvojitý klopný obvod R-S-T a J-K

Příklad dvojitýho klopného obvodu R-S-T je na obr. 2.41. Je složen ze dvou klopných obvodů R-S-T a z jednoho invertoru. Je-li na vstupu C úroveň L, nemůže být stav klopného obvodu A ovlivněn signálem na vstupech R a S. Přechází-li vstup C na úroveň H, tj. přišel-li hodinový impuls, způsobí signál na vstupech R a S příslušnou změnu stavu klopného obvodu A. Na druhý klopný obvod B se však signál nedostane, protože jeho hodinový vstup je působením invertoru ve stavu L. Teprve s ukončením hodinového impulsu přejde vstup C klopného obvodu B na úroveň H a stav výstupu klopného obvodu A se přenesou do klopného obvodu B. Současně se klopný obvod A izoluje od dalších signálů na vstupech R a S. První klopný obvod (A) se označuje jako řídicí (master – pán), druhý klopný obvod (B) je řízený (slave – otrok). Tento typ klopných obvodů se proto též označuje pojmem "master - slave".



Obr. 2.41: Zapojení dvojitého klopného obvodu R-S-T

V popsaném dvojitém klopném obvodu jsou použity obvody R-S-T, jejichž výstupy mohou být v neurčitěm stavu při nevhodné kombinaci signálů na vstupech. Vyloučení těchto neurčitých stavů lze dosáhnout zavedením zpětné vazby z výstupů na vstupy klopného obvodu (obr. č. 2.42). Takto uspořádaný klopný obvod se označuje jako klopný obvod J-K.



Obr. 2.42: Zapojení dvojitého klopného obvodu J-K

Předpokládejme, že klopný obvod J-K byl na počátku ve stavu H a že na vstupech J a K jsou úrovně H. Jeden vstup logického členu C má logickou úroveň L od výstupu $\bar{Q}$, jeho výstup je tedy na úrovni H. Člen D má na dvou svých vstupech úrovně H. S příchodem hodinového impulsu budou všechny vstupy členu D na úrovni H a jeho výstup tedy bude na úrovni L. Řídící klopný obvod tím přejde do stavu L. S týlem hodinového impulsu se analogicky změní stav řízeného klopného obvodu a celý klopný obvod přejde do stavu L. Předpokládejme, že na vstupech J a K zůstanou nadále úrovně H. Protože se změnil stav výstupu, bude mít nyní jeden vstup členu D úroveň L od výstupu Q. Dva vstupy členu C budou mít úrovně H. S příchodem hodinového impulsu bude mít úroveň H i třetí vstup členu C a jeho výstup nabude úrovně L. Vstupní členy NAND C a D si tedy vyměnily úlohu. Řídící klopný obvod přejde do stavu H. S týlem hodinového impulsu pak do téhož stavu přejde celý klopný obvod. Z uvedeného je

zřejmé, že je-li na vstupech J a K klopného obvodu J-K trvale úroveň H, bude se s týlovou hranou každého hodinového impulsu měnit stav klopného obvodu v opačný.

Obdobným postupem můžeme vysledovat funkci klopného obvodu J-K pro jiné kombinace signálů na vstupech.

Činnost klopného obvodu J-K můžeme jednoduše popsat pravdivostní tabulkou (obr. 2.42b). V tabulce je t_n stav před příchodem hodinového impulsu, t_{n+1} je stav po ukončení hodinového impulsu, Q_n je stav výstupu před příchodem hodinového impulsu.

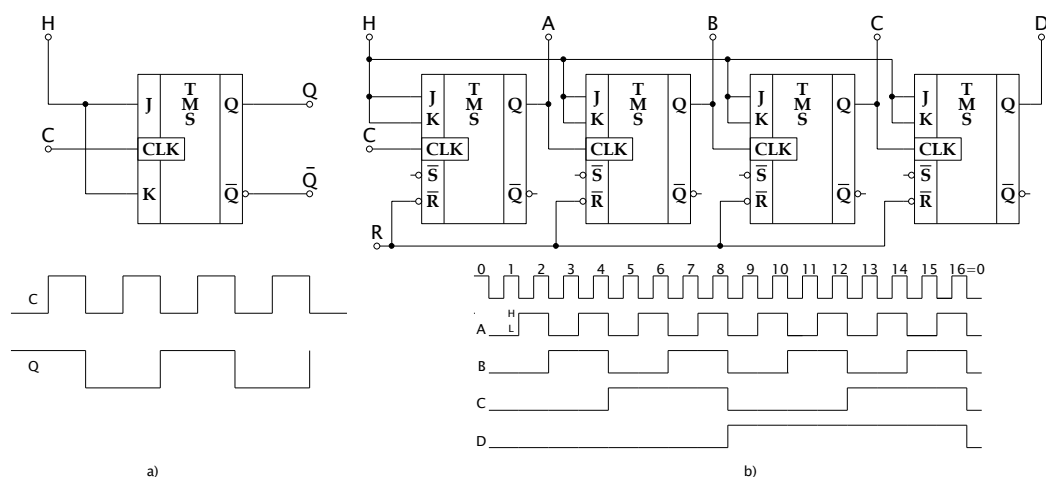
2.5.2 Čítače

Mnohá logická rozhodnutí jsou založena na vyhodnocení počtu opakujících se jevů. Každé opakování jevů může být vyjádřeno elektrickými impulsy. Počet opakování pak odpovídá počtu elektrických impulsů. Informace o počtu impulsů přitom musí být k dispozici v takové formě, jakou je možno dále zpracovávat logickými obvody. Máme-li vytvořeny vhodné elektrické impulsy, je třeba nějakým způsobem je spočítat. Tuto úlohu řeší čítače.

Čítač je obecně zařízení, které v nějakém kódu počítá elektrické impulsy přivedené na vstup čítače. Stav čítače, který odpovídá počtu impulsů, je pak možno dekodovat kombinačními obvody. Výsledkem dekodování může být např. povel k realizaci následné operace.

2.5.2.1 Dvojkové čítače asynchronní

Má-li bistabilní klopný obvod J-K na vstupech J a K současně úroveň H, bude se jeho stav měnit s ukončením každého hodinového impulsu. To lze vyjádřit časovým diagramem na obr. 2.43a. Klopný obvod v uvažovaném zapojení se označuje jako počítací klopný obvod. Na obr. 2.43b je naznačeno spojení čtyř počítacích obvodů tím způsobem, že výstup Q předchozího je spojen s hodinovým vstupem C obvodu následujícího, činnost takového zapojení můžeme popsat časovým diagramem (obr. 2.43b), v němž jsou zakresleny změny stavu jednotlivých klopných obvodů, jak k nim dochází s hodinovými impulsy přiváděnými na vstup prvního obvodu.

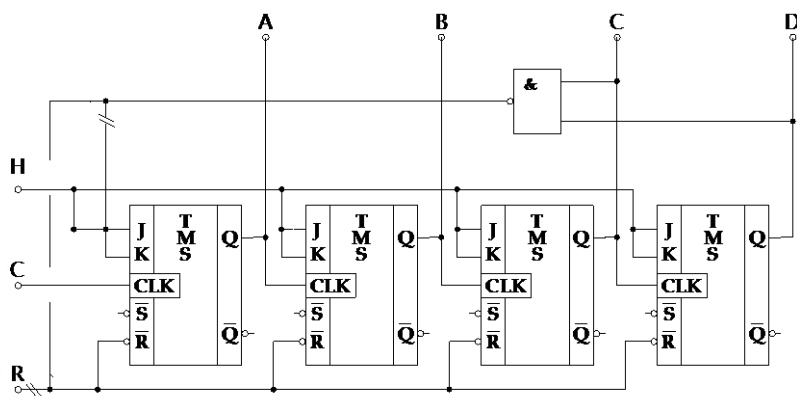


Obr. 2.43: Zapojení asynchronního čítače

Obvod sestavený uvedeným způsobem je dvojkový čítač. Počet impulsů přiváděných na vstup je na výstupech vyjádřen ve dvojkovém kódu. Čítač lze dalšími klopnými obvody rozšířit do libovolné délky. Tento čítač mění svůj stav s přibývajícími hodinovými impulsy směrem k větším hodnotám dvojkového čísla v čítači obsaženém. Je to tedy čítač vpřed. Změna stavů jednotlivých klopných obvodů probíhá postupně. Stav klopného obvodu příslušného vyššího bitu se změní až po změnách stavů klopných obvodů příslušejících nižším bitům. Klopné obvody tedy nepracují synchronně. Čítače tohoto typu jsou čítače asynchronní.

Z uvedených časových diagramů je patrná další užitečná vlastnost čítače. Každý klopný obvod produkuje impulsy o kmitočtu, který je polovinou kmitočtu impulsů přiváděných na jeho hodinový vstup. Vycházíme-li z kmitočtu impulsů přiváděných na vstup celého obvodu, dělí jej první klopný obvod dvěma, druhý čtyřmi, třetí osmi a čtvrtý šestnácti. Čítač se tedy uplatňuje také jako dělič kmitočtu. V této oblasti má řadu použití i mimo obor číslicové techniky.

Obsah čítače můžeme dekódovat kombinačním logickým členem. Můžeme využít výstupů Q nebo $\bar{Q}$ podle potřeby a podle použitého kombinačního členu. Přitom není nutno brát vždy v úvahu každý klopný obvod. Chceme-li např. dekódovat stav čítače odpovídající číslu 1100 (hodnota 12), postačí uvažovat výstupy C a D. Jednou z operací, která může následovat po dekódování obsahu čítače, je zastavení čítače. Jelikož jde o asynchronní čítač, zatavíme jej tím, že znemožníme prvnímu klopnému obvodu další změny stavu, tj. uvedeme-li např. jeho vstupy J a K na úroveň L. Způsob je ukázán na obr. 2.44. U daného čítače je logickým členem NAND dekódován stav čítače odpovídající číslu 1100.

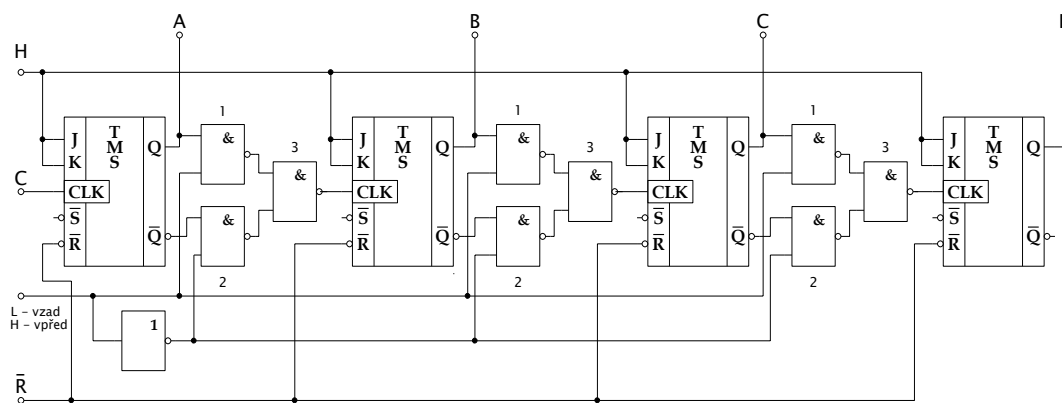


Obr. 2.44: Zkrácení čítacího cyklu asynchronního čítače

Pokud není tohoto čísla dosaženo, je některý vstup členu NAND na úrovni L a výstup členu spolu se vstupy J a K prvního obvodu jsou na úrovni H. Jakmile se dosáhne čísla 1100, budou všechny vstupy NAND na úrovni H. Jeho výstup se vstupy J a K klopného obvodu přejdou na úroveň L a čítač se zastaví. Činnost čítače je možno obnovit vynulováním čítače impulsem úrovně L, přivedeným na vstup „nulování“. Podobně lze čítač zastavit při jiných jeho stavech. Zastavit chod čítače lze i zablokováním přístupu vstupních impulsů kombinačním členem.

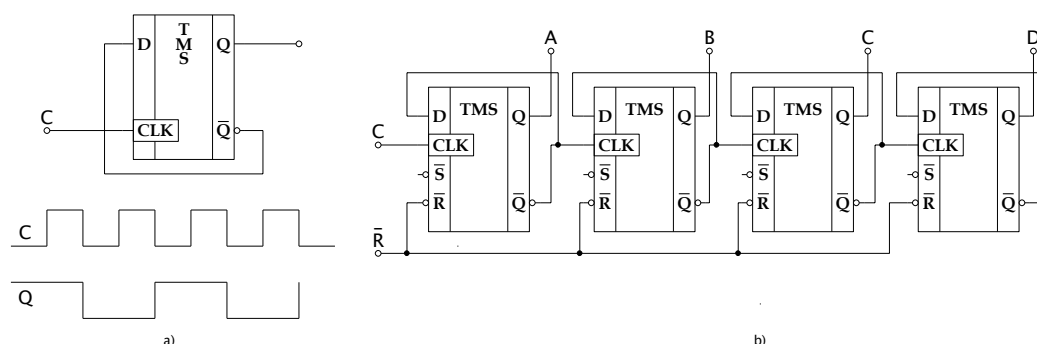
Jinou následnou operací může být zkrácení početního cyklu čítače. Znamená to, že čítač nevyčerpá své plné početní pořadí. Po dosažení určitého čísla se vynuluje a počítá automaticky znovu do tohoto čísla. Pro zkrácení početního cyklu čítače můžeme použít vstup „nulování“. Příklad je na obr. 2.44 naznačen čárkovaně. Výstup logického členu je připojen na vstup „nulování“ čítače. Dostane-li se čítač do stavu, kdy je na výstupech C a D úroveň H přejde výstup členu NAND do stavu L, čímž se všechny klopné obvody vynulují. Od dalšího hodinového impulsu počíná početní cyklus znovu. Nulováno je jen potud, pokud trvá úroveň H na výstupech C a D.

Variantní změna asynchronního dvojkového čítače vpřed v čítač vzad spočívá jen v přepojení hodinových vstupů druhého až čtvrtého klopného obvodu z výstupu Q na výstup $\bar{Q}$ obvodů předchozích. Obvody lze přepojit i s použitím kombinačních obvodů. Tak vznikne dvojkový čítač obousměrný, v němž je možno povelom na řídicím vstupu změnit směr počítání. Příklad je na obr. 2.45. Je-li na řídicím vstupu úroveň H, prochází signál z výstupů Q členem 1 a členem 3 (dvojnásobná negace) na vstup C následujících obvodů a čítač počítá vpřed. Je-li na řídicím vstupu úroveň L, prochází signál z výstupů $\bar{Q}$ členem 2 a 3 na vstupy C a čítač počítá vzad.



Obr. 2.45: Asynchronní čítač s přepínáním vpřed - vzad

Až dosud jsme uvažovali počítací klopný obvod vycházející z klopného obvodu J-K. Počítací klopný obvod je však možno získat také z klopného obvodu D řízeného změnou stavu hodinového vstupu. Zapojení a časový diagram jsou na obr. 2.46a.



Obr. 2.46: Asynchronní čítač realizovaný pomocí klopných obvodů D

Vstup D je spojen a výstupem $\bar{Q}$. Potom je před příchodem hodinového impulsu na vstupu D vždy opačný stav, než jaký je na výstupu Q. S příchodem hodinového impulsu se stav klopného obvodu změní v opačný. Je zřejmé, že s počítacími klopnými obvody typu D lze realizovat dvojkové čítače stejně dobře jako s obvody J-K. Protože stav klopného obvodu D se mění při změně stavu hodinového vstupu z úrovně L a H je třeba jednotlivé klopné obvody v čítači propojovat tak, aby byl výstup $\bar{Q}$ vždy spojen s hodinovým vstupem následujícího obvodu. Takové zapojení čítače vpřed je na obr. 2.46b. Změna zapojení na čítač vzad spočívá jen v přepojení hodinových vstupů druhého a čtvrtého klopného obvodu z výstupů $\bar{Q}$ na výstupy Q obvodů předchozích.

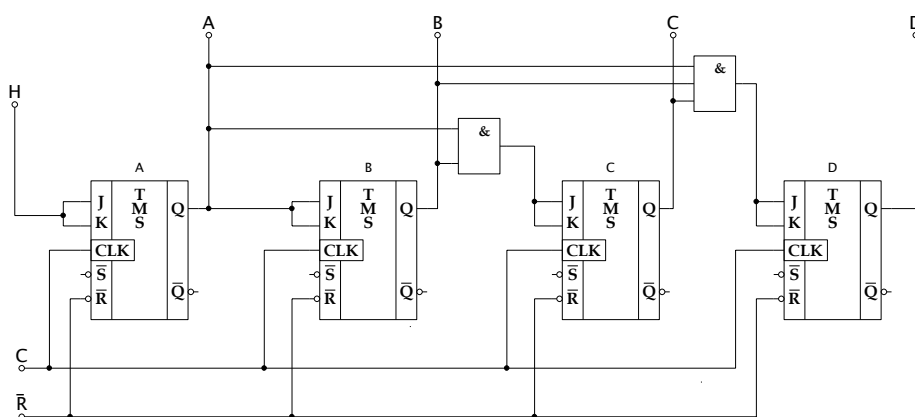
Všechny dříve uvedené operace založené na dekódování obsahu čítače lze u čítačů složených z D klopných obvodů realizovat obdobně, jak bylo ukázáno. Jen je třeba mít na paměti, že klopné obvody mění svůj stav a čelem hodinového impulsu.

2.5.2.2 Dvojkové čítače synchronní

V asynchronním čítači postupuje informace od nejméně významného bitu postupně k bitům významnějším. Např. přechází-li čítač z čísla 0111 (7) na číslo 1000 (8), musí se nejprve změnit stavy klopných obvodů příslušných bitů 0, 1 a 2. Instrukci ke změně stavu dostává klopný obvod od předchozího klopného obvodu. Čítač lze sestavit také tak, že se stav všech klopných obvodů, které přísluší určitému číslu početního pořadí čítače, změní současně.

V takovém uspořádání musí být hodinové vstupy všech klopných obvodů spojeny paralelně a jsou vstupem čítače. Aby však bylo dosaženo změny stavu jednotlivých klopných obvodů jen tehdy, kdy je to žádoucí, tj. v souladu s početním pořadím čítače, musí být některé změny stavu blokovány. K tomu lze použít kombinační logické členy.

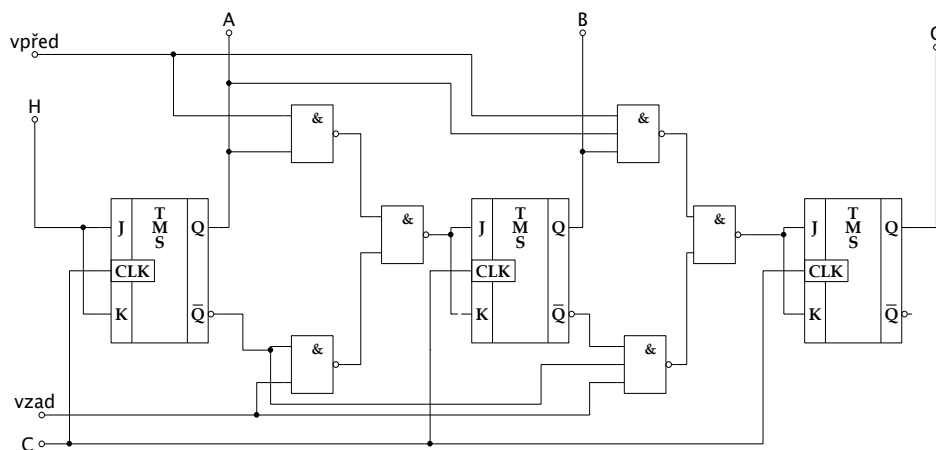
Všimněme si nyní zapojení na obr. 2.47. Je to dvojkový čítač vpřed, sestavený podle naznačeného principu. Jeho činnost je také určena časovým diagramem na obr. 2.43b. Obecně řečeno, každý klopný obvod čítače mění svůj stav tehdy, jsou-li všechny klopné obvody příslušející méně významným bitům ve stavu H. Jakmile je tato podmínka splněna, mění jednotlivé klopné obvody s hodinovým impulsem své stavy současně, tj. synchronně, čítače tohoto typu jsou čítače synchronní.



Obr. 2.47: Synchronní čítač realizovaný pomocí klopných obvodů J-K

Tento druh čítačů můžeme výhodně sestavovat z klopných obvodů MH7442, které mají po třech vstupech J a K, ve funkci logického součinu. Vstupy J a K lze tedy realizovat potřebné součinnové funkce a samostatné logické členy, naznačené na obr. 2.47, již není nutno používat. Kdybychom však použili klopné obvody D typu, byly by dodatečné logické členy nezbytné.

Na stejném principu jako čítač vpřed může být realizován čítač vzad a čítač obousměrný. Obousměrný synchronní čítač může být sestaven se dvěmi kombinačními sítěmi. Jedna z nich určuje chod vpřed, druhá chod vzad. Kombinační síť pro chod vpřed pracuje tak, že umožní změnu stavu klopného obvodu, jsou-li klopné obvody příslušející nižším bitům ve stavu H. Kombinační síť pro chod vzad pak umožní změnu stavu klopného obvodu tehdy, jsou-li klopné obvody příslušející nižším bitům ve stavu L. Příklad takového řešení je na obr. 2.48. Pro chod vpřed musí být úroveň H na svorce „vpřed“ a úroveň L na svorce „vzad“. Pro chod vzad jsou úrovně opačné.



Obr. 2.48: Synchronní čítač vpřed – vzad realizovaný pomocí klopných obvodů J-K

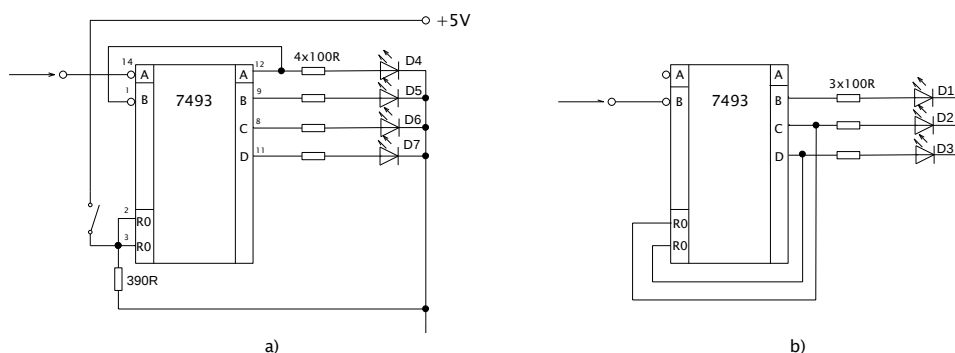
U synchronních čítačů lze zasahovat do chodu čítače obdobně jako u čítačů asynchronních. S pomocí vstupů „nulování“ je možno čítač vynulovat. Logickým členem, jímž se dekóduje stav čítače je možno zkrátit početní cyklus nebo zastavit chod čítače.

2.5.2.3 Integrované čítače

Integrovaný čítač MH7493 (obr. 2.49a) je binární asynchronní čítač vpřed, který počítá do šestnácti (tj. 0 – 15). Skládá se ze čtyř dvojitých klopných obvodů J-K. Jsou uspořádány tak, že vnějším propojením svorek obvodu umožňují určitou variabilitu čítače. První klopný obvod

má vyveden hodinový vstup a výstup, je tedy samostatný. Další tři obvody jsou spojeny jako binární čítač o třech bitech.

S použitím tohoto logického členu je možno zkracovat početní cyklus čítače, jak je patrné z obr. 2.49b. Signál přichází na vstup B a až do čísla 5 pracuje čítač známým způsobem. Pak se objeví na výstupech C, D stav H. Protože jsou oba zmíněné vstupy propojeny s nulovacími vstupy, následuje okamžité vynulování čítače.



Obr. 2.49: Základní zapojení integrovaného čítače MH7493

Integrovaný čítač MH7490 je desítkový čítač vpřed kódu BCD 1248. Jeho skladba je podobná, jako u čítače předchozího.

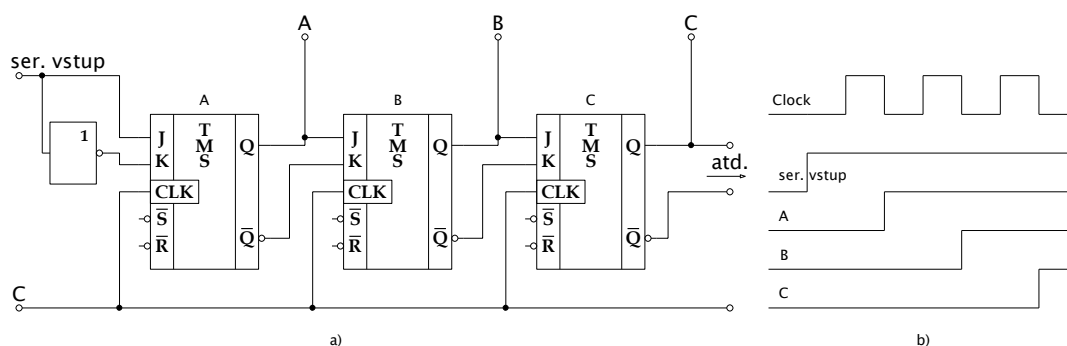
Integrovaný čítač MH74193 je binární synchronní obousměrný čítač. Čítač je sestaven ze čtyř dvojítkých klopných obvodů a z kombinační sítě logických členů pro řízení synchronní činnosti a chodu čítače. Obvod je opatřen dvěma hodinovými vstupy, z nichž jeden slouží pro chod vpřed, druhý pro chod vzad. Uplatňuje se vždy ten vstup, na který jsou přiváděny impulsy, zatímco na druhém je úroveň H. Oba vstupy jsou vedeny přes invertory, takže stav klopných obvodů se mění s čelem hodinových impulsů. Čítač má programovatelný obsah. Jeho výstupy mohou být nastaveny na potřebný výchozí stav prostřednictvím vstupů „data“. Aby se zápis realizoval, je nutno na tyto vstupy přivést žádané úrovně. Poté se uvede vstup „nastavení“, který je připojen rovněž přes invertor, na úroveň L (impuls). Obsah čítače je možno vynulovat dalším asynchronním vstupem „nulováním“. Je připojen přes invertor, takže k vynulování je třeba na něj přivést impuls úrovně H. Čítače je možno pohodlně řadit do kaskád. Pro chod vpřed se spojí výstup „přenos nahoru“ se vstupem „počítání vpřed“ následujícího čítače. Pro chod vzad se spojí výstup „přenos dolů“ se vstupem „počítání vzad“ dalšího čítače.

2.5.3 Posuvné registry

Bistabilní klopný obvod jako paměťový člen může uchovat jeden bit dvojkového čísla. V číslicové technice je často třeba uchovat dvojkové číslo o větším počtu bitů. K tomu je nutno využít většího počtu klopných obvodů. Tyto obvody musí být ovšem vhodně organizovány tak, aby bylo možno dvojkové číslo pohodlně vkládat a vyjímat. Jeden ze způsobů takové organizace je použit v posuvných registrech. Posuvným registrem rozumíme zařízení využívající bistabilních klopných obvodů, do něhož můžeme vložit informaci a tu pomocí vnějších řídicích signálů posouvat od jednoho klopného obvodu ke druhému. Počet klopných obvodů uvnitř registru udává délku registru a současně počet bitů dvojkového čísla, který může být uložen.

2.5.3.1 Posuvné registry J-K

Uvažujme jednoduché zapojení tří klopných obvodů podle obr. 2.50a. Předpokládejme, že všechny klopné obvody jsou na počátku ve stavu L. Na vstup J obvodu A nyní přivedeme úroveň H, jejíž negovaná hodnota se objeví na vstupu K. Přivedeme-li nyní hodinový impuls na vstup C, přejde s jeho týlem obvod A do stavu H. Ostatní obvody měly před příchodem hodinového impulsu na vstupech J úroveň L a na vstupech K úroveň H. Podle pravdivostní tabulky zůstane tedy jejich stav po ukončení hodinového impulsu nezměněn. Předpokládejme, že úroveň H na vstupu A trvá. Po druhém hodinovém impulsu setrvá obvod A ve stavu H a obvod B přejde do stavu H. Po třetím hodinovém impulsu budou ve stavu H všechny tři klopné obvody. Informace H, která byla na vstupu registru, se třemi hodinovými impulsy přesunula na výstup registru. Činnost obvodu můžeme vyjádřit časovým diagramem podle obr. 2.50b.

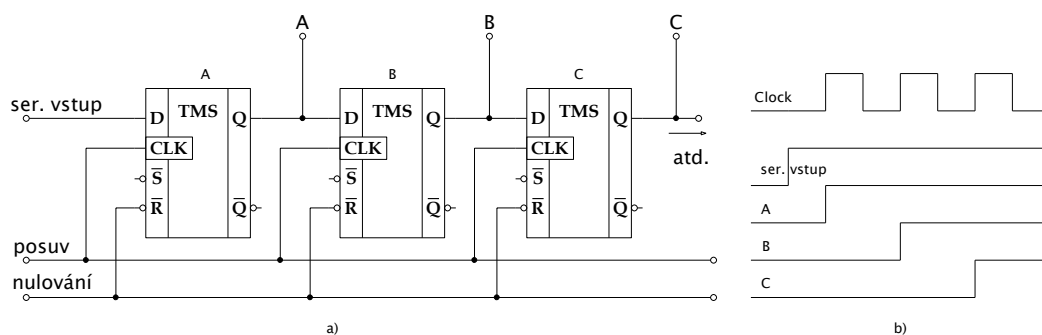


Obr. 2.50: Posuvný registr realizovaný pomocí klopných obvodů J-K

Jsou-li všechny obvody na počátku ve stavu H, můžeme analogicky posouvat úroveň L přivedenou na vstup. Chceme-li posouvat úroveň L, je ovšem také možno ponechat původní uspořádání a posouvanou úroveň odebírat z výstupů Q klopných obvodů. Uvedený obvod je posuvným registrem se sériovým vstupem a sériovým výstupem.

2.5.3.2 Posuvné registry D

Posuvné registry D jsou budovány na shodném principu, jako registry J-K. Je si však nutno uvědomit, že stav klopného obvodu D se mění s čelem a nikoli a týlem hodinového impulsu. Zapojení posuvného registru D, které je analogické obvodu podle obr. 2.50a, je na obr. 2.51a. Úroveň H se přesune s prvním hodinovým impulsem na výstup A, s druhým impulsem na výstup B a s třetím na výstup C. Časový diagram je na obr. 2.51b. Podobně může být posouvána úroveň L, právě tak je možno posouvat jediný impuls nebo sled impulsů.



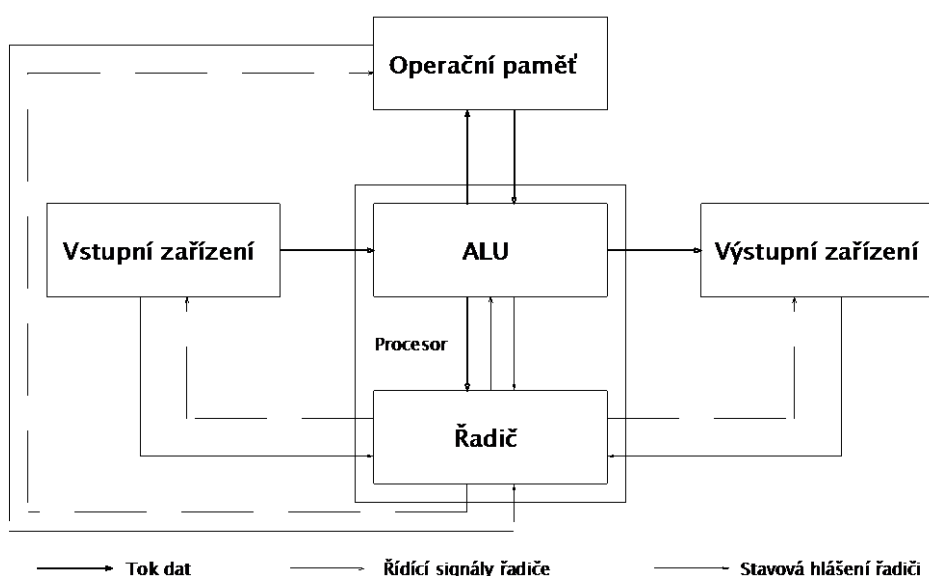
Obr. 2.51: Posuvný registr realizovaný pomocí klopných obvodů D

2.6 Základy mikroprocesorové techniky

V současné době jsou mikropočítače celosvětově nejrozšířenějšími počítači. Rozsah jejich nasazení je až nezvykle široký. V nejjednodušších variantách se jich používá ke zvýšení "inteligence" takových zařízení, jako jsou výrobky spotřební elektroniky (mobilní telefon, televizor apod.), výrobky měřicí techniky (osciloskopy, voltmetry aj.), ale i v mnoha dalších odvětvích (např. automobil). Mikropočítač vybavený příslušnými periferiemi se může stát výkonným kancelářským pomocníkem, můžeme si s ním doplňovat své znalosti a dovednosti, případně si i hrát.

2.6.1 Struktura mikropočítače

Mikropočítač se skládá z funkčních bloků, které se nazývají mikroprocesor, operační paměť a vstupní a výstupní obvody. Tyto funkční bloky spolu s dalšími logickými obvody a vodiči potřebnými pro jejich spolupráci se nazývají technické vybavení nebo technické prostředky (angl. **hardware**). Pro komunikaci s okolním světem slouží celá řada periferních zařízení, jakými jsou u PC např. klávesnice, myš, či monitor, a mikropočítače běžně využíváme tlačítka, maticovou klávesnici, krokový motor, servomotor, LED či LCD zobrazovač. Mikropočítač je řízen programem, který obsahuje sadu instrukcí, a právě program „říká“ počítači, co má vlastně dělat. Pro programy se běžně užívá souhrnný název **software**.



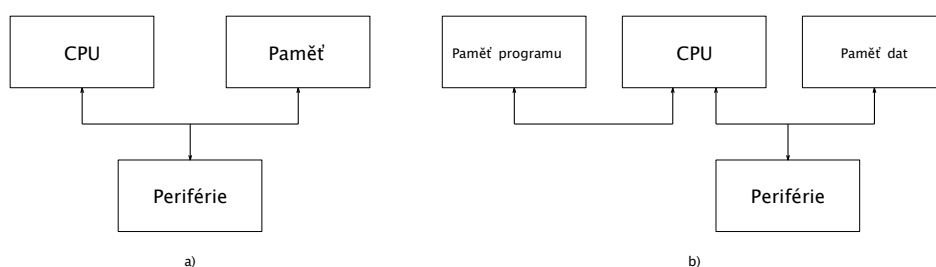
Obr. 2.52: John von Neumannův model počítače

V roce 1945 definoval americký matematik maďarského původu John von Neumann základní koncepci počítače EDVAC (Electronic Discrete Variable Automatic Computer). Od

té doby se objevilo několik různých modifikací, ale v podstatě se počítače, tak jak je známe v dnešní době, konstruují podle tohoto modelu. Architektura dle von Neumanna je na obr. 2.52.

Několik let po von Neumannovi přišel vývojový tým z Harvardské univerzity s vlastní koncepcí počítače, která se od von Neumanna příliš nelišila, ale odstraňovala některé její nedostatky. V podstatě jde pouze o oddělení paměti pro data a program.

Abychom si mohli obě koncepce porovnat, můžeme vycházet ze zjednodušených schémat. Na obr. 2.53a je schéma počítače podle von Neumanna a na obr. 2.53b je schéma počítače harvardské koncepce.



Obr. 2.53: Srovnání John von Neumannova a harvardského modelu počítače

Srovnání výhod a nevýhod obou koncepcí:

a) von Neumann

Výhody:

- rozdělení paměti pro kód a data určuje programátor,
- řídicí jednotka procesoru přistupuje do paměti pro data i pro instrukce jednotným způsobem,
- jedna sběrnice – jednodušší výroba.

Nevýhody:

- společné uložení dat a kódu může mít při chybě za následek přepsání vlastního programu,
- jediná sběrnice tvoří úzké místo.

b) harvardská koncepce

Oddělení paměti programu a paměti dat přináší výhody:

- program nemůže přepsat sám sebe,

- paměti mohou být vyrobeny odlišnými technologiemi,
- každá paměť může mít jinou velikost nejmenší adresovací jednotky,
- dvě sběrnice umožňují přístup pro instrukce i data současně.

Nevýhody:

- dvě sběrnice kladou vyšší nároky na vývoj řídicí jednotky procesoru a zvyšují i náklady na výrobu výsledného počítače,
- nevyužitou část paměti dat nelze použít pro program a obráceně.

2.6.2 Procesor

Pokud v následujícím textu bude používat pojem procesor, budeme mít na mysli mikroprocesor, tedy procesor integrovaný v jednom integrovaném obvodu, v jedné elektronické součástce. Obsahuje aritmeticko-logickou jednotku (ALU), univerzální registry a další logické obvody.

Nejjednodušší skupinou procesorů jsou tzv. mikrořadiče často označované jako **MCU** (Micro Controller Unit), jejichž uplatnění je velmi široké, od běžné spotřební elektroniky až po výkonné počítače. Mezi výhody těchto čipů patří nízká cena, malé rozměry a nízká spotřeba energie. Tyto procesory jsou vyráběny pro přesně určenou specifickou činnost a mají jen malou možnost rozšíření. Rovněž ve výkonnosti nedosahují vysokých hodnot.

Procesory nesoucí označení **CPU** (Central Processor Unit) tvoří základní řídicí jednotku počítače. Oproti předcházející skupině mají mnohem vyšší výkon, větší rozměry, je možné je díky jejich otevřené architektuře a velkému množství vyvedených signálů lépe rozšiřovat. Samozřejmě tyto výhody jsou zaplacené vyšší spotřebou a ztrátovým výkonem, rovněž cena je vysoká.

DSP (Digital Signal Processor) je určitým kompromisem mezi oběma předcházejícími skupinami procesorů. Signálové procesory se většinou vyznačují vysokým výkonem v oblasti zpracování matematických výpočtů a schopností zpracovávat velké objemy dat. Součástí těchto procesorů jsou často i digitálně-analogové a analogově-digitální převodníky. Používají se například ve zvukových kartách.

Jak to tedy v procesoru funguje? Do procesoru vstupují data, ale také instrukce. V řadiči jsou jednotlivé instrukce dekodovány na mikroinstrukce, které pak řídí činnost ALU a ta následně zpracovává data a posílá je na výstup. Sběrnice procesoru mají také svou šířku, která se neustále zvyšuje s vývojem nových generací procesorů (8 bitů, 16 bitů, 64 bitů, 128 bitů).

Procesory pracují s instrukcemi a podle nich vykonávají příkazy. Zjednodušeně řečeno, je každá akce, která se má provést rozložena na posloupnost instrukcí (popř. dále mikroinstrukcí) a ty jsou postupně prováděny procesorem. Procesory se z hlediska vnitřní stavby dají rozdělit na dva typy:

- **CISC** (Complete Instruction Set Computing) - jsou procesory používané ve většině současných i dřívějších osobních počítačů. Hlavním rysem těchto procesorů je, že používají tzv. plnou instrukční sadu neboli se snaží mít na každou úlohu jednu instrukci. Tyto instrukce jsou uloženy v mikrokódu, což je vlastně program vložený do paměti procesoru. Tento systém vytváření je z hlediska technologického jednodušší, ale instrukce se provádějí pomaleji než u obvodového řešení. Příklady: i8086, i80486, Pentium,...

- **RISC** (Reduced Instruction Set Computing) - procesory s redukovanou instrukční sadou. Obsahují jen několik základních instrukcí. Každá z nich by se měla vykonávat co nejkratší dobu, pokud možno během jediného strojového cyklu. Instrukce jsou vytvořeny obvodově a tudíž se většinou provádějí rychleji než u mikrokódového řešení, např. mikrokontroléry PIC.

2.6.3 Paměť

Pojmem paměť obecně označujeme zařízení libovolného typu, do kterého je možno zaznamenat informaci, dlouhodobě ji zde uchovat a podle potřeby znovu načíst (např. magnetické disky, optické disky, magnetické pásky a mnoho dalších). Paměti, které lze realizovat na křemíkovém čipu bipolární nebo unipolární technologií, nazýváme polovodičové paměti. Za nejjednodušší paměťové prvky můžeme považovat klopné obvody (např. R-S, D a J-K), které jsou schopny zapamatovat si právě jeden bit informace.

Nejdůležitější parametry pamětí:

- **kapacita** – udává množství informací, které lze do paměti uložit,

- **přístupová doba** – doba, kterou je nutné čekat od zadání požadavku do zpřístupnění požadované informace,
- **šířka přenosu dat** – počet bitů přenášených po sběrnici (jednotka bit),
- **přenosová rychlost** – množství dat, které lze z paměti přečíst (popř. zapsat) za jednotku času
- **energetická závislost** - paměti, které při ztrátě napájecího napětí přichází o svůj obsah, se v angličtině označují přívlastkem volatile (česky: nestálé, nebo též: **volatilní**), Paměti, které si svůj obsah dokáží podržet i po odpojení napájecího napětí, se pak označují jako non-volatile (stálé, **nevolatilní**).
- **cena za bit**.

Kapacita paměti

Bit je základní jednotkou informace, může nabývat hodnot 1 nebo 0 (ANO-NE). Bit je velmi malá jednotka a tak se udává kapacita paměti v jednotce větší, která se jmenuje Byte (čti Bajt). Byte je složen z 8 bitů, tj. 1 Byte = 8 bitů.

Byte [B] – 8 bitů, slabika, obvykle nejmenší adresovatelná jednotka

$$1 \text{ kB} = 2^{10} \text{ B} = 1024 \text{ B}$$

$$1 \text{ MB} = 2^{20} \text{ B} = 1048576 \text{ B}$$

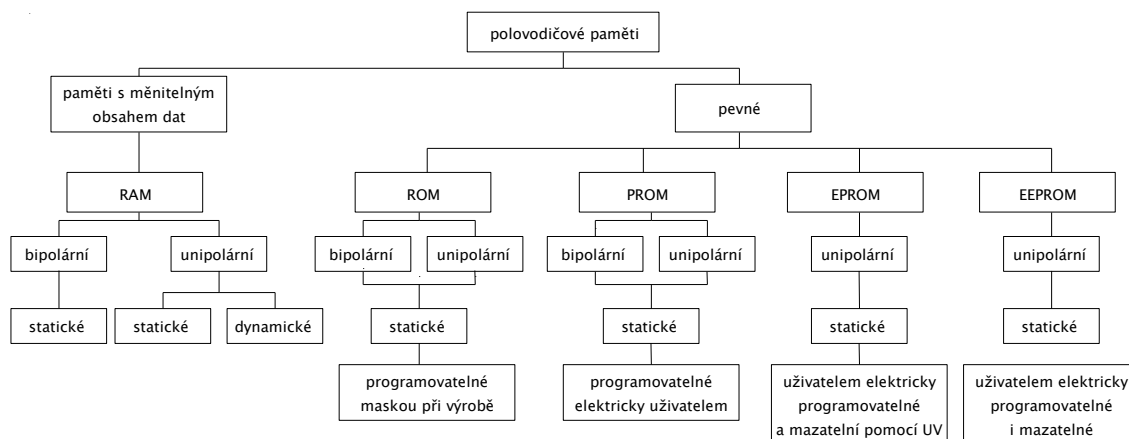
$$1 \text{ GB} = 2^{30} \text{ B} = 1073741824 \text{ B}$$

$$1 \text{ TB} = 2^{40} \text{ B} = 1099511627776 \text{ B}$$

V praxi se velmi často používá přepočet kapacity v dekadické soustavě, kdy se vychází z přibližného vztahu: 2^{10} je přibližně 10^3 , tedy 1024 je přibližně 1000 – zde dochází k rozdílu mezi údaji uvedenými v technických parametrech některých zařízení, jako jsou např. pevné disky.

Rozdělení polovodičových pamětí:

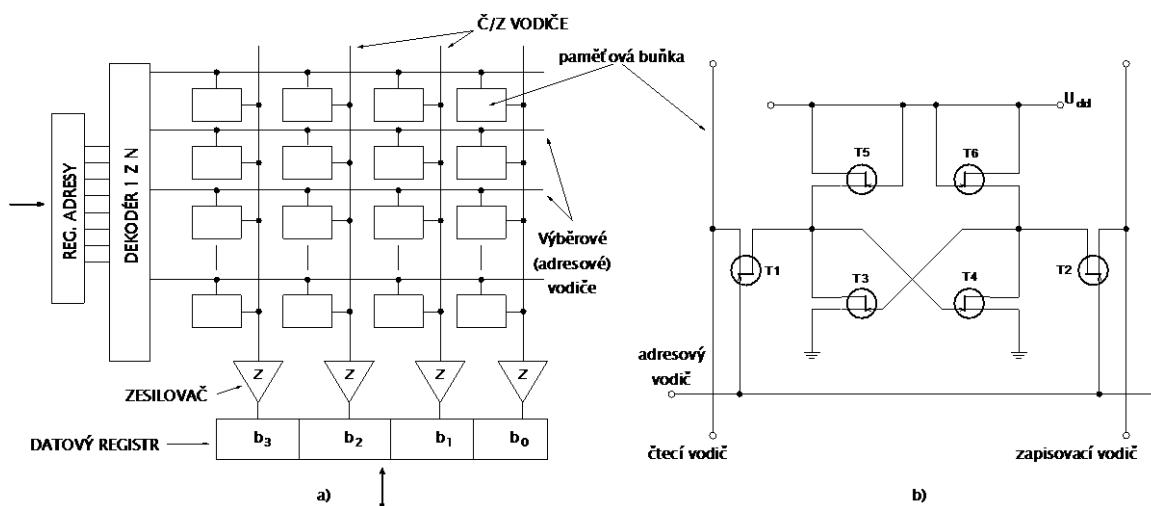
Základní rozdělení polovodičových pamětí je znázorněno na obr. 2.54.



Obr. 2.54: Rozdělení polovodičových pamětí

Paměti SRAM (Static Random Access Memory)

Jako všechny polovodičové paměti jsou i statické paměti RAM uspořádány na čipu do matice, uspořádání je na obr. 2.55a. Matice je tvořena sítí paměťových buněk které jsou realizovány bistabilními klopnými obvody, tj. obvody, které se můžou nacházet vždy v jednom ze dvou stavů, které určují, zda v paměti je uložena 1 nebo 0. Statická paměťová buňka je na obr. 2.55b.



Obr. 2.55: Matice paměťových buněk a statická paměťová buňka

U SRAM paměti se používá dvou datových vodičů (čtecího a zapisovacího). Vodič označený jako čtecí se používá ke čtení. Hodnota na tomto vodiči je vždy opačná než hodnota uložená v paměti. Takže na konci je nutno ji ještě negovat.

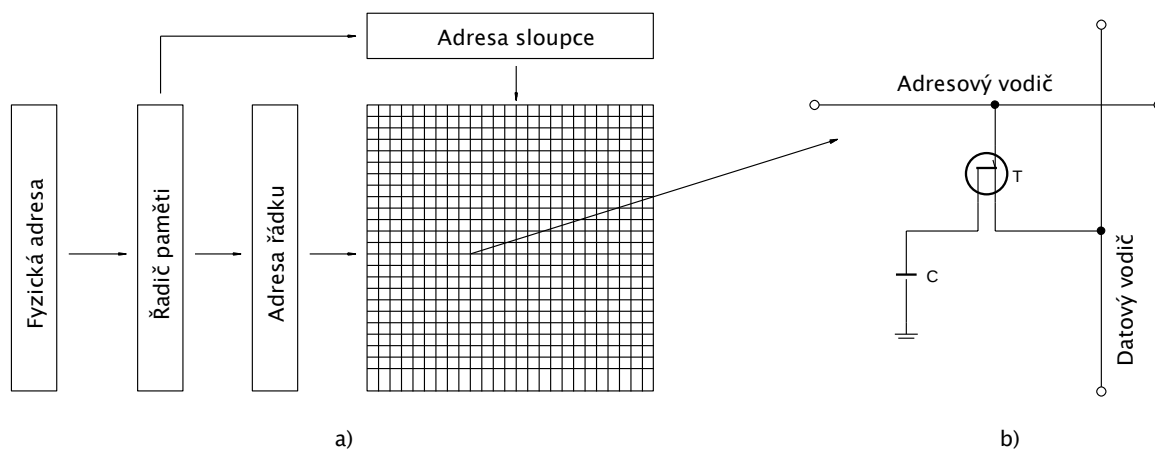
Při zápisu se na adresový vodič umístí hodnota logická 1. Tranzistory T1 a T2 se otevřou. Na zapisovací vodič se přivede zapisovaná hodnota (např. 1). Tranzistor T1 je otevřen, takže jednička na čtecím vodiči otevře tranzistor T4 a tímto dojde k uzavření tranzistoru T3. Tento stav obvodu představuje uložení hodnoty 0 do paměti. Zcela analogicky tato buňka pracuje i při zápisu hodnoty 1. Rozdíl je pouze v tom, že tranzistor T4 zůstane uzavřen a to způsobí otevření tranzistoru T3.

Při čtení je opět na adresovací vodič přivedena hodnota logická 1, což opět způsobí otevření tranzistorů T1 a T2. Jestliže byla v paměti zapsána hodnota 1, je tranzistor T4 otevřen (tj. na jeho výstupu je hodnota 0). Tuto hodnotu obdržíme na zapisovacím vodiči. Opět zcela analogicky v případě uložené hodnoty 0, kdy tranzistor T4 je uzavřen (tj. na jeho výstupu je hodnota 1).

Tranzistory T5 a T6 plní pouze funkci rezistorů.

Dynamická paměť RAM (DRAM - Dynamic Random Access Memory)

V současné době jsou DRAM nejpoužívanějším typem polovodičových pamětí. To je způsobeno velmi výhodným poměrem kapacita/cena.



Obr. 2.56: Matice paměťových buněk a dynamická paměťová buňka

Paměťová buňka je tvořena dvojicí kondenzátor C (využívá se kapacity PN přechodu) a tranzistor T typu MOSFET (obr 2.56b). Kondenzátor je řízeně nabíjen. Mezi jeho vývody lze naměřit napětí, které je úměrné velikosti náboje. Logické hodnoty 0 a 1 odpovídají vybitému a nabitému kondenzátoru. Aby nedošlo ke ztrátě informace (z důvodu samovolného vybíjení

kondenzátoru), provádí se periodická obnova dat, tzv. **DRAM refresh**. Refresh se provádí po celých řádcích. V okamžiku obnovování informace nelze provádět operace čtení/zápis.

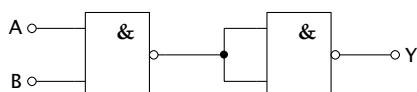
Při čtení je na adresový vodič přivedena hodnota log. 1, která způsobí otevření tranzistoru T. Pokud byl kondenzátor nabitý, zapsaná hodnota přejde na datový vodič. Tímto čtením však dojde k vybití kondenzátoru a tedy ztrátě uložené informace. Přečtenou hodnotu je nutné opět do paměti zapsat.

Z důvodu periodické obnovy informace a obnovy informace po jejím přečtení jsou paměti DRAM pomalejší při čtení/zápisu než paměti SRAM. Výhodou je menší velikost paměťové buňky na polovodičovém čipu – dosahují tedy větších kapacit než paměti SRAM.

Využití: operační paměť, videopaměť grafické karty.

2.6.4 Otázky k procvičování

1. Čím se liší analogový a číslicový signál?
2. Vysvětlete pojem logický obvod!
3. Jakým způsobem se realizuje v číslicovém obvodu logická 0 nebo logická 1?
4. Vyjmenujte tři základní logické funkce!
5. Nakreslete schematickou značku a napište pravdivostní tabulku hradla NAND!
6. Napište De Morganovy zákony!
7. Co znamená pojem minimalizace logické funkce?
8. Odvoďte výslednou funkci Y následujícího obvodu:



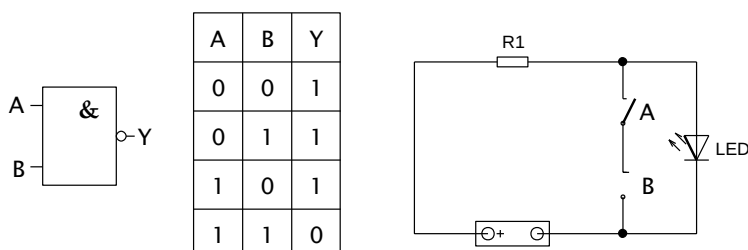
9. Jaký je rozdíl mezi kombinačními a sekvenčními logickými obvody?
10. Vyjmenujte základní kombinační logické obvody!
11. Na čem závisí stav sekvenčních logických obvodů?
12. Vyjmenujte základní sekvenční logické obvody!
13. K čemu dojde, přivedeme-li zároveň na oba vstupy RS klopného obvodu úroveň H?
14. K čemu slouží čítač?
15. Porovnejte John von Neumannovu a harvardskou architekturu mikropočítače? Jaké mají výhody a nevýhody?
16. Vysvětlete pojmy bit a Byte. Jak spolu uvedené pojmy souvisí?

2.6.5 Odpovědi na otázky za textem

1. Analogový signál (text obr. 1) může nabývat jakékoliv hodnoty od minimální hodnoty až po hodnotu maximální. Je spojitý. Číslicový (digitální) signál (text obr. 2) je na rozdíl od signálu analogového nespojitý, tzn. k jeho změně dochází skokem. Nabývá dvou hodnot (log. 0, log 1).
2. Logický obvod je integrovaný obvod, pracující s číslicovým signálem (úrovněmi 0/1), vzniká spojením několika logických členů (hradel) do skupiny.
3. U elektrických obvodů logické stavy 0 a 1 nepřísluší určitým diskrétním napětím, ale pásmům napětí. V symbolice elektronických logických obvodů se proto k odlišení nepoužívají symboly 0 a 1 ale symboly L a H. L znamená malé napětí, tj. stav 0, H znamená velké napětí, tj. stav 1. Úrovněová pásma TTL jsou zobrazena v následující tabulce.

Logická úroveň	Vstup	Výstup
Log. 0	0 – 0,8 V	0 – 0,4 V
Log. 1	2 – 5 V	2,4 – 5 V

4. K základním logickým funkcím patří OR (logický součet), AND (logický součin) a NOT (negace).
- 5.



6. Teorém De Morganův: $\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{A \cdot B} = \overline{A} + \overline{B}$
7. Zjednodušování obvodů pomocí Booleovy algebry a s využitím Karnaughových map. Minimalizací logické funkce získáme nejjednodušší možný zápis pro danou funkci, což obvykle vede k výraznému zjednodušení praktické realizace dané funkce.

8. Celkově zapojení plní logickou funkci AND – jde o zapojení prvního hradla NAND, kdy na výstup je připojen investor (druhé hradlo NAND v zapojení se spojenými vstupy).
9. Podstatnou částí sekvenčních logických obvodů na rozdíl od kombinačních logických nebo obvodů je paměťový člen.
10. Mezi kombinační logické obvody patří např. kodéry, dekodéry, multiplexery, demultiplexery, generátory parity, binární sčítačky, atd.
11. Stav výstupu sekvenčních logických členů a obvodů závisí nejen na kombinaci vstupních proměnných (tak jak tomu bylo u kombinačních obvodů), ale i na stavu výstupu při předchozí kombinaci vstupních proměnných.
12. Mezi základní sekvenční logické obvody patří klopné obvody RS, RST, D, JK, čítače a posuvné registry.
13. Zvláštní případ nastane, přivedeme-li na vstupy R a S současně úroveň H. Po dobu působení těchto úrovní budou mít výstupy obou logických členů úroveň L. Odstraníme-li úroveň H ze vstupů, přejde klopný obvod náhodně do jednoho z obou stavů. Výsledný stav klopného obvodu bude tedy neurčitý. Stav, při němž na vstupy R a S působí současně úroveň H, nemůžeme pro účely logického rozhodování používat.
14. Čítač je obecně zařízení, které v nějakém kódu počítá elektrické impulsy přivedené na vstup čítače.
15. Harvardská architektura na rozdíl od John von Neumannovy architektury má oddělenou paměť dat a paměť programu. Srovnání výhod a nevýhod obou koncepcí:
von Neumann - výhody: rozdělení paměti pro kód a data určuje programátor, řídicí jednotka procesoru přistupuje do paměti pro data i pro instrukce jednotným způsobem, jedna sběrnice – jednodušší výroba. **Nevýhody:** společné uložení dat a kódu může mít při chybě za následek přepsání vlastního programu, jediná sběrnice tvoří úzké místo.
Harvardská koncepce: Výhody: program nemůže přepsat sám sebe, paměti mohou být vyrobeny odlišnými technologiemi, každá paměť může mít jinou velikost nejmenší adresovací jednotky, dvě sběrnice umožňují přístup pro instrukce i data současně. **Nevýhody:** dvě sběrnice kladou vyšší nároky na vývoj řídicí jednotky procesoru a zvyšují i náklady na výrobu výsledného počítače, nevyužitou část paměti dat nelze použít pro program a obráceně.

16. Bit je základní jednotkou informace, může nabývat hodnot 1 nebo 0 (ANO-NE). Bit je velmi malá jednotka a tak se udává kapacita paměti v jednotce větší, která se jmenuje Byte (čti Bajt). Byte je složen z 8 bitů, tj. $1 \text{ Byte} = 8 \text{ bitů}$.

3. ZABEZPEČOVACÍ TECHNIKA

3.1 Úvod

Nutnost chránit majetek a informovat o nebezpečí je tu od nepaměti. Posun ve vývoji lidstva znamenal logicky i pokrok v technickém řešení problému ochrany osob a majetku. Od informace předávané pomocí obsluhy, například hlasem, troubením nebo zvoněním, se postupně přecházelo na přenášení poplachové informace pomocí technických prostředků.

Výrazný pokrok nastal v době vědecko-technické revoluce a to vynálezem telegrafu. Využití telegrafu pro přenos informace o požáru na požární stanici byl poprvé použit v New Yorku v roce 1847. První zabezpečovací systém jak ho dnes známe, byl patentován v roce 1853 a provoz prvního pultu centrální ochrany byl spuštěn roku 1858. Až do začátku 20. století se využívaly výhradně snímače s mechanickými kontakty. První senzory byly konstruovány jako vibrační, inerciální (reagují na změnu polohy), setrvačnickové a kyvadlové. V podstatě až do objevu polovodičů za druhé světové války se využívala relé, i když postupně zdokonalovaná.

Prudký rozvoj v elektronice znamenal vznik nových druhů čidel reagující na podněty, které senzory mechanickými nelze zaznamenat. Od padesátých let 20. století se začínají využívat elektronické snímače a to hlavně akustické, kapacitní a ultrazvukové. V tomto období začínají být používána magnetická čidla s jazýčkovými kontakty. Díky rozmachu elektronického průmyslu se od šedesátých let začíná používat vysokofrekvenčních čidel a posléze čidel mikrovlnných. V tom samém období se na trhu objevují tzv. světelné závory, a to hlavně díky miniaturizaci zdrojů světla.

Nejrozšířenější detektor používaný dodnes se na trhu objevil v druhé polovině sedmdesátých let a to pasivní infračervené čidlo (PIR). I přesto, že tato čidla nejsou nejspolehlivější, díky jednoduchosti a energetické náročnosti vytlačila mikrovlnná a ultrazvuková.

Další rozvoj elektroniky a miniaturizace přináší dnes na trh další možnosti registrace a analýzy podnětů a to nejen mechanických, ale i biologických. Tzv. biometrické systémy se těší velkému komerčnímu využití.

3.2 Systémy elektronického zabezpečení

Elektronické zabezpečovací systémy jsou soubory technických a organizačních opatření, které musí stát v cestě mezi chráněným zájmem a nebezpečím, kterému má čelit. Může jít o majetkové nebo jiné hodnoty, které chráníme před odcizením, poškozením, zničením nebo jiným způsobem narušení. V současné době se zvyšuje nutnost důsledné

ochrany majetku, ale také i důležitých dat. Proto jsou tyto systémy běžným základním standardem základní infrastruktury budov v podnikové i státní sféře, ale také v bytových objektech. Dnes již za příznivých cen se vstupní náklady na instalaci těchto systémů stávají výhodnou investicí s rychlou návratností.

Velmi důležitou otázkou je kvalitní analýza rizik daného objektu a následně zpracovaný projekt. Tato činnost spolu s pružným a rychlým servisem je dnes samozřejmostí, včetně maximálního využití technologických novinek. Elektronické zabezpečovací systémy nejsou systémy izolované, ale jsou schopny být provázány s ostatními systémy jako CCTV, EPS, přístupových či docházkových systémů, integračních a ovládacích systémů, zařízení pro provoz či pohodlí zaměstnanců uvnitř budov.

3.3 Termíny a definice hlavních pojmů

Elektrický (elektronický) zabezpečovací systém (EVS):

elektronické zařízení určené k detekci a indikaci vstupu nebo pokusu o vstup do střeženého objektu nebo prostoru

Podsystém (subsystém) EVS:

část EVS určená pro střežení části objektu nebo prostoru, umožňující individuální provoz.

Komponenty EVS:

jednotlivé části EVS, jejichž spojením se vytvoří funkční systém.

Čidlo EVS:

zařízení, které vyšle informaci o narušení střeženého prostoru.

Tísňový hlásič EVS:

zařízení, pomocí něhož je možno vyhlásit poplach ručně.

Ústředna EVS:

zařízení, které zpracovává informace o stavu jednotlivých čidel a reaguje předem nastaveným způsobem.

Přenosové zařízení (komunikátor) EVS:

zařízení určené k dálkovému přenosu výstupních informací.

Pult centralizované ochrany EVS:

zařízení s trvalou obsluhou, dohlížející na jednotlivé objekty EVS.

Ovládací zařízení EZS:

zařízení umožňující ovládat EZS.

Poplach:

reakce prvku EZS na příslušný podnět.

Sabotáž:

nepovolená manipulaci s jednotlivými prvky EZS.

Falešný poplach:

poplach nevyvolaný regulérním podnětem.

Provozní kniha EZS:

dokumentace o provozu EZS.

3.4 Stupně zabezpečení

1. Nízký

Znalost pachatele o EZS není téměř žádná, stejně jako možnost použití nástrojů.

2. Nízký až střední

Pachatel má nízké znalosti o zabezpečovacím systému a má k dispozici jednoduché nástroje.

3. Střední až vysoký

Pachatel má znalosti o EZS a má dostupné přístroje a jednoduchá elektronická zařízení.

4. Vysoký

Pachatel zná použitý systém elektronické ochrany, má celý sortiment zařízení a prostředků umožňujících výměnu hlavních komponent EZS.

Základním požadavkem na stupeň zabezpečení pro jednotlivé prvky EZS řeší norma ČSN EN 50131-1.

Hlavní kritéria začlenění do jednotlivých stupňů podle:

- citlivosti detektoru
- odolnosti proti sabotáži
- zakrytí čidla
- kontroly napájení
- možnosti autokontroly
- odolnost vůči elektromagnetickému poli

- těsnosti krytu
- způsobu záznamu událostí

Samozřejmostí je splnění požadavků na EMC podle norem ČSN EN 55 022 a ČSN EN 50 130-4.

3.5 Rozdělení jednotlivých komponent EZS na prvky:

1. Plášťové ochrany:

- magnetické kontakty
- mechanické kontakty
- detektory skleněných výplní
- vibrační detektory
- drátová čidla

2. Prostorové ochrany:

- infračervené detektory
- mikrovlnné detektory
- ultrazvukové detektory
- kombinované (duální) detektory

3. Předmětová ochrana

- kapacitní snímače
- otřesová čidla

4. Tísňová ochrana

- osobní hlásiče
- veřejné hlásiče
- skryté hlásiče

5. Detektory obvodové (perimetrické) ochrany

- infračervené závory
- mikrovlnné bariéry
- tlakové (zemní) hadice

6. Ovládací prvky

- klávesnice
- čtečky karet (čipů)
- dálkové ovladače

7. Ústředny

- drátové ústředny
- bezdrátové ústředny
- kombinované ústředny

8. Výstražná zařízení

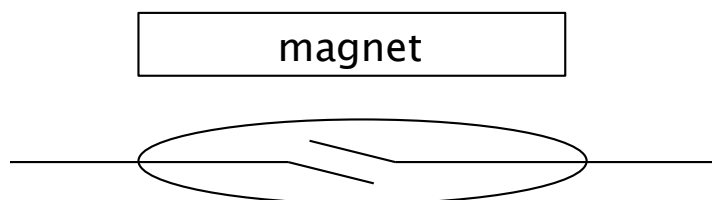
- sirény
- majáky

3.6 Plášťová ochrana

Jak plyne z názvu, tyto detektory slouží k indikaci narušení obvodových konstrukcí a vstupů do budovy, typicky dveří a oken.

3.6.1 Magnetické kontakty

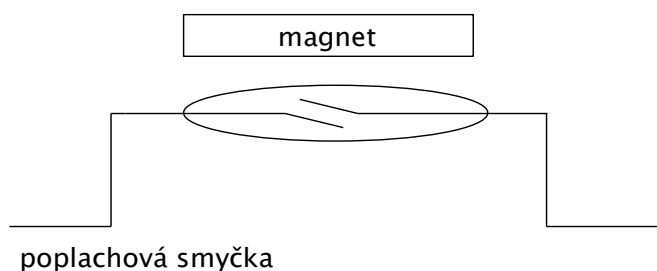
Funkce a konstrukce



Obr. 3.01 Princip magnetického detektoru

Jsou sestaveny ze dvou částí, permanentního magnetu a jazýčkového kontaktu. Kontakt je uzavřen ve skleněné trubičce naplněné inertním plynem. Přiblížení permanentního magnetu do blízkosti kontaktu způsobí jeho sepnutí. Při oddálení naopak dojde k rozpojení a tím k aktivaci čidla. Magnetické detektory se vyrábí v několika provedeních, a to jak pro povrchovou tak i skrytou montáž. Speciální provedení je odolné vůči jinému magnetickému poli.

Připojení do smyčky



Obr. 3.02 Zapojení detektoru do poplachové smyčky

Zásady instalace

Magnetická čidla se využívají převážně k indikaci otevření vstupů do budov, a to dveří, vrat a oken.

Permanentní magnet se instaluje na pohyblivou a kontakt na pevnou část okna nebo dveří. Pro správnou funkci detektoru je nutné dodržet doporučení výrobce pro správnou montáž.

Například maximální vzdálenost mezi magnetem a kontaktem, osazení všech křídel dveří a oken, montáž na vnitřní stranu apod.

Ukázky detektorů

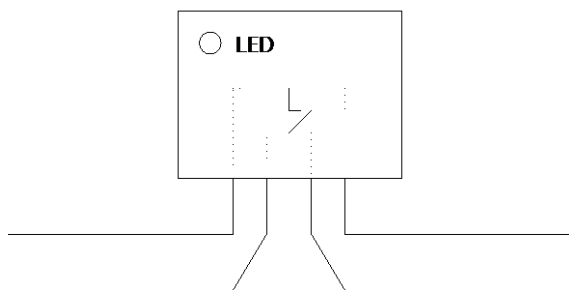


Obr. 3.03 Magnetický detektor SA-201

3.6.2 Detektory skleněných výplní

Tyto detektory mohou být umístěny přímo na chráněné skleněné ploše a reagují na vibrace při rozbití. Jiný typ čidel se instaluje na stěnu a snímá charakteristický zvuk při tříštění skla. Takové čidlo má tzv. duální detekci kdy přes pásmové propusti prochází jen přesně definovaná nízká a vysoká frekvence a čidlo vyvolá požadovanou akci.

Připojení do smyčky



Obr. 3.04 Připojení piezočidla

Ukázky detektorů



Obr. 3.05 Piezočidlo



Obr. 3.06 Detektor GBS-210

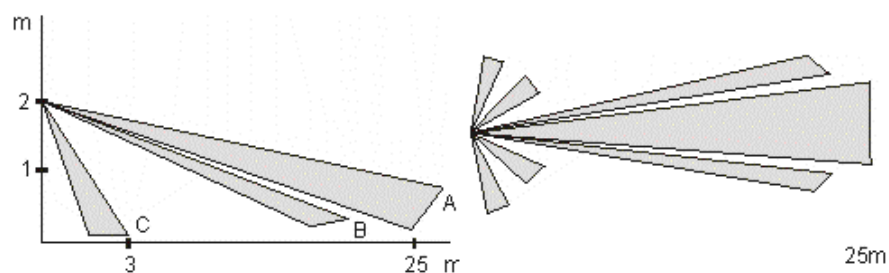
3.7 Prostorová ochrana

Detektory prostorové ochrany pokrývají prostor ohraničený pláštěm budovy. Tato čidla můžeme rozdělit na pasivní a aktivní. Do první skupiny patří velmi rozšířený detektor pohybu (PIR). Mezi aktivní se řadí čidla ultrazvuková (US), mikrovlnná (MW). Kombinací PIR – US nebo PIR- MW se dosahuje vyšší odolnosti vůči falešným poplachům. Uvedené typy čidel využívají pro svou činnost různé frekvenční pásmo.

3.7.1 Detektory infračerveného záření

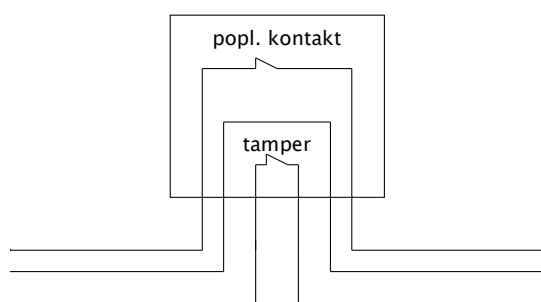
Funkce a konstrukce

Pasivní infračervená čidla (PIR) jsou jedny z nejpoužívanějších detektorů zabezpečovacích systémů. PIR detektory reagují na tepelné (infra) záření. Díky konstrukci optické čočky jsou schopny zachytit pohybující se předměty s teplotou odlišnou od teploty okolí.



Obr. 3.07 Příklad pokrytí PIR čidlem

Připojení do smyčky



Obr. 3.08 Připojení PIR čidla

Zásady instalace

Detektory pohybu střežíme zásadně vstupy do budovy, jako jsou schodiště, haly, vstupní chodby a samotné prostory střežených objektů. Náročnost montáže je nízká a proto se někdy využívá prostorová ochrana místo ochrany plášťové. Nenahrazují však plášťovou ochranu zcela.

Ukázky detektorů



Obr. 3.09 Detektor PIR drátový



Obr. 3.10 Detektor PIR bezdrátový

3.7.2 Mikrovlnná čidla

Funkce a konstrukce

Mikrovlnné detektory vyzařují do prostoru elektromagnetického záření o kmitočtu 10 nebo 24 Ghz. Detektor měří rozdíl kmitočtů mezi vyslaným a přijatým signálem. Tento rozdíl vzniklá při pohybu objektu na základě Dopplerova jevu. Zjednodušeně se dá říct, že od statického objektu se odrazí signál o stejném kmitočtu, zatímco od pohybujícího objektu se odrazí signál se změněným kmitočtem.

Připojení do smyčky

Elektrické připojení je obdobné jako u zapojení PIR a US čidel. Další zapojení doplňkových funkcí se liší podle daného typu a výrobce.

Zásady instalace

Stejně jako u PIR a US instalujeme čidla tak aby pravděpodobný směr pohybu pachatele vedl k čidlu nebo od čidla. V blízkosti čidla se nesmí nacházet větší předměty z kovu, obzvláště objekty s rovinným povrchem, od kterých se mikrovlny odrážejí a tím výrazně mění detekční charakteristiku. Mikrovlnná čidla nesmíme použít tam, kde dochází ke spínání zářivkového osvětlení. Více čidel pohromadě se smí použít, pokud každé z čidel pracuje na jiném vysílacím kmitočtu anebo je vyloučeno jejich vzájemné ovlivňování.

Ukázky detektorů



Obr. 3.11 Mikrovlnný detektor GT-432



Obr. 3.12 Čidlo pro spínání světel

3.7.3 Ultrazvuková čidla

Funkce a konstrukce

Ultrazvukové čidlo vysílá vysokofrekvenční impulsy, které se šíří prostorem rychlostí zvuku. Pokud narazí na nějaký předmět, odrazí se od něj a vrací se zpět k čidlu jako ozvěna. Z časového intervalu mezi vysláním impulsu a návratem ozvěny odvodí čidlo vzdálenost k předmětu. Čidlo využívá takzvaného Dopplerova jevu v pásmu ultrazvukových kmitočtů.

Připojení do smyčky

Elektrické připojení ultrazvukového čidla je stejné jako zapojení čidla PIR. Připojení doplňkových funkcí je specifické a liší podle konkrétního typu a výrobce.

Zásady instalace

Více US čidel je možné instalovat pouze, jsou-li vysílače synchronizovány nebo kmitočtově tak stálé, že se nebudou negativně ovlivňovat. US čidla nesmíme instalovat na pohyblivé konstrukce, na topná tělesa, v prostorech s horkovzdušným topením, v blízkosti zdrojů zvuku se širokým zvukovým kmitočtovým spektrem (telefon), tam kde se volně pohybují v době střežení zvířata.

US čidlo musíme instalovat tak, aby pravděpodobný pohyb pachatele směřoval k čidlu nebo od něj, typický dosah čidla je 10 m. Hlídaný prostor musí být uzavřený.

Předměty umístěné do blízkosti čidla až po jeho instalaci a nastavení mohou způsobovat poplach.

Ukázka detektorů



Obr. 3.13 Ultrazvukové čidlo CA 530

3.7.4 Kombinovaná čidla

Funkce a konstrukce

Použití těchto čidel je vhodné do prostorů s obtížnými podmínkami nebo nepříznivým vlivem okolního prostředí. Kombinovaná čidla jsou PIR – US nebo PIR – MW čidla. Základní myšlenka duálních čidel vychází z malé pravděpodobnosti, že nastanou takové jevy, které by vyvolaly poplach současně u obou čidel pracujících na různých fyzikálních principech. Spojením dvou čidel se eliminuje riziko vzniku falešných poplachů.

Zásady instalace

Při instalaci je nutné vycházet z pravidel, která platí pro jednotlivé systémy čidel. K použití kombinovaných čidel přistupujeme až ve velice nepříznivých podmínkách s ohledem na vyhlašování planých poplachů. Musíme si uvědomit, že díky použití dvou systémů pro detekci, se sám práh detekce poněkud posune oproti jednosystémovým čidlům.

Ukázka detektorů



Obr. 3.14 PIR + GBS (Glasbreak – snímač rozbití skla)

3.8 Tísňové hlásiče

3.8.1 Automatické tísňové hlásiče

Funkce a konstrukce

Jejich provedení umožňuje vyhlášení tísňového poplachu nezávisle na vůli obsluhy – pouze respektováním požadavků případného útočníka

Hlásiče jsou tvořeny čidly tzv. poslední bankovky.

Vyrábějí se ve dvou provedeních:

- Kontaktní (mechanická) čidla uzpůsobené k zasunutí bankovky do těla pouzdra
- bezkontaktní (optoelektronická čidla), ta pracují na principu reflexního optoelektronického vazebního členu, tedy bezkontaktně

Zásady instalace

Aplikujeme je tak, aby jejich montáž nebyla na první pohled patrná. V peněžních ústavech se umísťují do peněžních přihrádek a slouží k nepozorovanému vyvolání tísňového hlášení při přepadení.

Pracovní režim elektroniky čidla je nejčastěji pulsní, čímž je docíleno odolnosti proti osvětlení od cizích světelných zdrojů. Čidla jsou odolná proti cizím zdrojům světla do intenzity osvětlení cca 500 lx. Při montáži je třeba zajistit, aby nemohlo dojít k přímému průniku silného světelného paprsku do prostoru umístění optoelektronických reflexních čidel.

Ukázka detektorů



Obr. 3.15 Detektor poslední bankovky

3.8.2 Speciální tísňové hlásiče

Funkce a konstrukce

Jsou to magnetické kontakty nebo mikrospínače zapouzdřené do podoby tlačítka. Určena jsou zaměstnancům k nepozorovanému tísňovému hlášení v případě ohrožení.

Připojení do smyčky

Je stejné jako u veřejných tísňových hlásičů, akorát má-li elektronickou paměť, je k němu nutné přivést od poplachové ústředny napětí 12 V.

Zásady instalace

Montované by měly být tak, aby ze strany zákazníka nebyli vidět. Proto je umísťujeme na spodní stranu stolu či pultu. Nožní spínací lišty na trnože stolu zespodu či na můstky, peněžní svorky do peněžních přihrádek.

V případě zapojení většího počtu hlásičů na jednu smyčku, je účelné použít prvky s optickou signalizací pro potřeby identifikace planých poplachů způsobených chybou obsluhy i pro zpětnou analýzu poplachové události. Nutné je řádné proškolení obsluhy.

Hlásiče nemají žádnou ochranu před nechtěným použitím. Proto je důležité vybrat správné místo pro jejich montáž

Ukázka detektorů



Obr. 3.16 Nožní tísňová lišta

3.8.3 Veřejné tísňové hlásiče

Funkce a konstrukce

Je to buď magnetický kontakt či mikrospínač zapouzdřený do podoby tlačítka. Jeho hlavní smysl je vyvolání tísňového hlášení veřejností nebo pracovníkem.

Zásady instalace

Měly by se umisťovat na dobře viditelných místech, např. při schodištích, v chodbách a halách tak, aby je mohl použít každý kdo je v tísňové situaci, nebo je jejím svědkem. Pokud není hlásič přímo adresován, je účelné použít typ s mechanickou nebo elektronickou pamětí, aby bylo možné při analýze poplachové události zpětně zjistit, který hlásič ve smyčce byl aktivován.

Tísňové hlásiče bývají chráněny krycím sklem, které je třeba při jeho použití rozbít, sklo slouží jako ochrana před náhodným použitím a znesnadnění jeho zneužití. Falešné poplachy jsou v tomto případě minimálně.

Ukázky detektorů



Obr. 3.17 Požární hlásič

3.9 Perimetrická ochrana

3.9.1 Infračervené závory a bariéry

Funkce a konstrukce

Je to nejrozšířenější druh venkovních obvodových čidel. Mezi přijímací a vysílací stranou probíhá jeden nebo více infračervených paprsků. Dojde-li k přerušení některého z nich, dojde na přijímací straně k vyhodnocení a následnému vyhlášení poplachu. Pro zvýšení bezpečnosti proti cizím zdrojům světla, pracují infrazávory v pulsním režimu. Aby nedocházelo orosení optiky nebo nánosu vlhkosti z vnější strany, jsou většinou infrazávory vybaveny vyhříváním. Použitelný dosah je 50 až 150 metrů (někteří výrobci uvádějí až 250 m).

Připojení do smyčky

Elektrické připojení je specifické a vychází z instalačních manuálů konkrétního výrobce. Výstup bývá řešen jako bezpotenciálový kontakt relé, pro připojení do poplachové smyčky ústředny EZS.

Zásady instalace

Nižší cena oproti jiným druhům venkovních obvodových čidel je vykoupena pracnější přípravou, jako jsou stabilní montážní místa a zemní výkopové práce pro kabely. Při návrhu musíme pamatovat na příkon pro vytápění infrazávor. Při použití více čidel na jednom pozemku se musí úseky částečně překrývat, aby nevznikly tzv. mrtvé zóny. Rizikem pro tato čidla jsou především povětrnostní podmínky, jako například mlha, padající sníh, přímý sluneční svit. Proto jsou některá čidla vybavena automatikou, která snímá optickou propustnost prostoru mezi přijímačem a vysílačem. Při poklesu viditelnosti automatika vyřadí čidla z provozu a poplach není vyhlášován.

Ukázky detektorů



Obr. 3.18 Infra závora firmy OPTEX



Obr. 3.19 Bezdrátová infra závora Jablotron

3.9.2 Perimetrická pasivní infračervená čidla (infrateleskopy)

Funkce a konstrukce

Využívají princip infrapasivního čidla (PIR). Oproti PIR čidlům je zde použita jiná optika, složitější vyhodnocovací obvody, mechanicky je použita robustní, klimaticky odolná konstrukce s vytápěným pouzdrem. Dosah infrateleskopů je cca 150 metrů.

Připojení do smyčky

Připojení je typické jako u PIR čidel. Poplachový výstup je tvořen bezpotenciálovým kontaktem relé.

Zásady instalace

Infrateleskopy se využívají jako doplněk kamerových systémů, jako spínání poplachového monitoringu nebo záznamu. Místo na které čidlo montujeme, musí být mechanicky stabilní. Čidla je možné instalovat na stěny nebo sloupky. Musíme pamatovat na příkon pro vytápění krytu. Systém, díky použití diferenciálních vícenásobných pyrosenzorů a speciálních obvodů, eliminuje typické příčiny falešných poplachů (víření vzduchu, pohyb rostlin, dopadající sluneční paprsky nebo světla automobilů) na minimum. Schopnost eliminace se snižuje při nízkých rozdílech teploty pozadí a pachatele. Je vhodné instalovat čidla tak, aby neviděla až za hranice určeného střeženého objektu.

Ukázka detektorů



Obr. 3.20 Infradetektory s dlouhým dosahem

3.10 Ústředny EZS

Funkce a konstrukce

Rozhodující při výběru ústředny EZS je požadovaný stupeň zabezpečení (dle ČSN EN 50131-1) do kterého na základě bezpečnostního posouzení a následné analýzy rizik svým provozem objekt spadá. Dále rozsah objektu, jež má být střežen, a jeho stavební provedení finanční možnosti investora.



Obr. 3.21 Ústředna JA-63



Obr. 3.22 Ústředna JA-83K

Vstupní obvody

Každé čidlo EZS (napájené i nenapájené) je do poplachové nebo tísňové a zajišťovací smyčky propojeno pomocí vícežilového stíněného kabelu. Elektrické parametry všech smyček jsou shodné, liší se pouze způsobem hlášení svého narušení. Počet vstupů smyček je dán pouze druhem ústředny a může se prakticky pohybovat od čtyř až po stovky smyček.

Vstupní obvody nejjednodušších ústředěn bývají často velice primitivní a vyhodnocují pouze dva stavy, těmi jsou: „smyčka uzavřená“ nebo „smyčka rozpojená“. Jako správný funkční stav se volí „smyčka uzavřená“, protože poté každý destruktivní zásah do čidla nebo kabelové sítě vede k poplachovému stavu „smyčka rozpojená“.

Vstupní vyhodnocovací obvody ústředěn vyššího standardu (stupně zabezpečení dle ČSN EN 50131-1), jsou chytřejší a pracují na principu přesných odporových děličů nebo jako vyvážené měřicí můstky. Toto „chybové“ napětí je přiváděno na obvod napěťového komparátoru, který při překročení určité meze překlápí svůj výstup, a způsobí tak odpovídající reakci v logických obvodech ústředny. Z toho vyplývá, že takový vstupní obvod ústředny musí být zakončen odporem o definované hodnotě a toleranci.

Na vstupu ústředny je připojena smyčka, ta je tvořena kontakty poplachových relé čidel, kabeláží a zakončovacím odporem. Z toho je patrné, že konečná hodnota odporu smyčky je dána hodnotou vlastního zakončovacího odporu zvýšenou navíc o odpor vedení a o hodnoty přechodových odporů kontaktů výstupních relé čidel, ty ovšem v praxi můžeme zanedbat, neboť jsou řádově ve srovnání s hodnotami zakončovacích odporů velice malé. Výrobci často osazují do čidel sériový odpor kontaktu relé řádově 10 ohmů. Pokud zapojujeme větší počet čidel do jedné smyčky, je poté třeba s těmito odpory počítat.

Pro rozsáhlé ústředny je nutné používat kabely s dostatečným průřezem žil nebo vybírat takové ústředny, které pracují s vyššími hodnotami zakončovacího odporu. Oběma způsoby dosáhneme hodnoty odporu spojovacího vedení relativně menší oproti hodnotě zakončovacího odporu a vejde se tak do jeho povolené tolerance. Není-li možné tohoto stavu dosáhnout, poté je nezbytné zvolit takový zakončovací odpor, aby výsledný odpor celé smyčky byl v povolené toleranci.

Mimo standardní tzv. vyvážené diferenciální smyčky se s rozvojem techniky A/D převodníků objevují i systémy s tzv. dvojité vyváženou smyčkou, kdy je možné pomocí jediného smyčkového vedení vyhodnotit jak poplachové hlášení, tak i neoprávněný zásah do čidla či přerušení smyčkového vedení.

Vstupní obvody ústředen jsou náchylné na změny vstupních veličin, proto mohou být citlivé na elektromagnetické rušení. Především u větších kabelových sítí a zvláště potom v průmyslových objektech může toto rušení znemožnit funkci celého systému. Nejen z tohoto důvodu je nezbytné provádět všechny rozvody smyček stíněnými vodiči.

Stínění jednotlivých kabelů je třeba dobře propojit, a to tak, že nesmí tvořit uzavřenou zemní smyčku. V praxi tedy stromovitě propojíme stínění všech kabelových větví a stínění připojíme na svorku PE sítě v ústředně EZS.

Funkční vlastnosti jednotlivých vstupních obvodů ústředen z pohledu hlášení v rámci systému jsou pevně dané (tíseň, sabotáž, vloupání, poplachové smyčky zpožděné, poplachové smyčky okamžité), nebo programovatelné.

Výstupní obvody ústředen

umožňují aktivovat výstupní signalizační a indikační obvody a prvky systému EZS

Patří sem následující výstupy:

- výstup pro akustickou signalizaci
- výstup pro optickou signalizaci
- výstup telefonního voliče
- programovatelné výstupy
- pomocné zvukové výstupy
- výstupy pro periferie
- bezpotenciálové výstupy

Napájecí obvody

Slouží k napájení elektronických obvodů jak ústředny, tak i všech návazných prvků systému EZS. Napájecí zdroj je zálohován náhradním zdrojem napětí (systém musí být funkční i při výpadku proudu). Náhradní zdroj ústředny obsahuje bezúdržbové plynotěsné olověné akumulátory, pomocí kterých lze vytvořit náhradní zdroj s kapacitou 1,2 Ah až řádově 100 Ah. Na režim zálohování se vztahují normy ČSN EN 50131-1 čl. 9.2, tabulka 16 a norma ČSN EN 50131-6 tabulka 2.

Síťový zdroj dodává napětí + 13,8V se zatížitelností 1A až 5A. U rozsáhlých systémů je vhodné z hlediska úbytku napětí na vedení použít více zdrojů s menší zatížitelností.

Ústředna EZS vždy obsahuje dva zdroje -> základní a náhradní:

- základní napájecí zdroj – zdroj el. energie pro trvalé napájení zařízení EZS.
- náhradní napájecí zdroj – zdroj el. energie pro napájení zařízení EZS při výpadku základního zdroje.

Požadavky na napájecí zdroje:

Musí být schopen dodat potřebný proud a být dimenzován tak, aby po skončení nejdelšího výpadku sítě byl schopen dodat potřebný proud nejen pro všechny prvky na zdroj připojené, ale také proud potřebný k dobíjení připojeného akumulátoru nebo akumulátoru během doby stanovené v ČSN 50131-1.

Požadované doby zálohování systému EZS

	Stupeň 1	Stupeň 2	Stupeň 3	Stupeň 4
Minimální doba pohotovosti (hod. Dle ČSN EN 50131-1	12	12	60	60
Minimální doba pohotovosti (hod.) Dle ČSN EN 50131-6	8	15	24	24

Požadované doby nabíjení

	Stupeň 1	Stupeň 2	Stupeň 3	Stupeň 4
Maximální doba dobíjení na min 80% kapacity	72	72	24	24

Základní funkce a rozdělení ústředn EZS

Ústředna elektrické zabezpečovací signalizace je zařízení, které:

- přijímá a vyhodnocuje výstupní elektrické signály od čidel EZS
- ovládá signalizační, přenosová, zapisovací a jiná zařízení, která indikují narušení
- napájí čidla a další prvky EZS elektrickou energií
- pomocí elektromagnetických nebo kódových zámků, popřípadě vlastních ovládacích klávesnic, umožňuje uvedení celého systému EZS nebo jeho části do stavu střežení nebo klidového stavu
- umožňuje diagnostiku systému EZS

Ústředny můžeme rozdělit do čtyř hlavních skupin:

- ústředny smyčkové
- ústředny s přímou adresací čidel
- ústředny smíšeného typu
- ústředny s bezdrátovým přenosem poplachového signálu od čidel

3.10.1 Smyčková ústředna

Ústředna má pro každou poplachovou smyčku vstupní vyhodnocovací obvod, je řešen pro připojení smyček o definované hodnotě a toleranci. Smyčka je zakončena zakončovacím odporem tak, aby vykazovala předepsanou hodnotu odporu pro konkrétní typ ústředny. Pokud dojde v některé smyčce ke změně odporu, znamená to, že někde došlo k narušení a systém vyhlásí poplach. Poplachové smyčky jsou nejčastěji tvořeny sériovým zapojením rozpínacích kontaktů čidel.

3.10.2 Ústředna s přímou adresací čidel

Ústředna pracuje na principu komunikace po datové sběrnice **ústředna – čidla**. Adresy jednotlivých čidel jsou periodicky generovány ústřednou. Ústředna přijímá od čidel příslušné odezvy. Každé čidlo má svůj komunikační modul. Čidla jsou připojena v libovolném pořadí zpravidla na čtyřvodičové vedení. Dva vodiče slouží pro napájení čidla a dva jako datové sběrnice. Velká výhoda tohoto systému je, že pokud dojde k narušení objektu, ústředna oznámí, které konkrétní čidlo bylo aktivované a jaký je druh narušení.

Kabelový systém je zde sice jednoduchý, ale problém nastává v případě, kdybychom chtěli na datovou sběrnici přidat dodatekové funkce čidel. Stejně tak má svá omezení i konfigurace kabelové sítě. Jedním z nich je celková délka vedení, dále je nutné vyvarovat se uzavřených okruhů přes nezanedbatelnou plochu, do nichž by se mohlo indukovat elektromagnetické rušení.

Při návrhu musíme propočítat odběr jednotlivých částí systému a nezapomenout na úbytky na napájecích vodičích. Počet přímo adresovaných čidel se v tomto systému pohybuje v řádově v desítkách.

3.10.3 Ústředny smíšeného typu

Ústředna pracuje na principu datové komunikace **ústředna – koncentrátor** (sběrnice modul smyček). Komunikace mezi těmito dvěma prvky probíhá pomocí datové nebo analogové sběrnice. Na koncentrátory jsou čidla připojena pomocí smyček, stejně jako tomu je u smyčkových ústředen. Vlastní vyhodnocování probíhá různě, záleží na typu ústředny. Jednou variantou je analogový multiplex, kdy se na sběrnici připojují postupně, jednotlivé smyčky a vyhodnocení impedance smyčky s příslušnou odezvou provádí ústředna. Dalším případem je integrace vyhodnocovací logiky včetně vyrovnávací logiky přímo do koncentrátoru. Komunikace zde probíhá v datové podobě.

Umožňuje-li to kapacita ústředny, je možné, na jednotlivé vstupy koncentrátorů připojit přímo jednotlivá čidla. Tak přejde tento typ ústředny na ústřednu s přímou adresací čidel se všemi jejími výhodami. Nevýhodou tohoto systému jsou ovšem náklady na vybudování. Proto je důležité rozvrhnout čidla do smyček tak, aby byla zachována z hlediska uživatele účelná úroveň adresace. Důležitým bodem při návrhu systému je stejně jako u ústředen s přímou adresací, dostatečné dimenzování napájecích i datových vodičů, především u rozsáhlých

systémů je toto důležité (délka jedné datové sběrnice až do 1 km, s opakovači i více). Tato skupina navíc umožňuje realizaci dodatkových funkcí přímo přes datovou sběrnici.

3.10.4 Ústředny s bezdrátovým přenosem od čidel

Skupina ústředen, které se začali rozvíjet v poslední době. Nejčastěji pracují v pásmu telemetrie (868Hz) a jejich výkon je přibližně 10 mW.

Ve volném prostředí je dosah 100 – 200 m. V objektech musíme počítat s menšími vzdálenostmi. Napětí baterie je hlídáno a podle provedení dojde při poklesu napětí k místní akustické signalizaci interním bzučákem nebo se informace přenesou do poplachové ústředny.

Obecně se dají identifikovat u bezdrátových systémů následující výhody:

- rychlá a snadná instalace
- možnost instalace do hotových objektů s minimalizací stavebních zásahů
- snadné rozšíření systému doplněním dalších prvků
- snadná změna konfigurace (jednoduché přemístění detektorů při přestavení nábytku a podobně)

Nejnovější typy bezdrátových systémů pracují obousměrně (duplexně) a každý prvek v systému je vybaven jak vysílací, tak i přijímací elektronikou (modulem vysílač-přijímač). Tyto moduly jsou schopny najít si ve vyhrazeném kmitočtu pásma dva volné kanály pro přenos a automaticky se na ně naladit. Dojde-li k rušení těchto kanálů, jsou schopny se přeladit na jiné, narušené. Tím že tento systém pracuje obousměrně, odpadají nedostatky, které jsou u jednosměrných systémů.

Přednosti systémů s obousměrnou komunikací:

- po zapnutí si ústředna zkontroluje stav všech prvků
- v klidovém stavu čidla nic nevysílají a neplýtvají tak energií, navíc nemusejí být vybavena blokováním dalšího vysílání po vyslaném poplachu
- dálkově lze otestovat funkce
- lze uskutečnit funkci automatického přeladění, v případě, že je pokus o úmyslné nebo i neúmyslné rušení systému
- ústředna může ověřit, jestli je došla poplachová zpráva skutečný poplach, tím se dají vyloučit plané poplachy způsobené rušením

Jako díly bezdrátových systémů se vyrábějí:

- bezdrátová čidla pohybu (většinou PIR)
- bezdrátová tísňová tlačítka
- bezdrátové magnetické kontakty
- univerzální bezdrátové moduly pro připojení libovolných čidel
- stabilní bezdrátové ovládací díly
- mobilní ovládací bezdrátové díly
- bezdrátové sirény atd.

Bezdrátové systémy jsou vhodné, pokud zabezpečovací systém montujeme do objektu dodatečně a chceme minimální montážní hrubé zásahy do stávajícího objektu. Stejně tak se systém uplatní v historických objektech s vzácnými malbami na omítce nebo s kamennými zdmi, které neumožňují vedení kabelů. Při provozu musíme dbát na pravidelnou a častou kontrolu správné funkčnosti systému a také na stav baterií, především při nízkých provozních teplotách.

3.11 Doplnková zařízení ústředny

Pod tímto pojmem si můžeme představit všechna samostatná zařízení, která jsou umístěna přímo v krytu ústředny nebo mimo ni, ale jsou řízena řídicími výstupy ústředny

3.11.1 Akustická signalizace

Nejčastější doplňkové zařízení. Určená může být jak pro vnitřní tak i pro venkovní prostředí. Základ je vždy akustický měnič (piezoelektrický či dynamický) rozšířený o generátor kolísavého tónu a výkonový zesilovač. Akustický výkon by měl být stanoven v technických požadavcích na zařízení (typicky do 120 dB). Délka aktivace sirény je omezena normou ČSN EN 50131-1, ta udává minimální dobu 90 sekund a maximální dobu aktivace 5 minut s odkazem na národní předpisy. Umístění sirény je v průčelí střeženého objektu do výšky, která je pokud možno nedostupná bez použití žebříku či jiného pomocného prostředku.

Nejrozšířenější jsou dnes tzv. „inteligentní“ **sirény s vlastním zálohováním**. Některé sirény mají uvnitř optoelektronický hlídací modul, který pozná, pokud se někdo snaží o zapnění vnitřku skrz akustickou mřížku. Jinak se tento problém dá řešit pomocí dvojitého krytu se speciální konstrukcí.

Případy aktivace „inteligentní sirény“:

- regulérní poplach systému EZS
- přerušení kabelového propojení ústředna – siréna
- pokus o odstranění pláště sirény
- pokus o sejmutí sirény ze zdi

Ukázka sirén



Obr. 3.23 Vnější siréna



Obr. 3.24 Vnitřní siréna

3.11.2 Optická signalizace

Jinak také světelný maják, je nejčastější součástí krytu u venkovních sirén. Po technické stránce se jedná o výkonovou 12 V žárovku, která je buzena přes elektronický přerušovač. Doporučená barva optické signalizace je oranžová. Z hlediska vlastní funkce je účelné, aby řízení umožňovalo časově neomezenou aktivaci v případě vyhlášení poplachu. To z důvodu, aby bylo možno identifikovat narušení objektu i po doznění sirény v případě umístění více střežených objektů nedaleko od sebe.

Ukázka majáků



Obr. 3.25 Příklady optické signalizace

3.11.3 Grafické tablo

Používá se především u rozsáhlých objektů, kde slouží k usnadnění orientace obsluhy při ovládání systému a hlavně v případě vyhlášení poplachu. V podstatě to jsou panely s plánem objektu doplněné o indikační LED kontrolky. V dnešní době se tabla nahrazují monitorem s implementovaným speciálním ovládacím softwarem. Uživatelé s oboru vězeňské služby a armády ale i dnes vyžadují raději grafická tabla a to z důvodu jejich výhod. Výhodou tabla je především snadná orientace a sledování situace současně více osobami při řízení akce při vzniku mimořádných situací.

3.11.4 Tiskárny

Slouží pro potřebu pozdější analýzy událostí, ale také pro případnou kontrolu činnosti obsluhy spojenou s ovládáním rozsáhlých systémů EZS. S rozvojem mikroelektroniky a se zavedením mikropočítačů do řídicích jednotek ústředí EZS a hlavně poklesem cen polovodičových pamětí je archiv událostí dnes již řešen v elektronické podobě. Výhodou je dostatečná kapacita. Možnost připojení tiskárny dnes zůstává pouze pro případy výpisu dějů k zpětné analýze poplachových událostí. U rozsáhlých systémů EZS vybavených řídicím počítačem, se k archivaci používá pevný disk.

3.12 Poplachová přenosová zařízení – komunikátory

Zprostředkovává po zvoleném médiu (vzduch, JTS (veřejná telekomunikační síť, privátní komunikační síť) informaci o stavu systému nebo narušení objektu majiteli nebo na monitorovací pracoviště obsluhy. Přenos poplachových signálů je samostatná úloha vymezená funkcí poplachové přenosového systému.

Ukázka komunikátorů



Obr. 3.26 GSM komunikátor



Obr. 3.27 LAN komunikátor



Obr. 3.28 Telefonní komunikátor



Obr. 3.29 Rádiový komunikátor

3.12.1 Zařízení určená pro informování majitele objektu

Mezi nejjednodušší patří automatické telefonní hlásiče (ATH), ty umožňují na zvolené (naprogramované) telefonní číslo zavolat a informovat o poplachu, okamžitě poté co poplach nastane. Některá zařízení umožňují zvolit více různých čísel a jiná jsou doplněná o kontrolu stavu telefonní linky. Ty z nejmodernějších typů umějí v součinnosti s digitální ústřednou uvolnit obsazené telefonní číslo. Jiná přenosová zařízení umějí poslat prostřednictvím JTS krátkou textovou zprávu (SMS) na mobilní telefon nebo pager. V objektech kde není telefonní linka, je možné využít GSM komunikátorů, ty kromě vyslání poplachové zprávy umožňují například dálkové ovládání některých elektrických zařízení nainstalovaných v objektu.

Problémem u GSM komunikace je to, že mobilní operátoři nerozlišují priority zpráv a tak může zpráva o případném poplachu v objektu přijít se zpožděním.

3. 12. 2 Zařízení určená pro komunikaci s PCO/PPC

Slouží k předávání informací o systému EZS monitorovacím pracovištím hlídacích služeb (PCO/PPC). Zařízení na principu automatické telefonní volby (ATV) komunikují s obsluhou PCO nikoli hlasitým vzkazem, ale pomocí kódu, který nese adresu objektu a druh předávané zprávy. Jsou závislé na přítomnosti JTS. V určitých případech je ne nutné použít dvě přenosové cesty, JTS a elektromagnetické pole. Nejdokonalejší systémy umožňují oboustrannou komunikaci, a tím možnost monitorování stavu přenosové cesty. Formáty přenosu po JTS jsou normalizovány. Zvolený formát musíme volit v závislosti na zařízení (PCO), to z důvodu, aby si ústředna podporovala daný formát, jinak by spolu tyto dva prvky nekomunikovaly. U bezdrátových systémů se používají standardizované formáty.

3.13 Indikační a ovládací zařízení

Funkce a varianty ovládání

K správné a plnohodnotné funkci systému, musí být umožněno uvádět systém jak do stavu střežení, tak i naopak do stavu klidového. K tomu jsou právě určena ovládací zařízení. Vhodný typ ovládacího zařízení, je volen podle stupně zabezpečení a požadavků vlastníka. Cílem je, aby obsluha byla jednoduchá a byla minimalizována možnost vyvolat při manipulaci planý poplach. Způsobů jak přepínat stavy ze střežení, do stavu klidu, je mnoho a záleží na typu ústředny.

K čemu navíc slouží ovládací zařízení

- připínání a odepínání smyček
- volby speciálních funkcí
- zadávání uživatelských kódů pro ovládání systému
- programování instalačních parametrů systému
- odstavení a resetování poplachů atd.

Cílem indikačních prvků je informovat o provozních stavech ústředny a celé EZS, a to buď opticky nebo pomocí LED, či pomocí akustické signalizace. Úroveň indikace do značné míry podmiňuje úroveň diagnostiky systému. Diagnostika umožňuje selekci závad v EZS, ale i ve vlastní ústředně EZS a tím usnadňuje případné opravy.

Společné znaky indikace různých výrobců

- hlášení stavu střežení/klid
- hlášení připravenosti k uvedení systému do stavu střežení
- hlášení poruchy
- hlášení narušení smyček
- hlášení tísně
- hlášení poplachu

3.13.1 Kódové klávesnice

Mohou být využity jako spínací zámky. Pokud chceme tento prvek používat jako ovládací díl EZS, je nezbytné, aby elektronika klávesnice byla umístěna v samostatné skříni a ve střežených prostorech. Pro uživatele kódové klávesnice přináší nutnost zapamatovat si správný přístupový kód. Prevencí je po určité době změnit kód, jelikož stoupá riziko jeho prozrazení a fyzické opotřebení tlačítek, které snižuje počet kombinací kódu, toho by mohli zneužít lupiči. Výhodou je možné použití tísňového kódu, dojde-li k rizikovému vstupu do objektu. Lépe vybavené typy lze používat také jako ovládání elektrického vrátného nebo ovládání osvětlení.

Ukázka klávesnic



Obr. 3.30 Ovládací a programovací klávesnice

3.13.2 Kombinované indikační a ovládací díly

Spolu s ovládáním systému umožňuje indikovat informace o systému. Kombinované a ovládací díly se používají i pro nejvyšší úroveň rizik. Uplatnění naleznou i jako indikační tablo pro potřeby uživatele nebo najeté hlídací služby. Připojení k ústředně je řešeno pomocí datové sběrnice, která je pro větší rizika elektronicky hlídána proti napadení.

3.13.3 Ovládání kartou

Na trhu se začínají objevovat poplachové ústředny EZS, které umožňují spolupráci se čtečkou identifikačních karet (kontaktních, bezkontaktních). Tento systém je užíván především u integrovaných systémů, kde musí v rámci celého komplexu spolu korespondovat EZS, CCTV, přístupový systém, EPS a další slaboproudé systémy. Cílem takového snažení je multifunkčnost, karta by měla v komplexu sloužit k čemukoliv, jako například k vjezdu do garáže, ohlášení příchodu na pracoviště, odpojení střežení své kanceláře, pohybování po objektu, objednávání obědů v jídelně, nakupování v jídelně atd.

Ukázka čteček



Obr. 3.31 Čtečka karet a čipů JA-80N

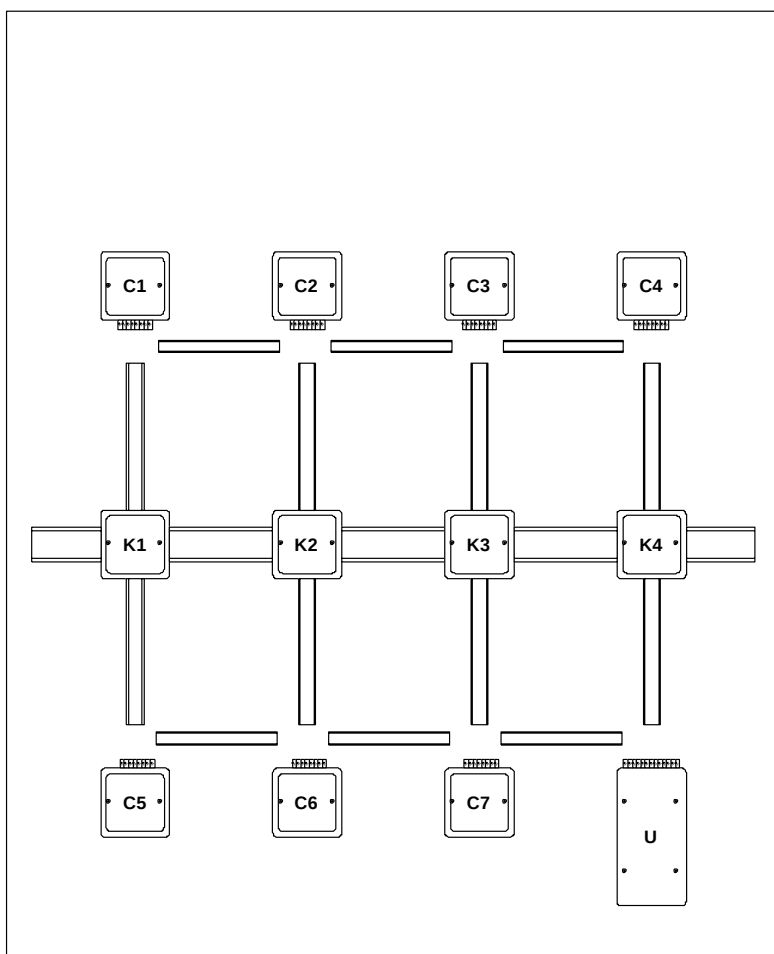


Obr. 3.32 Přístupová karta a čip firmy Jablotron

3.14 Konstrukční cvičení

Popis zapojení

Při výuce EZS je většině škol využíváno panelů, na kterých je zpravidla stabilně rozmístěn různý počet čidel a ústředna s příslušenstvím. Tento systém má nevýhodu v nemožnosti měnit počet jednotlivých prvků a jejich rozmístění. Problémem je i jednoznačná aktivace určitého čidla při ověřování činnosti. Žáci rychle ztrácejí zájem zapojovat stále stejnou úlohu. Navíc je toto řešení i poměrně nákladné. Naší snahou bylo vytvořit snadno modifikovatelný systém, ve kterém jsou skutečná čidla i ústředna nahrazena jednoduchými prvky připevněnými pomocí suchého zipu, které jejich činnost pouze simulují. Hlavním účelem tohoto řešení je žáky na různých úlohách naučit propojování čidel s ústřednou s využitím jednoduše i dvojité vyvažovaných smyček. Panel je tak využitelný pro výuku propojování komponentů EZS od nejjednodušších základních zapojení až po rozsáhlejší systémy. Panely umožňují připojení až 7 různých čidel, což odpovídá reálnému zabezpečení menšího objektu. Na panelech jsou připraveny instalační lišty pro vedení propojovacích kabelů. Chceme-li simulovat rozsáhlejší systém, můžeme propojit více panelů mezi sebou.



Legenda:

C1 – C7 čidla

K1 – K4 instalační krabice JB-16

U – ústředna EZS

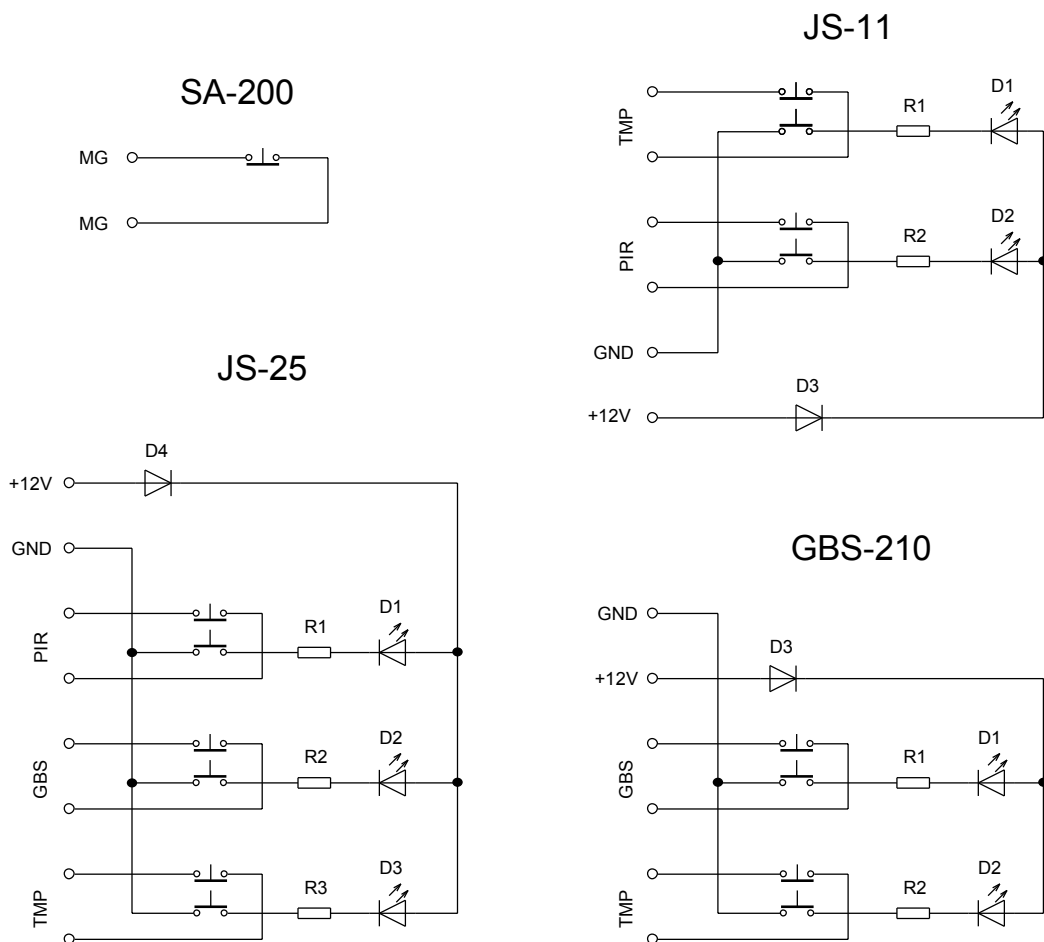
Simulace čidel

Čidla jsou nahrazena pouze tlačítky, která po stisknutí simulují jejich aktivaci. Aktivní čidla (PIR a GBS) obsahují navíc indikaci stisku tlačítka pomocí LED diody. Současně je kontrolována přítomnost i správná polarita napájecího napětí.

Výhody řešení:

- jednoduchá a jednoznačná aktivace čidla nebo jeho části s optickou indikací u aktivních čidel
- svorky odpovídají popisem i pořadím skutečným čidlům a jsou vyvedeny mimo kryt, takže odpadají problémy se snímáním krytu při montáži a aktivaci ochranného kontaktu
- jednoduché konstrukční řešení s použitím běžných součástek a dílů
- nižší pořizovací náklady v porovnání se skutečnými čidly

Schéma zapojení:

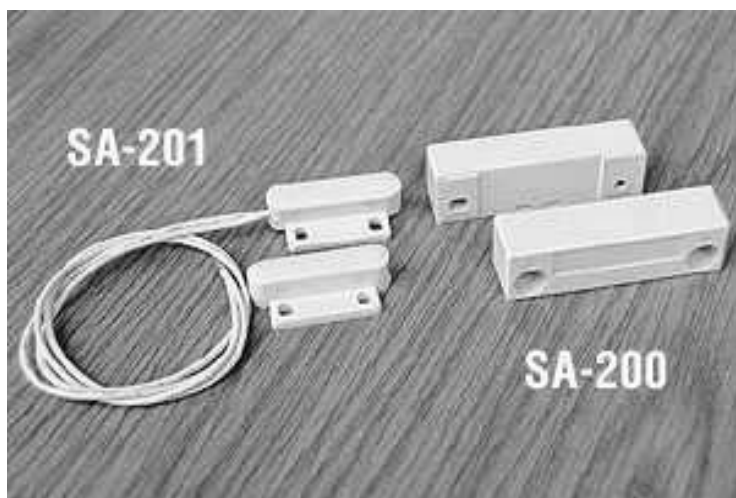


Konstrukční řešení:

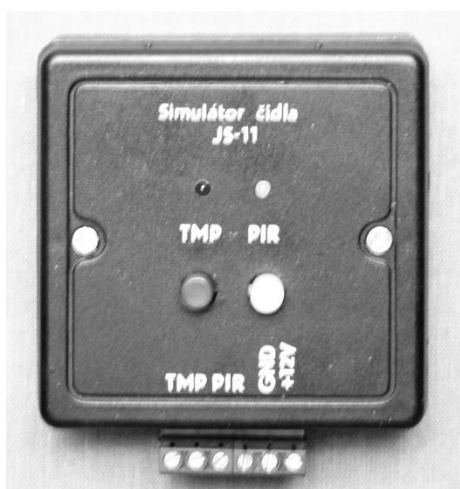
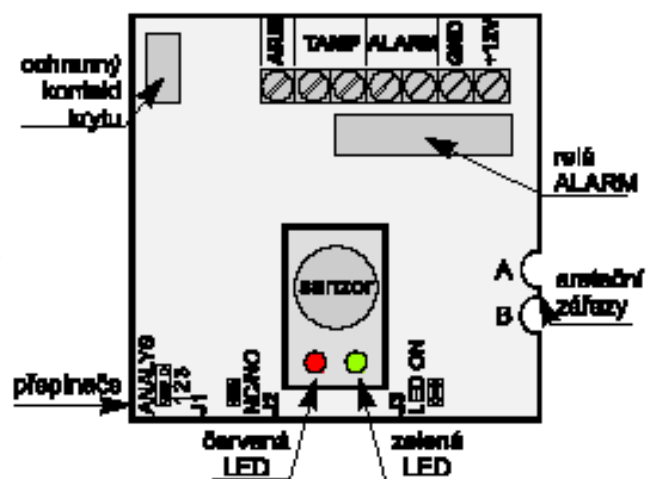
Všechny součástky každého čidla jsou umístěny na jedné desce plošných spojů, která je dvěma šroubky upevněna v běžné instalační krabici s víčkem. V krabici je třeba vypilovat zářez pro svorkovnici a ve víčku vyvrtat otvory pro indikační LED diody a ovládací tlačítka. Povrchovou úpravu je možné provést nastříkáním laku a popisy Propisotem nebo samolepicími štítky s potiskem.

Ukázky provedení a porovnání se skutečnými čidly:

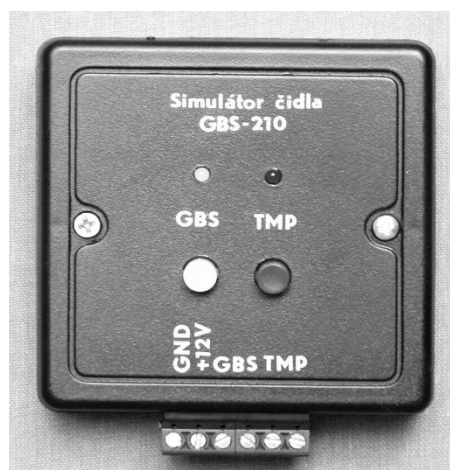
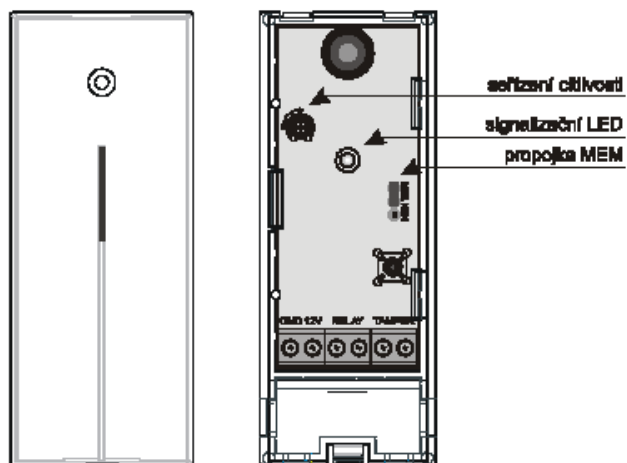
a) Magnetický dveřní kontakt SA-200



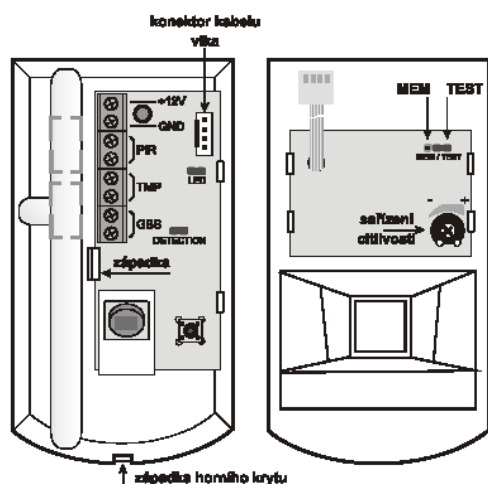
b) Snímač pohybu JS-11 (PIR čidlo)



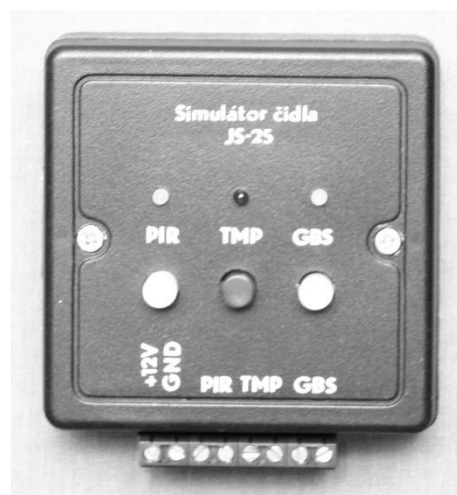
c) Akustický detektor rozbití skla GBS-210 (Glassbreak)



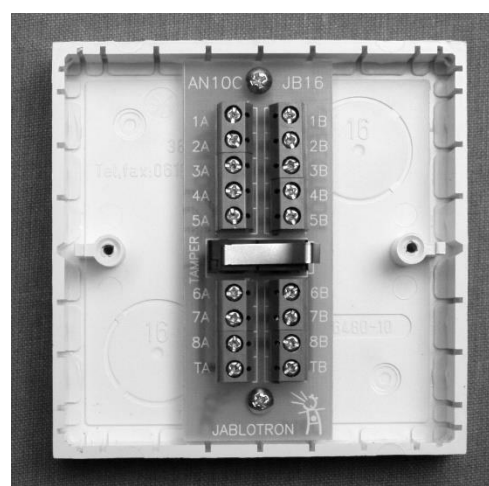
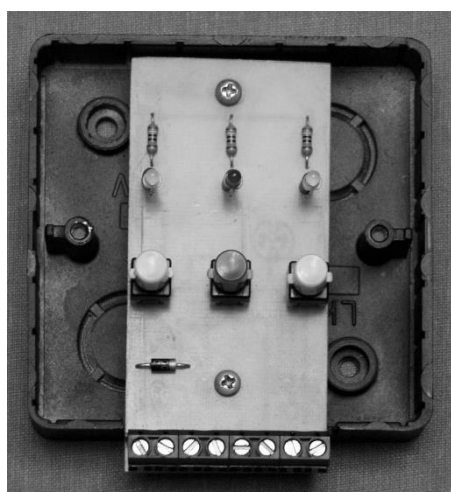
d) Kombinovaný detektor JS-25 (PIR čidlo + Glassbreak)



e) Detail upevnění desky PS v krabici



f) Svorkovnice JS-16 (Jablotron)



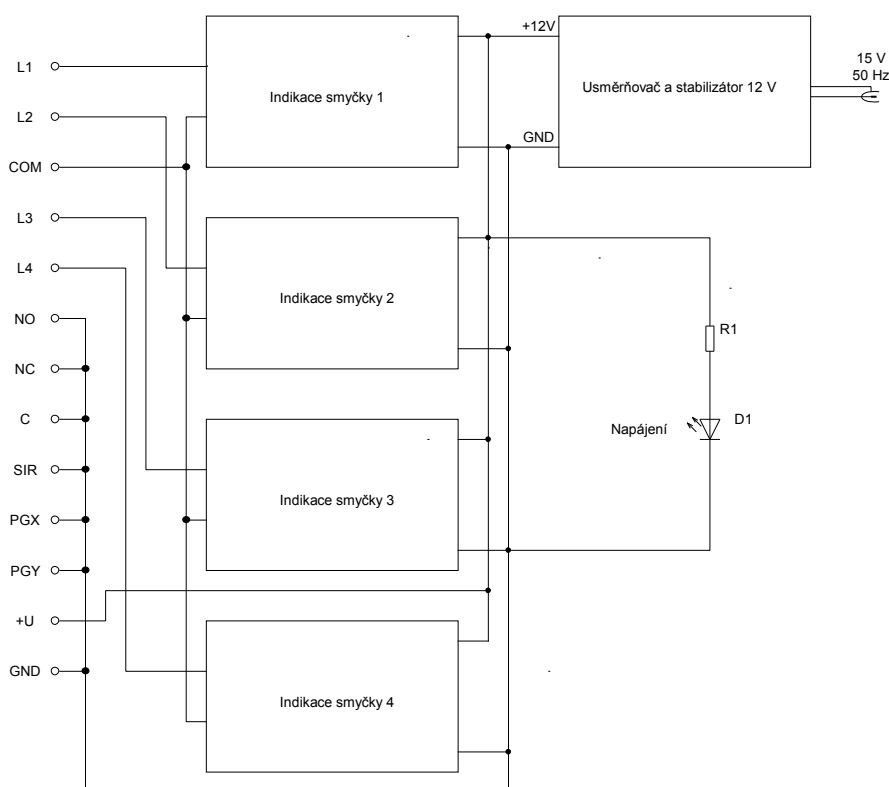
Simulace ústředny

Ústředna je nahrazena elektronickým obvodem, který pomocí LED diod indikuje stav celkem čtyř smyček (zkrat, vyvážení, přerušení). K napájení slouží síťový adaptér (např. DE20-15 firmy Jablotron). Pořadí i popis svorek odpovídá ústředně EZS JA-63. Nevyužité svorky jsou propojeny s nulovým potenciálem (GND). Střídavé napájecí napětí ze síťového adaptéru je usměrněno můstkovým usměrňovačem, filtrováno a stabilizováno na hodnotu 12 V. Přítomnost napájecího napětí je indikována LED diodou (Napájení). Jsou-li výstupní napájecí svorky (+U a GND) zkratovány, LED dioda nesvítí. K simulátoru lze stejně jako k ústředně JA-63 připojovat jak jednodporově tak dvouodporově vyvažované smyčky. K vyhodnocení stavu linky nebylo využito komparátorů, jak je u ústředěn EZS běžné, ale z důvodu jednoduchosti konstrukce obvodu LM 3914. Tento obvod slouží jako budič 10 LED diod v páskovém nebo

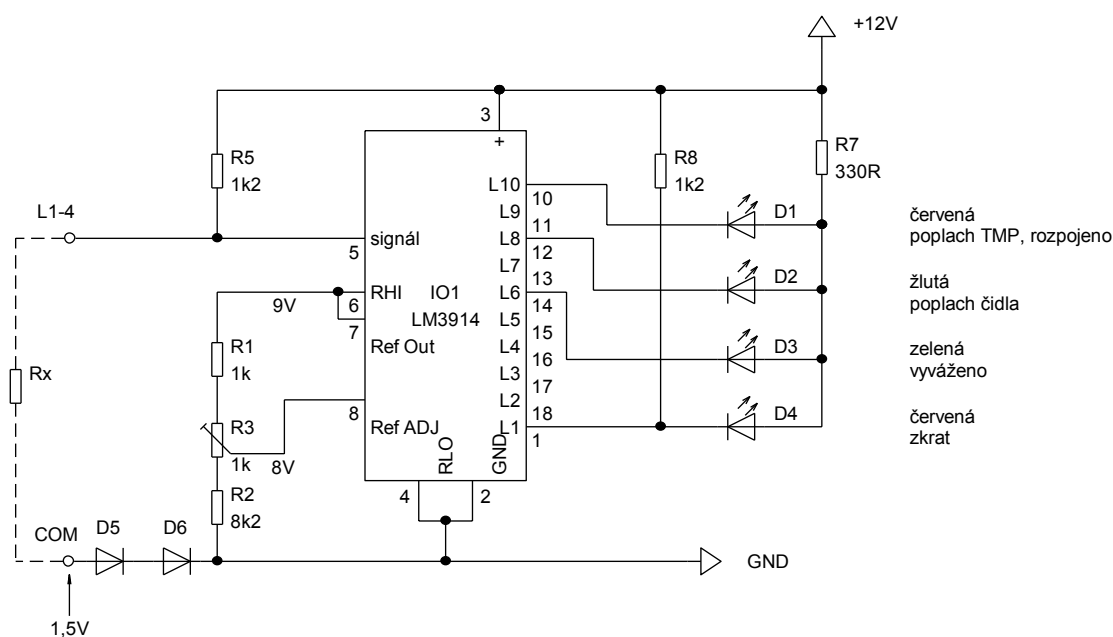
bodovém režimu. V zapojení jsou využity pouze čtyři LED diody, indikující zkrat, vyvážení, poplach čidla a poplach temperu nebo rozpojenou smyčku u dvouodporově vyvážené smyčky. Je-li smyčka vyvážena pouze jednoodporově, jsou funkční jen tři LED diody indikující zkrat, vyvážení a poplach temperu nebo rozpojenou smyčku. Aby bylo možno indikovat zkrat bylo třeba posunout potenciál na svorce COM o cca 1,3 V.

Svorky COM a GND mají tedy různý potenciál a nelze je mezi sebou propojit!

Blokové schéma:



Indikace stavu smyček



Popis zapojení:

Rezistor R5 spolu s rezistorem Rx tvoří dělič napětí, který určuje úroveň vstupního napětí indikátoru. Odporovým trimrem R3 nastavíme takové referenční napětí, aby po připojení $R_x = 1,1 \text{ k}\Omega$ (dva rezistory $2\text{k}2$ paralelně) svítla LED dioda D3, odpovídající vyvážené smyčce. Trimr nastavíme na střed oblasti, kdy je LED dioda rozsvícená. Zkontrolujeme, zda se při hodnotě $R_x = 2,2 \text{ k}\Omega$ rozsvítí LED dioda D2, odpovídající poplachu čidla. Dále zkontrolujeme, zda se při zkratu nebo rozpojení smyčky rozsvítí odpovídající LED diody.

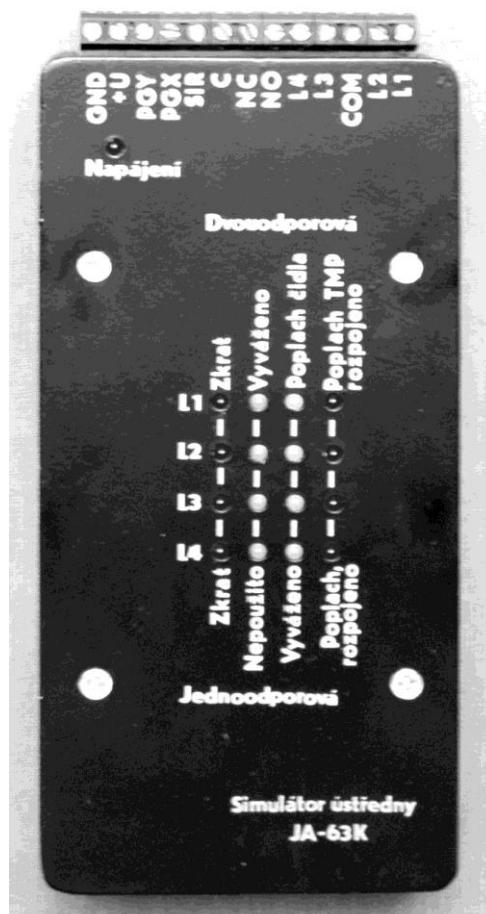
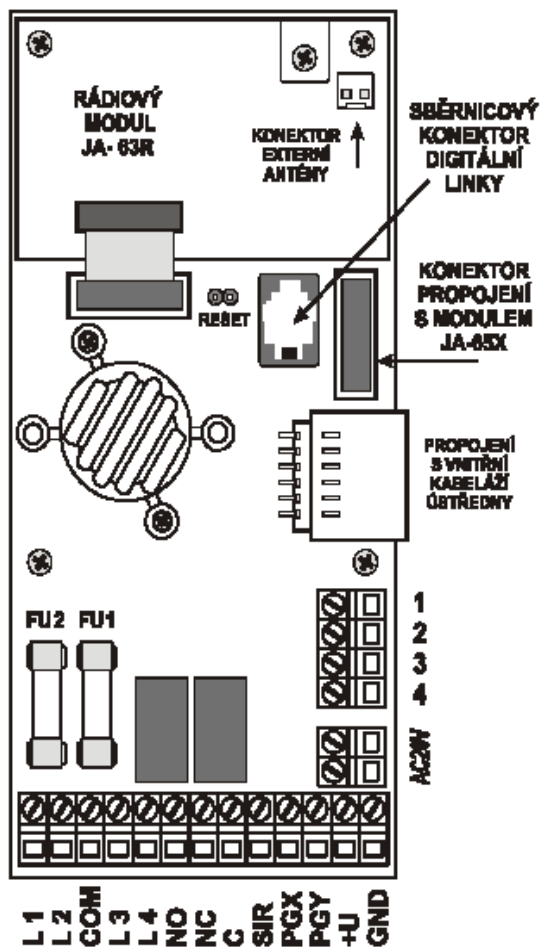
K indikaci stavu smyček bylo použito doporučené zapojení obvodu LM3914, které bylo nutno poněkud upravit:

1. Aby bylo možno indikovat zkrat na vedení, bylo třeba posunout úroveň vstupního napětí odpovídajícího zkratu. To je zabezpečeno diodami D5 a D6 zapojenými v propustném směru. Je-li $R_x = 0$ (zkrat na lince), objeví se na vstupu indikátoru napětí cca $1,3 \text{ V}$ a rozsvítí se LED dioda D4.
2. Přidaný rezistor R8 zajistí, aby při rozsvícení D4 byly zhasnuty ostatní LED diody.

Konstrukční řešení:

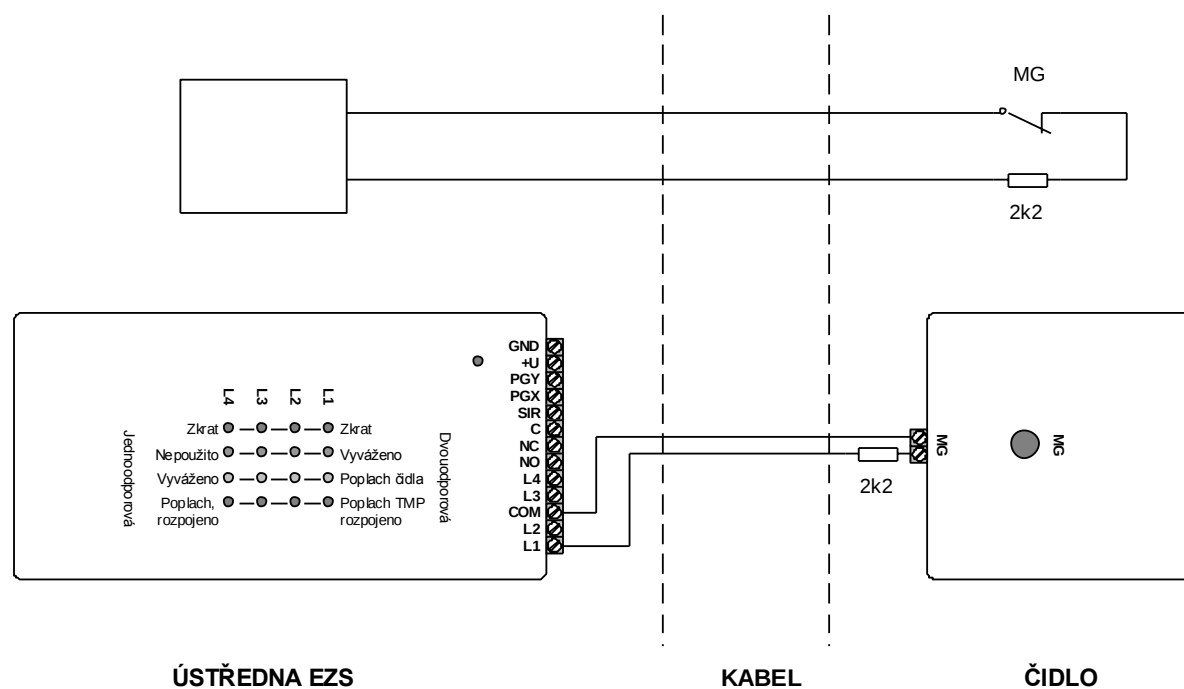
Osazená deska PS je zabudována do upravené dvojité instalační krabice. Úprava spočívá v odfrézování výlisků na bocích krabice a vypilování zářezu pro svorkovnici. Protože se ke krabici nedodává víčko, je potřeba z vhodného materiálu (laminát nebo PVC) víčko vyrobit a

vyvrtat v něm potřebné upevňovací otvory a otvory pro LED diody. Povrchovou úpravu je možné provést nastříkáním laku a popisy Propisotem nebo samolepícími štítky s potiskem.

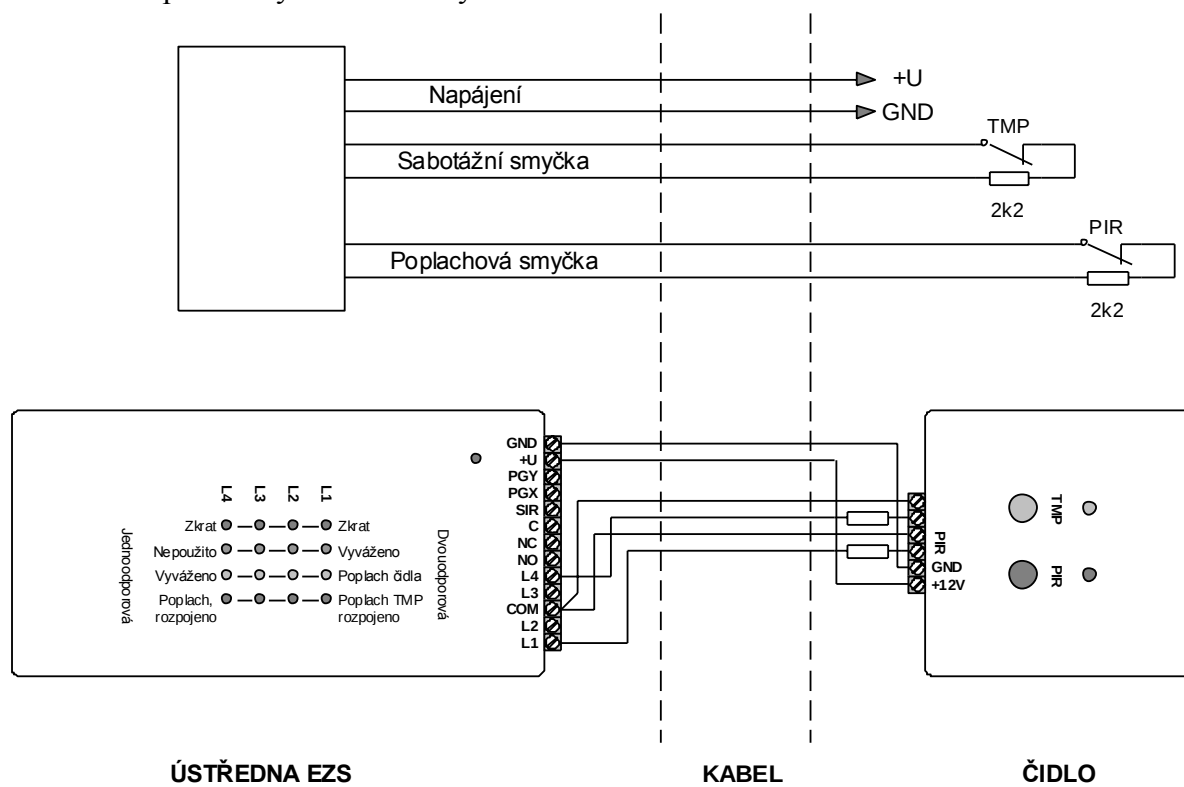


Příklady cvičných zapojení:

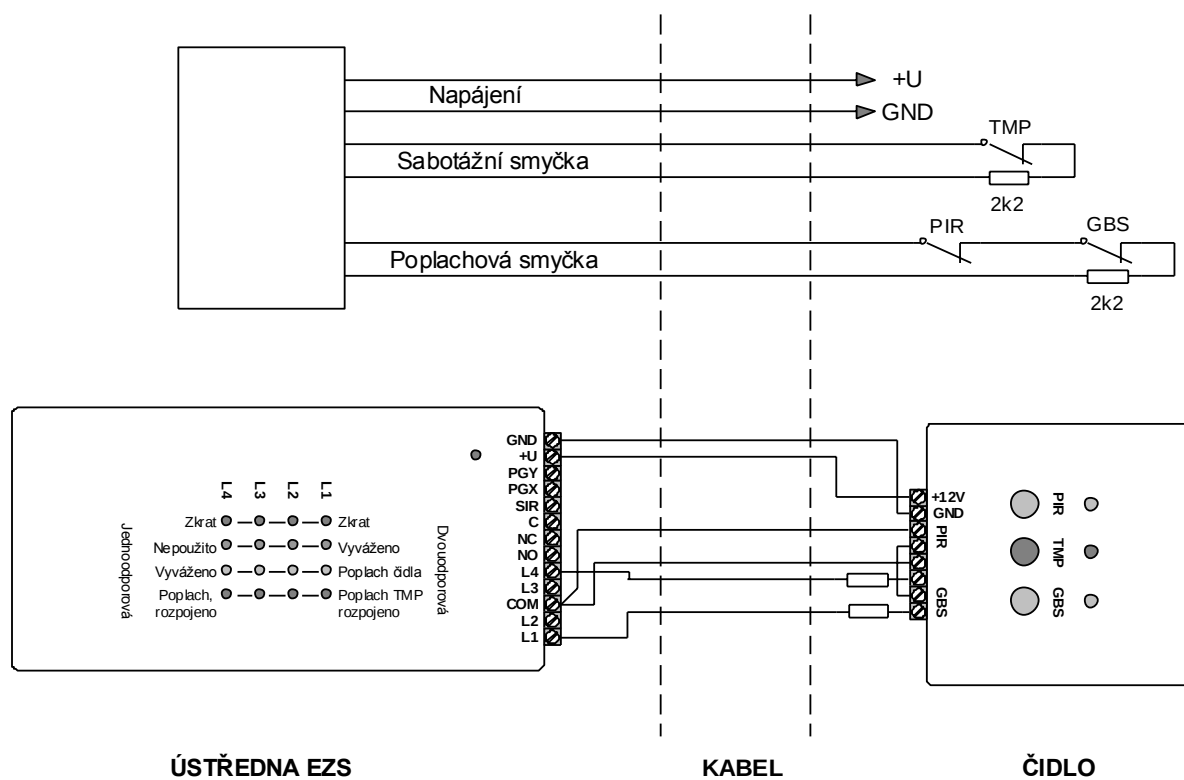
1. Jednoodporově vyvažovaná smyčka s magnetickým snímačem



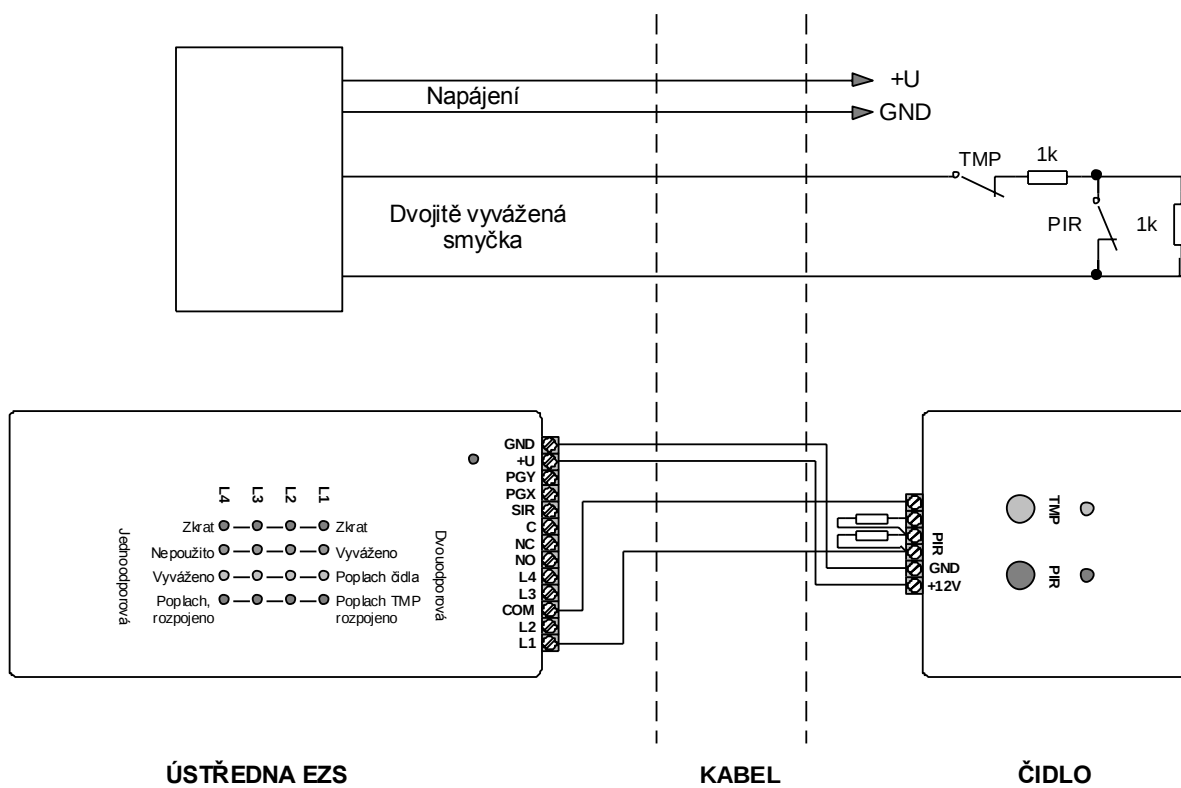
2. Jednoodporově vyvažovaná smyčka s PIR čidlem



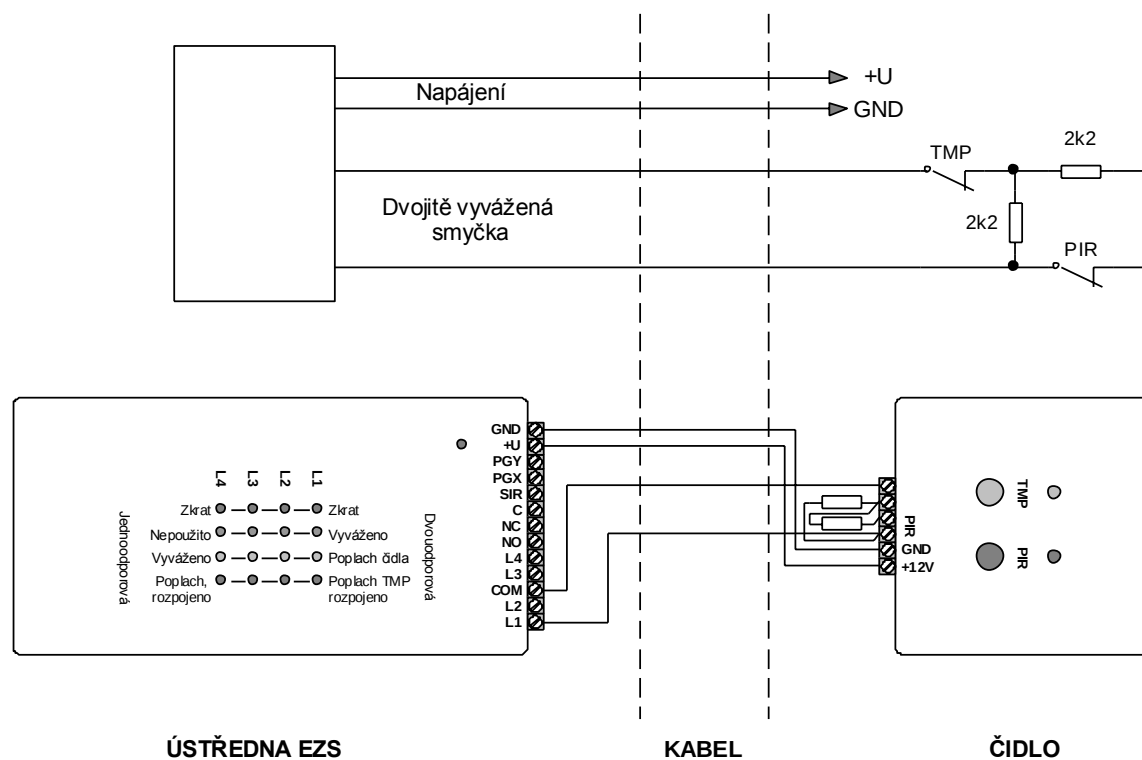
3. Jednoodporově vyvažovaná smyčka s kombinovaným čidlem



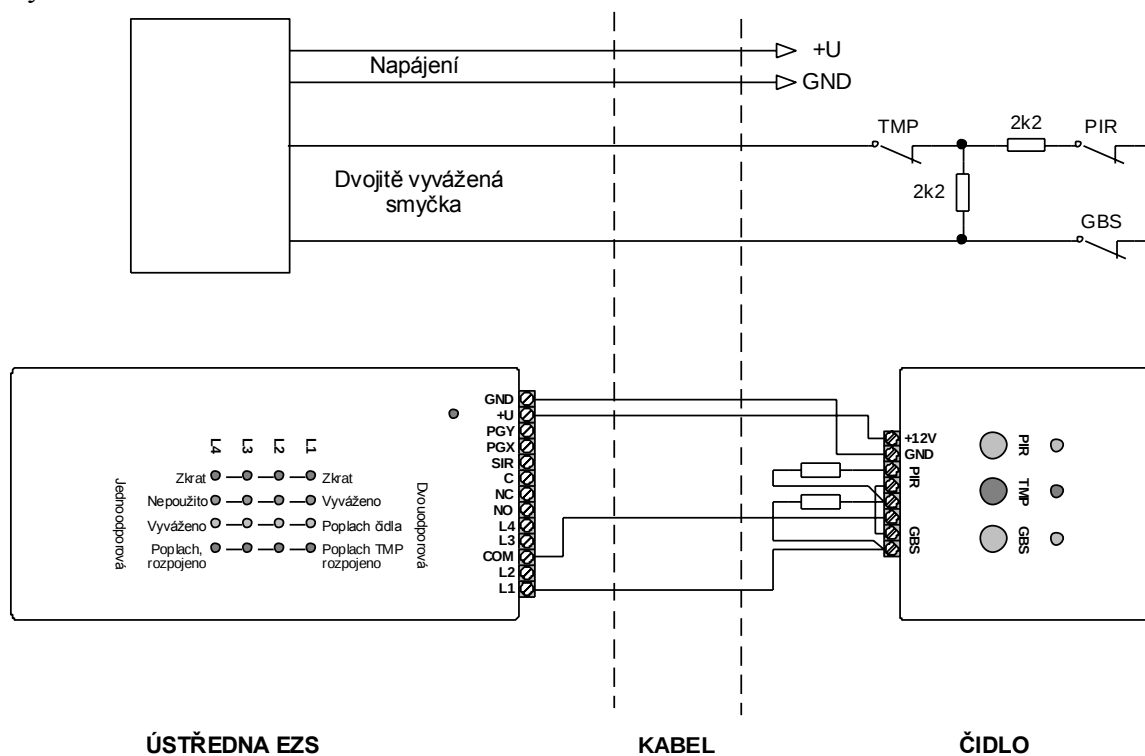
4. Dvouodporově vyvažovaná smyčka s PIR čidlem a sériovým řazením vyvažovacích rezistorů



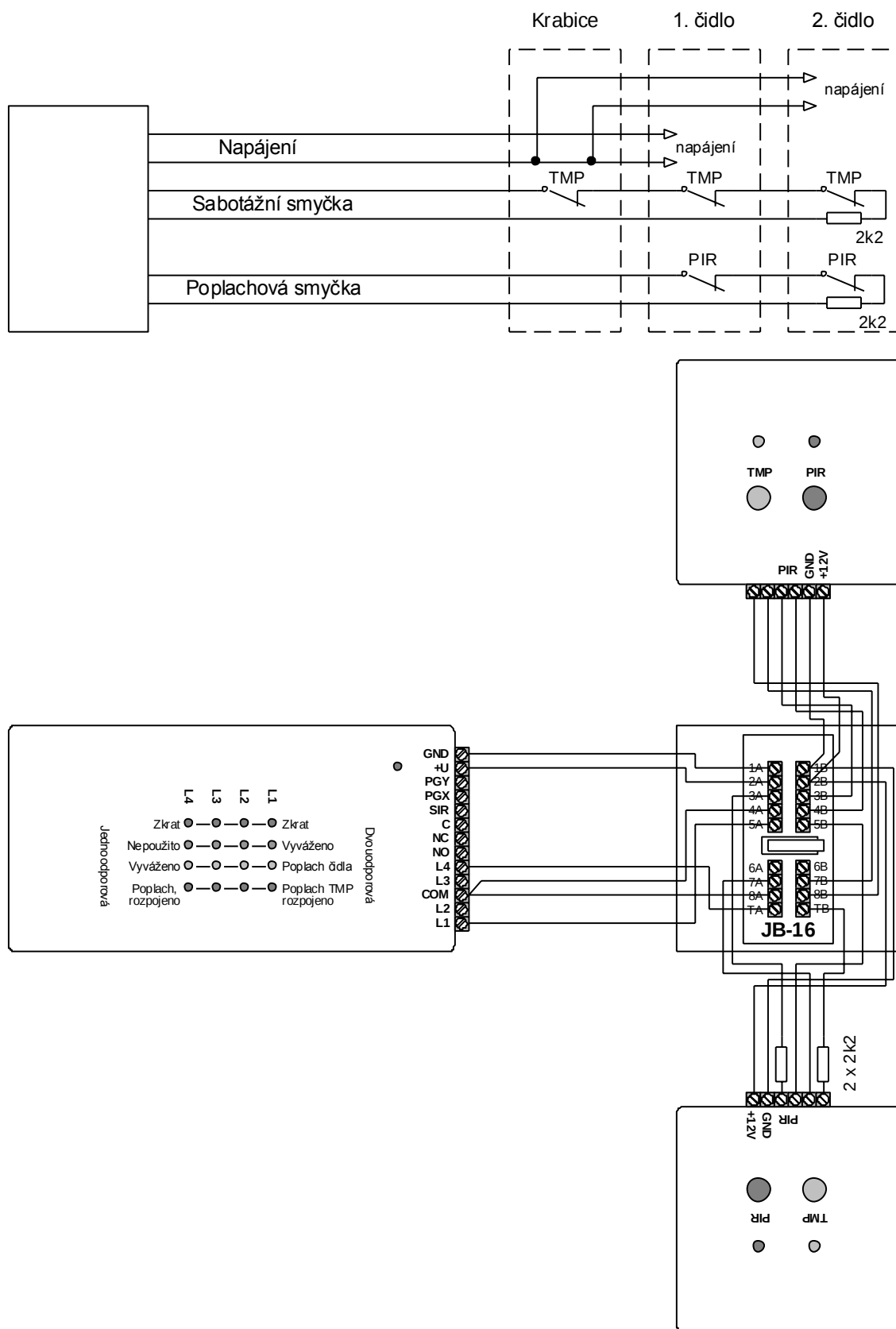
5. Dvouodporově vyvažovaná smyčka s PIR čidlem a paralelním řazením vyvažovacích rezistorů. Výhodou tohoto zapojení je, že hodnoty rezistorů jsou pro jednoduché i dvojité vyvážení shodné.



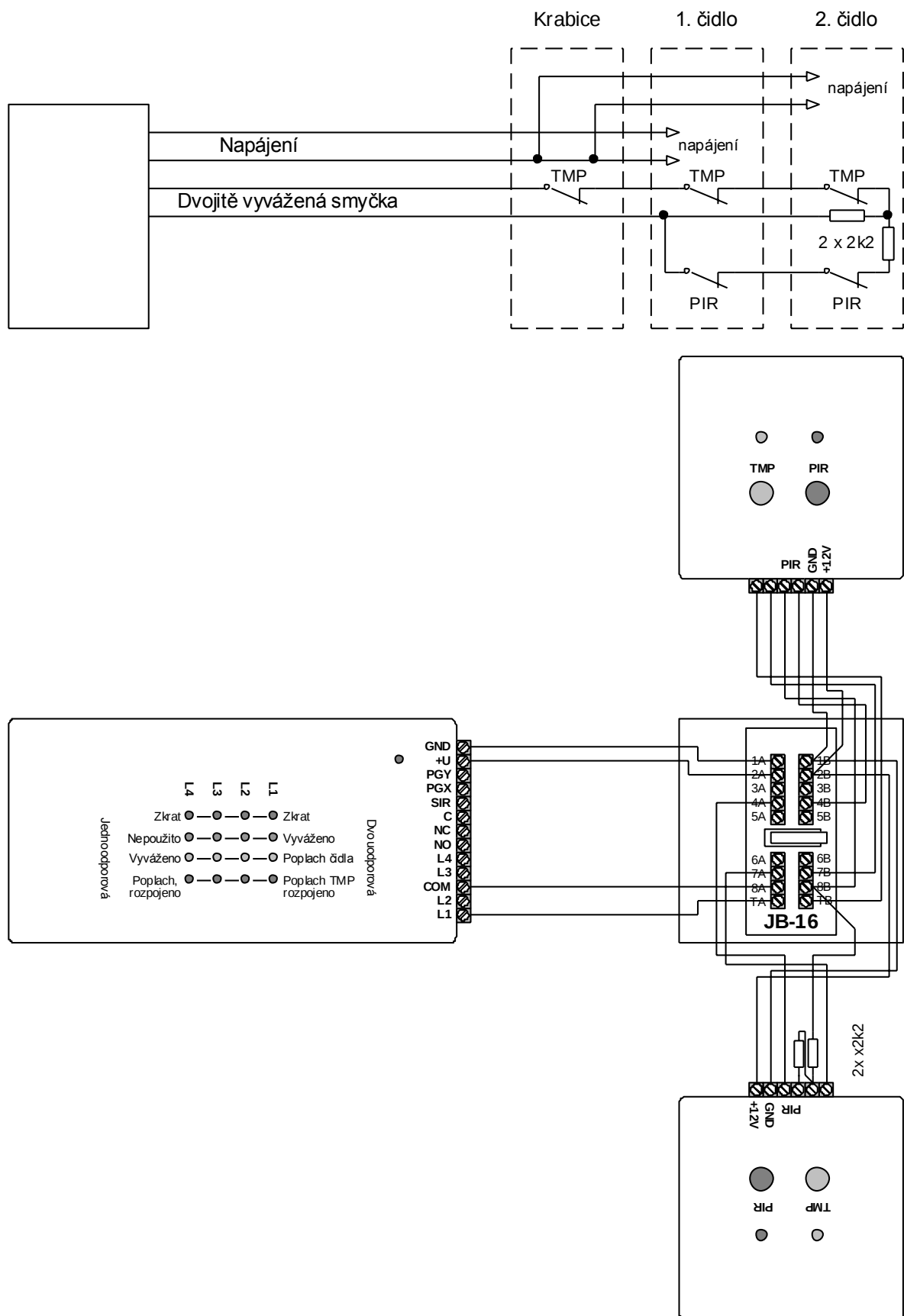
6. Dvouodporově vyvažovaná smyčka s kombinovaným čidlem a paralelním řazením vyvažovacích rezistorů.



7. Zapojení dvou čidel v jednoduše vyvažované smyčce s použitím instalační krabice

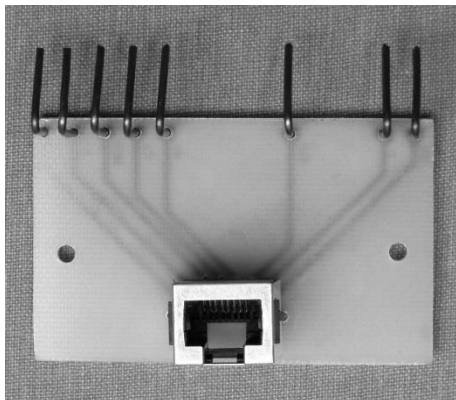


8. Zapojení dvou čidel v dvouduše vyvažované smyčce s použitím instalační krabice



Možnosti rozšíření:

1. K simulátoru ústředny lze připojovat reálná drátová čidla (viz příloha)
2. Simulátor ústředny lze nahradit jednoduchým přípravkem a propojit s ústřednou JA-



63K v základní konfiguraci a propojit jí s počítačem. Následující obrázek ukazuje příklad možného řešení. Drátové vývody jsou připojeny ke svorkovnici základní desky ústředny. Vyvedeno je pouze napájení (+U a GND) a vstupy drátových smyček (L1, L2, L3, L4 a COM) na konektor typu RJ-45. Otvory v desce plošného spoje slouží k upevnění v plastové skříni ústředny. K indikaci poplachu lze použít malou piezosirénu nebo LED diodu

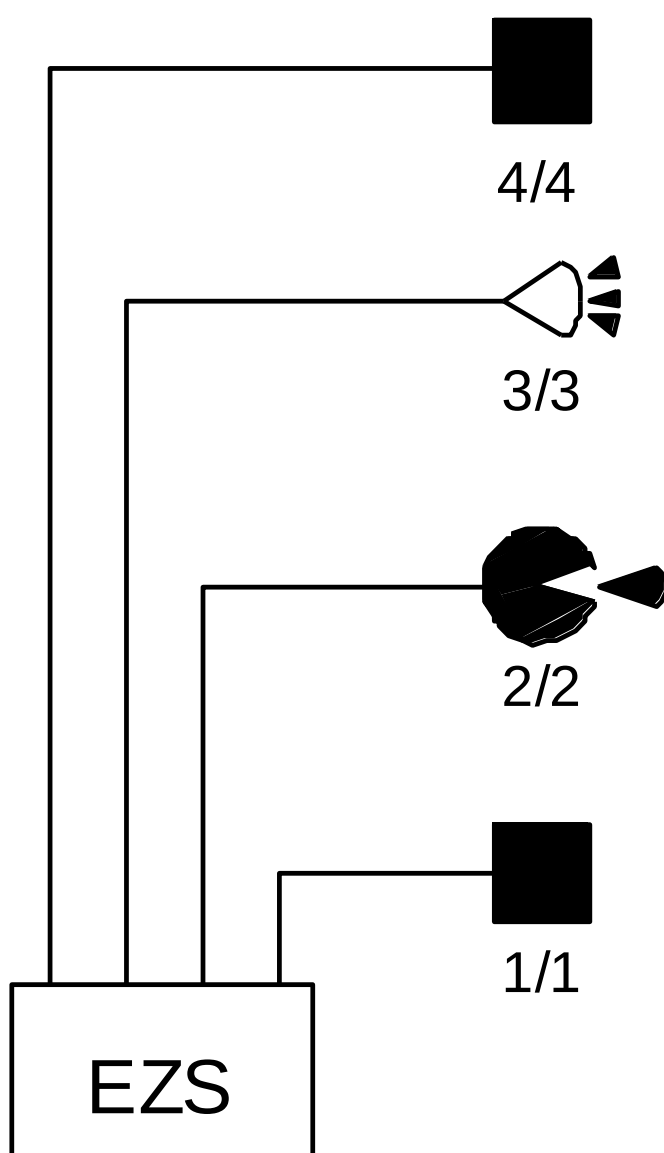
s ochranným rezistorem připojenou mezi svorky SIR a +U.

Pomocí programu COMLINK můžeme ústřednu ovládat a programovat bez použití klávesnice, číst data z paměti ústředny, nastavovat systém, apod.

3.15 Praktická úloha

- ÚKOL:** 1) Zapojte prvky EZS dle blokového schématu
- 2) Nastavte ústřednu JA-80 dle zadaných parametrů.
- 3) Zpracujte test

BLOKOVÉ SCHÉMA:



NASTAVENÍ ÚSTŘEDNY JA-80:

1. Proved'te reset ústředny
2. Aktualizujte datum a čas
3. Nastavte:
 - příchodové zpoždění 30s
 - odchodové zpoždění 20s
 - hlídání rušení ústředny
 - potvrzení zajištění sirénou
 - tel. číslo pro zasílání poplachových SMS zpráv
 - tel. číslo pro zasílání poruchových SMS zpráv
 - zapnutí výstupu PGX pro el. topení
 - změňte text příkazu
 - na pozici č. 4 klávesnici JA-80F
 - na pozici č. 5 magnet JA-80M (reakce zpožděná)
 - na pozici č. 6 PIR JA-80P (reakce následně zpožděná)
 - na pozici č. 7 požární detektor JA-80S
 - drátový magnet ve funkci garážových vrat
 - klíčenku ve funkci tísňového tlačítka

3.16 Opakovací otázky

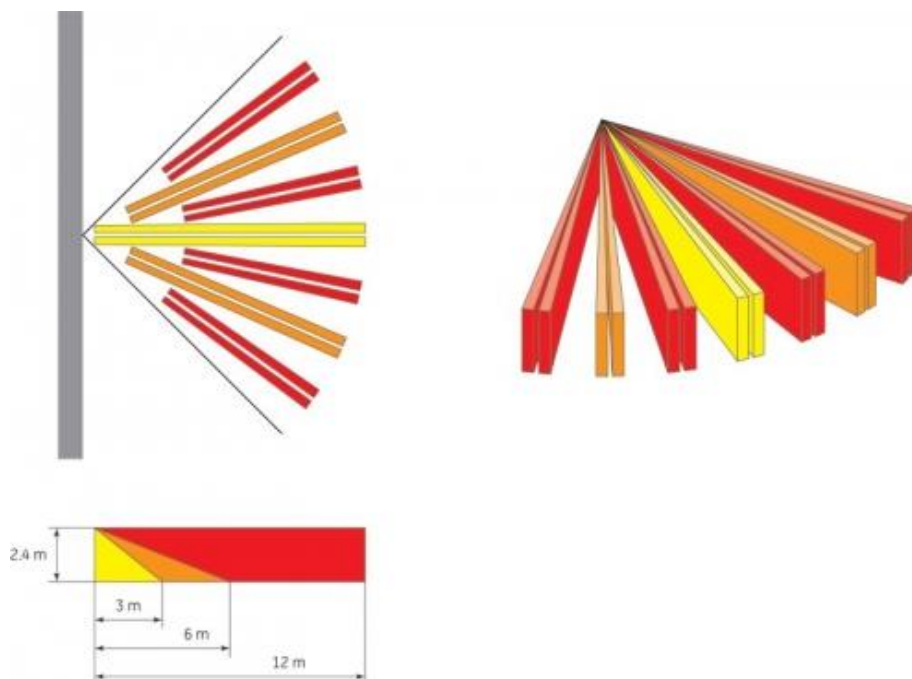
3.16.1 Detektory infračerveného záření

Infračervené záření (také IR, z anglického *infrared*) je elektromagnetické záření s vlnovou délkou větší než viditelné světlo, ale menší než mikrovlnné záření. Název značí „pod červenou“ (z latiny *infra* = "pod"). Infračervené záření zabírá ve spektru 3 dekády a má vlnovou délku mezi 760 nm a 1 mm. Infračervené záření je často považováno za „tepelné záření“. IR záření zapříčiňuje pouze přibližně 50 % zahřívání zemského povrchu, zbytek je způsoben viditelným světlem. Je však pravdou, že objekty při pokojové teplotě emitují nejvíce záření v infračerveném pásmu 8–12 μm . Infračervené záření se používá pro přenos informací na krátkou vzdálenost, nejčastěji podle standardu IrDA. Příkladem mohou být mobilní telefony či dálkové ovladače. Infračervené záření v nich vysílají LED.

Označována jako "PIR" jsou nejčastěji využívanými detektory ve standardních zapojeních elektrické zabezpečovací signalizace. Základem tohoto detektoru je Fresnelova čočka.

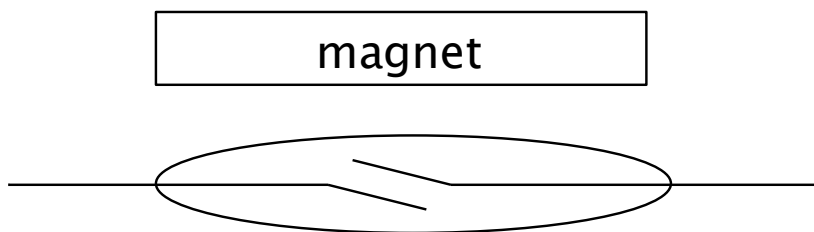
Zjednodušeně lze říci, že PIR detektory jsou schopny zachytit pohyb těles, která mají jinou teplotu, než teplotu okolí. Jejich funkce je založena na zachycení změn vyzařování v infračerveném pásmu kmitočtového spektra elektromagnetického vlnění.

Princip zachycení pohybu PIR čidlem:



Obr. 1

1. Magnetické detektory



Obr. 2 Princip magnetického detektoru

Jsou sestaveny ze dvou částí, permanentního magnetu a jazýčkového kontaktu. Kontakt je uzavřen ve skleněné trubičce naplněné inertním plynem. Přiblížení permanentního magnetu do blízkosti kontaktu způsobí jeho sepnutí. Při oddálení naopak dojde k rozpojení a tím k aktivaci čidla. Magnetická čidla se využívají převážně k indikaci otevření vstupů do budov, a to dveří, vrat a oken. Permanentní magnet se instaluje na pohyblivou a kontakt na pevnou část okna nebo dveří.

Detektory kouře

Optický detektor kouře pracuje na principu rozptýleného infračerveného světla v bezodrazové komoře. Je velmi citlivý na větší částice, které jsou v hustých dýmech, méně citlivý je na malé částice vznikající hořením kapalin, jako je například alkohol.

Další možností jak detekovat rozptýlené částice je měření vodivosti ionizovaného vzduchu. K tomu slouží požární detektory ionizační. Vzhledem k tomu, že ke své činnosti využívá radioaktivní materiál, nepoužívají se pro instalace v domácnostech.

Vyvážená smyčka

3.16.2 Jednoduše vyvážená smyčka

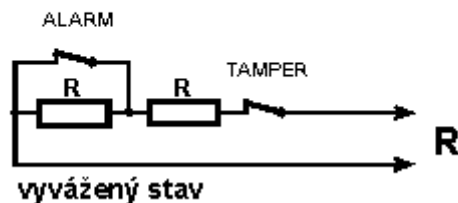
Používá se většinou tam, kde je zapojeno více detektorů v jedné smyčce. Kontakty jsou pak zapojeny v sérii a vyvažovací odpor je umístěn u nejvzdálenějšího kontaktu (v detektoru). Zapojení je jednoduché a průhledné. Nevýhodou je právě to, že je více detektorů v sérii a tedy není přesná identifikace místa aktivace. Pokud bychom tuto přesnou indikaci chtěli mít, spotřebovali bychom jeden detektor smyčky dvě - jednu na indikaci aktivace detektoru, druhou na indikaci sabotáže.

3.16.3 Dvojitě vyvážená smyčka

Pokud má ústředna dostatečný počet drátových smyček, pak je výhodné každý detektor připojit na samostatnou smyčku. Smyčka je pak schopna indikovat jak aktivaci, tak i sabotáž detektoru případně smyčky.

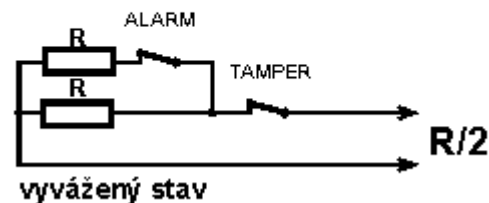
Dvojitě vyvážená smyčka

1. varianta zapojení



Dvojitě vyvážená smyčka

2. varianta zapojení



Obr. 4 Dvojitě vyvážená smyčka

Na první pohled je patrné že se musí lišit hodnota použitých odporů tak, aby byla zachována stejná hodnota na výstupu. V první variantě se hodnota odporu R rovná hodnotě, při které je vstup vyvážen – např. $1k1$ pro ústřednu JA-63K. V druhé variantě se hodnota odporu R rovná dvojnásobku hodnoty, při které je vstup vyvážen – např. $2k2$ pro ústřednu JA-63K. U ústředny systému OASIS je hodnota vyvážené smyčky $1k$.

3.16.4 Ústředna JA-82K OASIS

Ústředna JA-82K je stavebnicový systém, který má 50 adres (01 až 50) pro periferie. Základem je deska ústředny JA-82K, která má 4 drátové vstupy. Tuto desku lze doplnit o další rozšiřující moduly – JA-82R je rádiový modul, pomocí kterého lze do ústředny naučit až 50 bezdrátových periférií řady JA-8x, JA-82C je modul 10 drátových vstupů, který rozšíří kapacitu ústředny až na 14 drátových vstupů. Pro rádiovou komunikaci používá OASIS spolehlivý komunikační protokol v pásmu 868 MHz. Maximální ochrana bezdrátové komunikace je zajištěna použitím technologie plovoucího kódování a digitálním přenosem. Nastavení je možné pohodlně provádět připojeným počítačem pomocí SW OLink. Pro připojení ústředny k PC se používá interface **JA-80T**. Spojení ústředny s PC prostřednictvím Bluetooth umožní interface **JA-80BT**, nebo dálkově díky některému z komunikátorů a to např. pomocí webového rozhraní GSMLink. Bezdrátové prvky systému OASIS používají 3,6V lithiové baterie. Systém baterie průběžně kontroluje a požadavek výměny včas oznámí. Ústředna se ovládá klávesnicemi, dálkovými ovladači nebo dálkově např. mobilním telefonem prostřednictvím některého z komunikátorů. V klávesnicích systému je zabudována čtečka RFID karet, které mohou být použity samostatně nebo jejich použití lze pro zvýšení bezpečnosti podmínit zadáním kódu. Systém může být ovládán až 50 přístupovými kartami nebo kódy. Důležité informace o provozu systému se zaznamenávají do vnitřní paměti ústředny, kde je uloženo posledních 255 událostí včetně data a času. S vhodným komunikátorem umí systém hlásit

vybrané události formou SMS na mobilní telefon, zasílat veškeré podrobnosti na pult centralizované ochrany,

ale také informovat technika o potřebě servisu. Umožňuje také dálkový přístup do systému. Mobilním telefonem nebo z internetu lze na dálku systém programovat, zapínat, monitorovat či ovládat spotřebiče v domě (topení, žaluzie, světla apod.). K ústředně jsou volitelně dodávány GSM komunikátor, kombinovaný komunikátor LAN/TEL nebo hlasový telefonní komunikátor. Poslední lze kombinovat s GSM komunikátorem. Ústředna kromě přímého ovládání spotřebičů prostřednictvím výstupů poskytuje řadu dalších užitečných automatizačních funkcí. Společně s bezdrátovými termostaty optimalizuje regulaci vytápění a úspory energie. Všechny důležité události vysílá ústředna rádiem a pomocí bezdrátových modulů s relé je lze snadno, v dosahu ústředny, zpracovávat. Komunikační dosah jednotlivých bezdrátových periférií je podle konkrétních podmínek až stovky metrů. Široká nabídka bezdrátových detektorů, ovládacích prvků, sirén a automatizačních modulů dává maximální volnost pro nejvhodnější řešení individuálních požadavků zákazníka.

3.16.5. Výstupy sirén

EW – výstup externího poplachu, (**max. 0,5A**). Při poplachu tato svorka spíná na GND. Stav výstupu EW ústředna vysílá pro bezdrátové vnější sirény.

IW - výstup interního poplachu. Při poplachu tato svorka spíná na GND. Klasickou sirénu zapojte mezi svorky +U a IW (**max. 0,5A**). Stav výstupu IW ústředna vysílá pro bezdrátovou interní sirénu.

Funkce výstupu IW - interní poplach se od externího poplachu EW liší chováním v době příchodového zpoždění. Je-li během příchodového zpoždění aktivován detektor s okamžitou reakcí (např. při odjišťování vběhne dítě do obývacího pokoje) zapne systém pouze výstup IW. K aktivaci výstupu EW a reportování poplachu dojde, až když není systém včas odjištěn.

4 PROGRAMOVÁNÍ JEDNOČIPOVÝCH MIKROPOČÍTAČŮ

4.1 Základní pojmy

Mikrokontrolér, jinak též označovaný jako jednočipový mikropočítač, je integrovaný obvod, který v jediném pouzdře obsahuje všechny důležité části mikropočítače. Tento obvod lze naprogramovat tak, aby prováděl určitou úlohu v elektronickém obvodu.

Bit je základní element ve výpočetní technice. Může nabývat dvou logických stavů (1 – pravda, žárovka svítí, 0 – nepravda, žárovka nesvítí). *Byte* (čti bajt) vzniká spojením osmi bitů do jednoho celku:

b7	b6	b5	b4	b3	b2	b1	b0
----	----	----	----	----	----	----	----

Tab. 4.01: Byte

Bit s nejnižší váhou je psán vždy vpravo a bit s nejvyšší váhou je vždy nejvíce vlevo. V této kapitole bude vždy byte (spojení osmi bitů) nazýváno registrem.

Používané číselné soustavy:

Soustava desítková	Soustava dvojková	Soustava šestnáctková
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

Tab. 4.02: Číselné soustavy

4.2 Popis mikrokontroléru PIC16F84A

4.2.1 Základní vlastnosti

Mikrokontrolér PIC16F84A používá redukovanou sadu 35 instrukcí (RISC), jejichž převážná většina se vykonává za jeden instrukční cyklus programu, výjimku tvoří pouze instrukce větvení programů a skoků. Je použita Harvardská architektura, pro niž je charakteristická oddělená paměť programu a dat. PIC16F84A má paměť programu o velikosti 1024 bajtů (umožňuje 1000 přepisů), paměť RAM o velikosti 68 bajtů a paměť EEPROM pro uložení dat po odpojení napájení, jejíž velikost je 64 bajtů. Pro konfiguraci mikrokontrolér využívá 15 speciálních funkčních registrů (SFR).

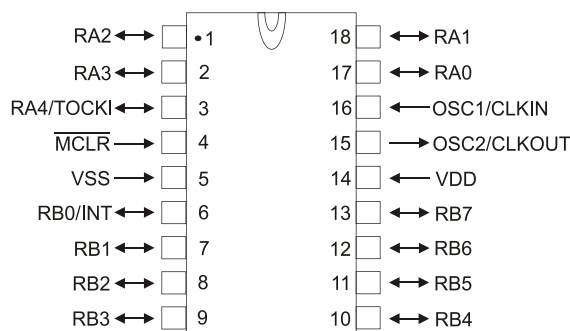
K další výbavě PIC16F84A se řadí přerušení ze 4 zdrojů (dvou vnějších zdrojů: vstup RB0/INT a změna hodnoty na čtyřech vyšších bitech PORTB <7:4> a dvou vnitřních zdrojů: přetečení od časovače TMR0 a ukončení zápisu dat do paměti EEPROM), dále můžeme využít integrovaný 8 bitový čítač/časovač s 8 bitovou předděličkou.

Obvod je vyroben technologií CMOS při rozsahu pracovního napětí 2,0 V - 5,5 V, má nízký odběr méně než 2 mA při 5 V napájení a frekvenci oscilátoru 4 MHz, 15 μ A při 2 V a frekvenci oscilátoru 32 kHz a méně než 1 μ A při stand-by režimu a napájení 2 V.

4.2.2 Popis periférií

Mikrokontrolér PIC16F84A obsahuje 13 nastavitelných vývodů, které mohou být nakonfigurovány buď jako vstupní (a poté z nich hodnota programově čtena) nebo jako výstupní (a hodnota do nich naopak programově zapisována). Z pohledu zatížení snese vývod nakonfigurovaný jako výstup pro ovládání LED max. proud 25 mA na výstup (vývod), celkově 80 mA pro 5 vývodů na Portu A a 100 mA 8 vývodů na Portu B.

4.2.3 Popis vývodů



Obr. 4.01: Pouzdro mikrokontroléru PIC16F84A

Vývod	Pin	Funkce vývodu	Provedení	Popis vývodu
RA0	17	vstup/výstup	TTL	PORT A je obousměrný vstupně/výstupní port
RA1	18	vstup/výstup	TTL	
RA2	1	vstup/výstup	TTL	
RA3	2	vstup/výstup	TTL	
RA4/TOCKI	3	vstup/výstup	ST	Čítač / časovač. V zapojení jako výstup má otevřený kolektor!
MCLR	4	reset	ST	Reset (proveden v log. nule) / vstup programovacího napětí.
Vss	5	napájení	-	Napájení GND
RB0/INT	6	vstup/výstup	ST	PORTB je obousměrný vstupně/výstupní port (I/O).
RB1	7	vstup/výstup	TTL	I/O + RB0 může být vybrán jako zdroj vnějšího přerušení
RB2	8	vstup/výstup	TTL	I/O
RB3	9	vstup/výstup	TTL	I/O
RB4	10	vstup/výstup	TTL	I/O + přerušení při změně vstupu
RB5	11	vstup/výstup	TTL	I/O + přerušení při změně vstupu
RB6	12	vstup/výstup	TTL/ST	I/O + přerušení při změně vstupu / CLK při programování souč.
RB7	13	vstup/výstup	TTL/ST	I/O + přerušení při změně vstupu / DATA při programování souč.
Vdd	14	napájení	-	Napájení +5V
OSC2/CLK out	15	výstup	-	Výstup krystalového oscilátoru. Připojení krystalu
OSC1/CLK in	16	vstup	CMOS	vstup pro krystalový oscilátor příp. RC oscilátor

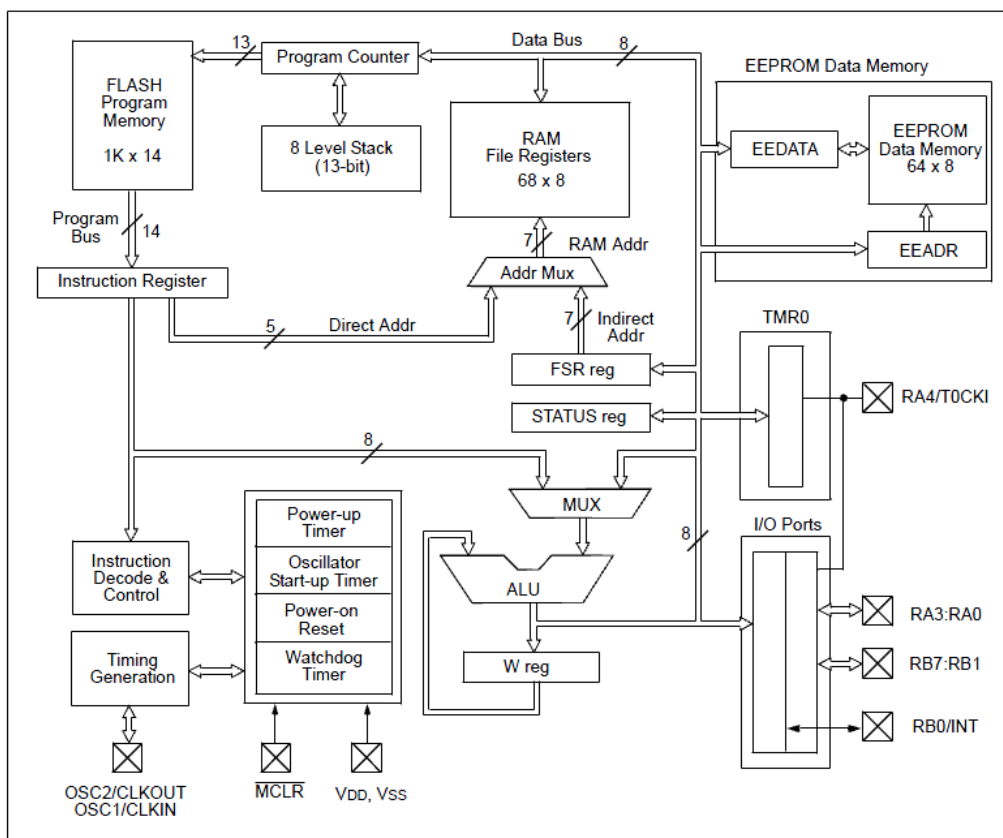
ST = Schmitt Trigger input (na vstupu Schmittův obvod)

Tab. 4.03: Přehled speciálních funkčních registrů [2]

4.2.4 Architektura

PIC16F84A využívá redukované sady instrukcí - RISC. Data a program jsou ukládány v oddělených částech paměti (tzv. Harvardská architektura). Data jsou šířky 8 bitů a kód programu má šířku 14 bitů. Tato šířka instrukčního slova (14 bitů) umožňuje vykonání většiny instrukcí během jediného instrukčního cyklu. To platí pro většinu instrukcí kromě instrukcí, které provádějí skoky či větvení programu (ty pro vykonání vyžadují 2 instrukční cykly).

ALU (aritmeticko-logická jednotka) umožňuje provádět základní aritmetické operace (sčítání, odčítání, rotace a základní logické operace). Aritmetické operace mají dva operandy, z nichž jeden je nutno vždy uložit do pracovního registru W a druhý operand je registr v paměti nebo konstanta. U jednoduchých instrukcí je operandem vlastní pracovní registr (W-registr), nebo registr v paměti. Pracovní registr W je 8 bitový a je určen pro práci ALU a zároveň je využíván značnou částí instrukcí pro odkládání dat. Pracovní registr W není možné programově adresovat (nemá adresu).



Obr. 4.02: Blokové schéma PIC16F84A [1]

V závislosti na vykonávání instrukcí ALU jsou ovlivňovány hodnoty příznakových bitů v registru STATUS [3]:

C (CARRY – přenos hodnoty do vyššího/nižšího řádu přes 8 bitů)

DC (Digit Carry – přenos hodnoty do vyššího/nižšího řádu přes 4 bity)

Z (ZERO - příznak nulového výsledku po aritmetické operaci)

Pro přímé adresování se používá pátý bit registru STATUS nazvaný RP0. Pro nepřímé adresování se využívá registru FSR (adresa 0x04 SFR), kam je zapsána adresa registru, do

kterého chci zapsat a následně je vlastní hodnota zapsána do registru nepřímá adresa (0x00 v SFR).

4.2.5 Uspořádání paměti

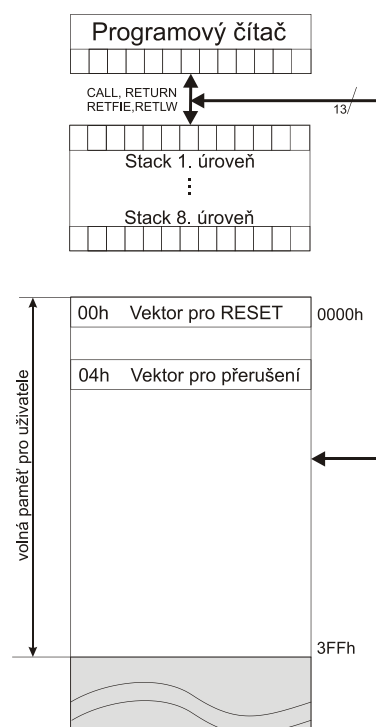
PIC16F84A využívá Harvardské architektury, to znamená, že má navzájem oddělenou paměť programu (velikost 1024 B) a paměť dat (RAM 68 B a EEPROM 64 B pro trvalé uložení dat po odpojení napájení).

4.2.5.1 Paměť programu

Procesor PIC16F84A má 13 bitový programový čítač (PC), který je schopen adresovat programovou paměť o velikosti 8K x 14. Pro PIC16F84A se používá první 1K x 14 bitů (rozsah 0000h-03FFh). Po resetu procesor začíná na adrese 0000h. Vektor přerušení se nachází na adrese 0004h. Při volání podprogramů či vykonání je využíván 8 úrovnový zásobník (Stack) typu LIFO.

4.2.5.2 Zásobník návratových adres

Procesor má 8 úrovnový zásobník s šířkou 13 bitů. Zásobník není součástí programové ani datové paměti a není možné programově provádět jeho čtení nebo do něho zapisovat. Při instrukci volání podprogramu CALL je do zásobníku uložena celá 13 bitová hodnota programového čítače (Program Counter) zvětšená o jedničku. Obsah zásobníku využívají instrukce návratu z podprogramu RETURN a RETLW nebo instrukce návratu z přerušení RETFIE.



Obr. 4.03: Organizace paměti programu

4.2.5.3 Speciální funkční registry a paměť dat

Paměť (tab. č. 4) je rozdělena do dvou stránek (tzv. banků), z nichž v každé je 128 adresovatelných buněk, ke kterým přistupujeme pomocí adresování v hexadecimální soustavě. Adresy 00h - 0Bh v banku 0 a 80h - 8Bh v banku 1 obsahují Speciální funkční registry a provádí se zde konfigurace procesoru, registry 0Ch - 4Fh jsou určeny pro data uživatele (paměť RAM).

adresa	registr	registr	adresa
00h	INDF (nepřímá adresa)	INDF (nepřímá adresa)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	--	--	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 *	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch až 4Fh	68 bajtů paměti (SRAM)	namapováno do stránky 0	8Ch až CFh
50h až 7Eh	neimplementováno (čteno jako 0)	neimplementováno (čteno jako 0)	D0h až FFh

Tab. 4.04: Organizace speciálních funkčních registrů [2]

BANK 0									
Adresa	registr	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
00h	INDF	(není fyzický registr)							
01h	TMR0	8 bitový časovač							
02h	PCL	programový čítač - spodních 8 bitů							
03h	STATUS	IRP	RP1	RP0	/TO	/PD	Z	DC	C
04h	FSR								
05h	PORTA	--	--	--	RA4/T0CKI	RA3	RA2	RA1	RA0
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
07h	--	není implementován							
08h	EEDATA	pracovní registr pro uložení dat paměti EEPROM							
09h	EEADR	pracovní registr pro uložení adresy paměti EEPROM							
0Ah	PCLATH	--	--	--					
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

BANK 1									
Adresa	registr	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
80h	INDF	(není fyzický registr)							
81h	OPTION	/RBPU	INTEDG	T0CS	TS0E	PSA	PS2	PS1	PS0
82h	PCL	programový čítač - spodních 8 bitů							
83h	STATUS	IRP	RP1	RP0	/TO	/PD	Z	DC	C
84h	FSR								
85h	TRISA	--	--	--	nastavení PORTu A				
86h	TRISB	nastavení PORTu B							
87h	--	není implementován							
88h	EECON1	--	--	--	EEIF	WRERR	WREN	WR	RD
89h	EECON2	(není fyzický registr)							
8Ah	PCLATH	--	--	--					
8Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

Tab. 4.05: Přehled speciálních funkčních registrů (SFR) [2]

4.2.6 Porty

Mikrokontrolér PIC16F84 má 2 porty, port A a port B. Celkem nabízí 13 vstupně-výstupních (I/O) vývodů. Funkce vývodů (tedy vstup či výstup) u těchto portů mohou být měněny v závislosti na požadavku programu, a to i během samotného vykonávání programu.

4.2.6.1 Port A

Port A je 5 bitový (má vývody RA0 až RA4, zbylé vývody PORTA nejsou implementovány). Vývod RA4 má na vstupu Schmittův obvod a jako výstup má otevřený kolektor. Všechny ostatní vývody portu A mají jako vstupní úroveň TTL a jako výstup budiče CMOS. Všechny I/O vývody portu A se nastavují v registru TRISA, registru na adrese 85h v banku 1 SFR, kde můžeme nastavit každý z vývodů jako vstup nebo výstup nezávisle na ostatních. Nastavením příslušného bitu na "1" (I – Input, vstup) v registru TRISA se nastaví

příslušný vývod jako vstupní, nastavením příslušného bitu na "0" (O – Output, výstup) se nastaví příslušný vývod jako výstupní. Po inicializaci procesoru jsou všechny I/O piny nastaveny jako vstupy.

Vývod portu RA4 je možné přepnout jako vstup pro hodinový signál TMR0.

4.2.6.2 Port B

Port B je 8 bitový (RB0 až RB7). Všechny I/O výstupy se nastavují v TRISB, registru na adrese 86h v banku 1 Speciálních funkčních registrů, kde můžeme nastavit každý z vývodů jako vstup nebo výstup nezávisle na ostatních. Nastavením příslušného bitu na "1" v registru TRISB se nastaví příslušný vývod jako vstupní, nastavením příslušného bitu na "0" se nastaví příslušný vývod jako výstupní. Po inicializaci procesoru jsou všechny I/O piny nastaveny jako vstupy.

Každý z vývodů portu B může mít programově připojen slabý vnitřní pull-up rezistor na všech vývodech konfigurovaných jako vstupní.

Čtyři vývody portu B (RB4 - RB7) mají při příslušném nastavení schopnost vyvolat přerušení při změně stavu. Avšak tato schopnost je dána pouze vývodům nastaveným jako vstupní. Přerušení může být vyvoláno i změnou hodnoty vstupního bitu RB0 (aktivní hrana se nastavuje v registru OPTION na adrese 81h SFR).

4.2.7 Čítač / časovač TMR0

Modul TMR0 je 8 bitový registr čítače/časovače, kde je povolen zápis i čtení obsahu registru, navíc obsahuje 8 bitovou programovatelnou předděličku a výběr aktivní hrany při externím vstupním hodinovém signálu.

V režimu časovač (TIMER) je obsah registru TMR0 inkrementován při každém instrukčním cyklu ($f_{osc}/4$). Toto platí, nebude-li použita předdělička. Je-li do registru TMR0 zapsána hodnota, je zvýšení jeho obsahu povoleno až po dvou následujících instrukčních cyklech.

V režimu čítač (Counter) bude obsah registru TMR0 zvyšován s každou vzestupnou případně sestupnou hranou na vývodu RA4/T0CKI. Přeteče-li obsah registru čítače/časovače z hodnoty FFh -> 00h, dochází k vyvolání rutiny přerušení čítače TMR0, kdy je nastaven příznakový bit TOIF.

Předděličku je volitelně možné přiřadit buď k funkčnímu modulu TMR0 nebo WDT. Obsah předděličky není možné číst ani její hodnotu měnit. Je-li předdělička předřazena čítači /

časovači TMR0 je možné programově zvolit tyto dělicí poměry 1:2, 1:4,..., 1:256. Je-li předdělička předřazena časovači Watchdog (WDT) je možné programově zvolit dělicí poměry 1:1, 1:2,..., 1:128.

4.2.8 Přerušení (interrupt)

PIC16F84A má 4 zdroje přerušení: dva vnější - vstup RB0/INT a čtyři vyšší bity PORTu B <7:4> a dva vnitřní - přetečení od časovače TMR0 a ukončení zápisu dat do paměti EEPROM. Tyto zdroje mohou vyvolat přerušení na základě vnější události.

Při přijetí přerušení (zde se předpokládá, že je povoleno) mikrokontrolér skočí na adresu 0x04 programové paměti a po vykonání zde začínajícího obslužného podprogramu se program vrátí zpět do místa, kde k přerušení došlo [4].

4.3 Základy jazyka C pro programování mikrokontrolérů

Tento manuál obsahuje pouze základy jazyka C potřebné s ohledem na programování jednočipových mikropočítačů PIC, proto celá řada standardních kapitol byla z tohoto manuálu vypuštěna. Nejedná se tedy o referenční manuál, který by pokrýval celou problematiku programování v jazyce C.

4.3.1 Jazyk C

První verzi jazyka C původně navrhl a implementoval Denis Ritchie (1972) pod operačním systémem UNIX na počítači PDP-11. Podoba jazyka byla časem vylepšována a normována, v současné době se používá na téměř všech komerčně dostupných operačních systémech. V současné době se používá ANSI-C (ANSI – American National Standard for Information System – Programming Language C) verze z roku 1999.

Jazyk ANSI-C je pro požadavky v oboru mikrokontrolérů velmi vhodný. Všeobecně je jazyk C v současné době nejčastěji používaný programovací jazyk, a to nejen při programování mikrokontrolérů, ale i pro tvorbu win aplikací [6].

Přechod z assembleru na jazyk C se zdá v prvopočátku velmi obtížný. Většinou jsou zde obavy týkající se větší velikosti kódu a také rychlosti zpracování. Velikost výsledného kódu napsaného v jazyce C je sice o něco větší, než z assembleru, ale v současné době již mikrokontroléry disponují obvykle dostatkem paměti, takže pojmu i náročné aplikace vytvořené v jazyce C [6]. Obvykle platí, že začne-li se programátor nejprve učit psát programy v assembleru a pak následně přejde na C, nechce se již k assembleru nikdy vrátit. Naopak při začátku psaní programů v jazyce C a následném přechodu k assembleru zjistíme, že zde většinou již panuje neochota se učit assembler z důvodu „zhýčkanosti“ z jazyka C (přece jen u programování v assembleru je potřeba detailnější znalost hardwaru mikrokontroléru a zvláště jeho instrukční sady).

Naopak program v jazyce C je jednodušší a velmi dobře přenositelný pro různé typy mikrokontrolérů, ať již mezi mikroprocesory od stejného výrobce nebo mikroprocesory ostatních výrobců.

4.3.2 Základy jazyka ANSI-C

Každý program v jazyce C je složen z určitých základních částí. Program v jazyce C se skládá z jedné nebo více funkcí (program v jazyce C je sám o sobě funkcí), z nichž každá obsahuje jeden nebo více příkazů. Funkce je v jazyce C pojmenovaný podprogram, který lze volat z jiných částí programu. Funkce jsou tedy základními stavebními kameny jazyka C.

Všechny příkazy jazyka C končí středníkem. Jazyk C nepovažuje konec řádku za ukončení příkazu. Na jednom řádku tedy můžeme umístit i dva nebo více příkazů nebo jeden příkaz může pokračovat přes více řádků.

4.3.3 Zápis programu v jazyce C

```
typ jméno_funkce (parametry){  
    příkazy  
}
```

Položka **typ** určuje typ dat vrácených funkcí.

Položka **jméno_funkce** určuje jméno funkce. Parametry lze funkci předávat pomocí seznamu parametrů (v závorce).

Na místě položky **příkazy** může být jeden nebo více příkazů. Teoreticky by funkce nemusela obsahovat žádný příkaz, ale protože by v takovém případě neprováděla žádnou činnost, nemělo by to v praxi žádný význam.

Příklad jednoduchého programu v jazyce C

```
#include <stdio.h>  
  
int main (void){  
    printf ("AHOJ");  
  
    return 0;  
}    // [7]
```

Poznámka: Výše uvedený program obsahují snad všechny učebnice, které se věnují základům jazyka C. Výsledkem je vypsání řetězce „AHOJ“ na obrazovku monitoru.

Vysvětlení jednotlivých řádků:

#include <stdio.h> Způsobí, že překladač jazyka C přečte hlavičkový soubor *stdio.h* a tento soubor začlení do programu. Tento soubor obsahuje mimo jiné informace o funkci *printf()*.

int main (void) { Začíná funkci *main()*. Položka *int* určuje, že *main()* vrátí celočíselnou hodnotu. Položka *void* říká překladači, že *main()* nemá žádné parametry. (*void* = prázdný)

printf ("AHOJ"); Je příkaz jazyka C. Volá standardní knihovni funkci *printf()*, která zobrazí zadaný řetězec znaků (v našem případě AHOJ).

return 0; Indikuje nulovou vracející hodnotu z funkce *main()*, což indikuje normální ukončení programu. Jakákoliv jiná hodnota představuje chybu. Položka *return* je jedním z klíčových slov jazyka C.

} Nakonec je program formálně ukončen, když dojde na uzavírací závorku funkce *main()*.

První program v jazyce C pro PIC16F84A

```

/*****
// program pro rozsvícení LED připojené anodou na RB0
*****/

#include<PIC1684.h>    //hlavičkový soubor pro 16f84A - PICC Lite

void main(void){      //zacatek hlavni funkce

// nastavení vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]

    TRISB=0b00000000;    //PORTB nastaven jako vystupni

    PORTB=0b00000001;    // rozsvícena led dioda na RB0

}   konec programu
```

Přidám opět vysvětlení jednotlivých řádků:

// vše, co je za těmito znaky do konce řádku, je považováno za komentář (nepřekládá se)

#include<PIC1684.h> vložíme hlavičkový soubor pro 16f84A (použitý překladač je PICC, při použití jiného překladače je potřeba použít odpovídající hlavičkový soubor)

void main(void) { Začíná hlavní funkci *main()*. Položka void určuje, že *main()* nevrátí žádnou hodnotu. Parametr void říká překladači, že *main()* nemá žádné parametry. V naší funkci navíc netestujeme řádné ukončení funkce *main()* (void = prázdný)

TRISB=0b11111110; Tento řádek nastavuje RB0 jako výstupní (registr TRISB obsahuje na pozici RB0 hodnotu 0, RB1-RB7 jsou ponechány jako vstupní , 1 – vstup, 0 – výstup). Tělo hlavní funkce může obsahovat řadu příkladů navzájem oddělených od sebe středníkem.

Poznámka: Hodnotu můžeme zadat i zápisem v desítkové soustavě, potom by řádek vypadal následovně: TRISB=0;

PORTB=0b00000001; Tímto řádkem pošleme na RB0, který je nastaven jako výstup, hodnotu log. 1.

} konec programu

4.3.4 Základní elementy jazyka C

V této kapitole se budeme zabývat základními elementy jazyka C, tedy základními prvky, z nichž se jazyk C skládá. Jsou to:

- klíčová slova,
- identifikátory,
- odsazovače,
- konstanty,

- proměnné,
- komentáře,
- operátory.

4.3.4.1 Klíčová slova

Klíčová slova jsou zvláštní identifikátory, které mají v jazyce C speciální význam. Proto nesmí být použity v jiném významu, než jak určuje norma ISO C, což znamená, že nesmí být použita jako jména proměnných, konstant nebo funkcí. Použití malých písmen v klíčových slovech je také důležité. Například *RETURN* nebude považováno za klíčové slovo *return*.

Poznámka: klíčová slova pro různé překladače se mohou nepatrně lišit.

Přehled klíčových slov jazyka C [8]:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

Tab. 4.06: Klíčová slova jazyka C

4.3.4.2 Identifikátory

Identifikátory jsou jména, která přiřazujeme například proměnným, funkcím a typům.

V identifikátoru je možné použít:

- písmena a..z a A..Z,
- číslice 0..9,
- znak podtržítka (_).

Prvním symbolem smí být pouze písmeno nebo podtržítko, poté následuje libovolná kombinace písmen, číslic a podtržítka. Nesmí být používána česká diakritika.

Délka identifikátoru je v normě ISO C omezená 31 znaky. Každý typ kompilátoru však upravuje toto omezení.

POZOR! Rozlišují se malá a velká písmena!

4.3.4.3 Odsazovače (bílé znaky)

Odsazovače, jinak zvané také bílé znaky. Mezi odsazovače patří mezera, tabulátor, nový řádek, posun řádku, návrat vozíku, vertikální tabulátor a nová stránka.

V obvyklém zdrojovém textu se nejčastěji můžeme setkat s prvními třemi představiteli odsazovačů (tabulátor, nový řádek, posun řádku). Odsazovače spolu s operátory a oddělovači stojí mezi identifikátory, klíčovými slovy, řetězci a konstantami použitými ve zdrojovém textu. Pro překladač představují dále nedělitelné celky.

4.3.4.4 Konstanty

Konstanty jsou čísla, znaky nebo řetězce znaků, které se nemění po celou dobu běhu programu. Konstanty definujeme po klíčovém slově `const` následovaném typem konstanty, jejím identifikátorem a po rovnítku její hodnotou ukončenou středníkem. Nejlépe si možné definice konstant vysvětlíme na příkladech:

```
const    int    konstanta = 5;
const    char    pismo='A';
```

4.3.4.5 Proměnné

Proměnné jsou paměťová místa, která jsou přístupná prostřednictvím identifikátoru. Hodnota proměnné může být během výpočtu měněna. Tím se proměnné zásadně liší od konstant, které mají po celou dobu chodu programu hodnotu stejnou - konstantní.

Během provádění programu se veškeré informace nacházejí v paměti. Tamtéž jsou umístěny pomocné hodnoty, mezivýsledky i výsledky. Rozhodně by nebylo vůbec příjemné k takovým hodnotám přistupovat prostřednictvím jejich adresy. Proto je vhodné si odpovídající paměťová

místa pojmenovat a pak se na ně odkazovat jménem. Nejprve tedy deklarujeme typy a identifikátory proměnných, čímž pro ně vyhraďujeme v paměti počítače místo a současně si překladač spojuje s identifikátorem proměnné informaci o jejím umístění v paměti (adrese) [6].

Proměnné deklarujeme uvedením datového typu, za kterým následuje identifikátor, nebo seznam identifikátorů, navzájem oddělených čárkami. Deklarace končí středníkem. Současně s deklarací proměnné můžeme, ale nemusíme, definovat i její počáteční hodnotu:

```
int a, b, c;

int a=10;
```

Přehled užívaných datových typů v jazyce C:

Datový typ	Počet bitů v paměti	Rozsah hodnot
bit	1	0 nebo 1
sbit	1	0 nebo 1
(unsigned) char	8	0 až 255
signed char	8	- 128 až 127
(signed) short (int)	16	- 128 až 127
(signed) int	16	-32768 až 32767
unsigned (int)	16	0 až 65535
(signed) long (int)	32	-2147483648 až 2147483647
unsigned long (int)	32	0 až 4294967295
float	32	$-1.5 * 10^{45}$ až $+3.4 * 10^{38}$
double	32	$-1.5 * 10^{45}$ až $+3.4 * 10^{38}$
long double	32	$-1.5 * 10^{45}$ až $+3.4 * 10^{38}$

Tab. 4.07: Přehled datových typů

Poznámka: Přehled datových typů byl převzat z nápovědy k programu MikroC. Reálné datové typy float (32 bit), double (64 bit) a long double (80 bit) mají obvykle jinou hodnotu [9].

4.3.4.5 Komentáře

K dobrým programátorským zvykům patří vkládat do programů komentáře. Komentář je poznámka, kterou přidáváme do zdrojového kódu programu. Všechny komentáře jsou

překladačem ignorovány. Usnadňují čtení programů při jejich ladění a případných pozdějších úpravách.

Komentář je část programu umístěná mezi dvojicí párových symbolů `/* */`. V tomto případě jde o komentář, který může zasahovat přes více řádků.

```
/* Komentář */  
/* Dlouhý komentář  
   zapsaný na více řádcích */
```

Některé verze jazyka C dovolují podle vzoru jazyka C++ zapisovat i komentáře ve tvaru

```
// toto je jednořádkový komentář
```

Tyto komentáře končí vždy koncem řádku (je však možno jich zapsat více za sebou, pokud na každém řádku zopakujeme úvodní `//`).

4.3.4.6 Příkaz přiřazení

Přiřazovací příkaz je nejčastěji používaným příkazem ve většině programovacích jazyků. Operátorem přiřazení v jazyce C je symbol `=`. Vlevo od `=` musí být výraz, odkazující se někam do paměti. Hodnotu zprava přece musíme někam zapsat. A kam jinam ji může program zapsat, než do paměti na adresu, kterou určí právě z levé strany přiřazovacího výrazu.

Příklady užití přiřazovacího příkazu:

```
a = 5;  
d = 'a';
```

4.3.4.7 Operátory

Aritmetické operátory

V aritmetických výrazech je nutné zdůraznit, že výraz ukončený středníkem se stává příkazem. Konstruuje se z operandů a aritmetických operátorů:

```
i=2          // výraz s přiřazením
i=2;        // příkaz
```

Aritmetické výrazy konstruuujeme z operandů a aritmetických operátorů.

Operátor	Činnost
+	sčítání
-	odčítání
*	násobení
%	zbytek po celočíselném dělení

Tab. 4.08: Aritmetické operátory

Příklad:

```
c = a + b;
```

Logické operátory

Logické hodnoty jsou dvě, pravda a nepravda. Norma ISO C říká, že hodnota nepravda je představována nulou, zatímco pravda jedničkou.

Operátor	Činnost
&	and (logický součin)
	or (logický součet)
!	not (negace)

Tab. 4.09: Logické operátory

Pravidla pro určení výsledku známe z Booleovy algebry.

Relační operátory

Relační operátory slouží pro porovnání operandů, tedy zjištění, zda mají operandy stejnou hodnotu, zda je jeden větší než druhý nebo naopak (použití např. v podmínkách).

Operátor	Činnost
<	menší
>	větší
<=	menší nebo rovno
>=	větší nebo rovno
==	rovno
!=	nerovno

Tab. 4.10: Relační operátory

Bitové operátory

Již podle názvu můžeme usuzovat, že nám bitové operátory umožňují provádět operace nad jednotlivými bity. Bitové operace je však možné provádět pouze s celočíselnými hodnotami.

Operátor	Činnost
<<	bitový posun vlevo
>>	bitový posun vpravo
&	bitový and (logický součin, konjunkce)
	bitový or (logický součet, disjunkce)
^	bitový not (negace, inverze)
~	bitový xor (nonekvivalence)

Tab. 4.11: Bitové operátory

Priorita operátorů

Priorita určuje pořadí vykonávání operací. Obsahuje-li výraz více aritmetických operátorů, budou určité operace vykonány před jinými operacemi, například násobení se vyhodnotí dříve, než třeba sčítání. Asociativita říká, vyhodnocuje-li se výraz zleva doprava, nebo naopak.

Přehled všech operátorů a jejich priorit od nejvyšší po nejnižší je uveden v tabulce 4.12.

Operátory	Asociativita
() [] ->.	zleva doprava
! ~ + - ++ -- (přetypování) *(pointer) &(adresní operátor) sizeof	zprava doleva
* / %	zleva doprava
+ -	zleva doprava
<< >>	zleva doprava
< <= > >=	zleva doprava
== !=	zleva doprava
&	zleva doprava
^	zleva doprava
	zleva doprava
&&	zleva doprava
	zleva doprava
?:	zprava doleva
= += -= *= /= %= >>= <<= &= = ^=	zprava doleva
,	zleva doprava

Tab. 4.12: Priorita operátorů

4.3.5 Řízení toku programu

4.3.5.1 Podmíněný operátor if

Příkaz *if* je používán pro větvení programu. Ternární operátor potřebuje tři operandy.

```
if(podminka) prikaz1;  
else prikaz2;
```

Podmínka je výraz, který je vyhodnocen jako pravdivý nebo nepravdivý. Je-li pravdivý, je vykonán příkaz1, je-li nepravdivý, vykoná se větev *else*, tedy příkaz2.

Příklad 1:

```
if(a<b)
    x=a;
else
    x=b;
```

Tento příklad si můžeme vysvětlit následovně: je-li proměnná *a* menší než *b*, přiřadí proměnné *x* hodnotu proměnné *a*, v opačném případě ji přiřadí hodnotu proměnné *b*.

Příklad 2:

```
if(TLACITKO==1) {    //test stisku tlacitka
    LED=1;           //rozsviceni LED (je-li stisknuto tlacitko)
}
else {
    LED=0;           //zhasnuti LED (neni-li stisknuto tlacitko)
}
```

Je-li stisknuto tlačítko (*TLACITKO*==1) LED se rozsvítí (*LED*=1), v opačném případě bude vykonána větev *else* (*LED*=0).

4.3.5.2 Přepínač switch

I když je příkaz *if* dobrý pro výběr ze dvou možností, stává se těžkopádným, když je potřeba pracovat s několika možnostmi. Jazyk C řeší tento problém příkazem *switch* (v Pascalu *case*). Příkaz *switch* je příkaz pro vícenásobný výběr. Používá se pro volbu jedné z několika variant a pracuje následovně. Hodnota je postupně testována podle seznamu celočíselných nebo znakových konstant. Když je nalezena shoda, provede se posloupnost příkazů spojená s touto hodnotou.

```
switch (celociselny_vyraz){
    case hodnota1: prikaz1;
    case hodnota2: prikaz2;
    .
    .
    .
```

```

        case hodnotaN: prikazN;

        default: prikazD;

    }

```

Příklad:

```

// ukazka pouziti prikazu switch - maticova klavesnice 3x3

switch (kod_klavesy) {

    case 1: jedna(); break; // 1

    case 2: dva(); break; // 2

    case 3: tri(); break; // 3

    case 4: ctyri(); break; // 4

    case 5: pet(); break; // 5

    case 6: sest(); break; // 6

    case 7: sedm(); break; // 7

    case 8: osm(); break; // 8

    case 9: devet(); break; // 9

    default: nula();

}

```

Je-li kód klávesy 1 (tedy *kod_klavesy* = 1), vykoná se první řádek (*case 1: jedna(); break;*), tedy bude zavolána funkce *jedna()*, v opačném případě se testují další řádky. Obsahuje-li proměnná *kod_klavesy* jinou hodnotu než je interval celých čísel 1 až 9, vykoná se řádek *default*, tedy bude zavolána funkce *nula()*. Příkaz *break* říká, že tok programu nemá pokračovat následujícím řádkem.

4.3.5.3 Cykly

Cyklus je část programu, která je v závislosti na podmínce prováděna opakovaně. U cyklu obvykle rozlišujeme řídicí podmínku cyklu. Řídicí podmínka cyklu určuje, bude-li provedeno tělo cyklu, či bude-li řízení předáno za příkaz cyklu. Tělo cyklu je příkaz, zpravidla zapsaný v podobě bloku.

Cykly můžeme rozdělit podle toho, provede-li se tělo alespoň jedenkrát, a cykly, kde tělo nemusí být provedeno vůbec. Výběr typu cyklu ponechejme na vhodnosti použití v dané situaci i na zvyklostech programátora.

Cyklus while

Podmínka cyklu *while* se testuje před průchodem cyklu, což znamená, že cyklus tedy vůbec nemusí proběhnout. Programový kód pro výraz musí být uzavřen v kulatých závorkách.

```
while (výraz)
    příkaz
```

Příklad:

```
while(1){                                //zacatek nekonecneho cyklu while (nekonecny = 1 )
    while (TLACITKO==1) {                //test tlacitka pripojeneho na RA0
        LED=1;                           //rozsviceni LED (je-li stisknuto tlacitko)
    }
    LED=0;                               //zhasnuti LED (neni-li stisknuto tlacitko)
}                                         // konec cyklu while
```

Dokud je stisknuto tlačítko (*TLACITKO==1*), svítí LED (*LED=1*) dioda připojená k výstupu anodou. Pokud tlačítko není stisknuté, LED (*LED=0*) zhasne. Celý tento úsek programu bude probíhat neustále, protože je zde použit nekonečný cyklus *while* (*while(1)*), který je z obou stran uzavřen složenými závorkami.

Cyklus do - while

Příkaz `do` je jediným z cyklů, který zajišťuje alespoň jedno provedení těla cyklu. Jinak řečeno, jeho testovací výraz je testován až po průchodu tělem cyklu. Pokud je test, představovaný hodnotou získanou z výrazu splněn, provádí se tělo cyklu, tedy příkaz, který je typicky blokem.

```
do
    příkaz
while (výraz);
```

Příklad:

```
while(1){           //zacatek nekonecneho cyklu while  (nekonecny = 1 )
    do {             //zacatek cyklu while - probehne minimalne jednou
        LED=0;        //zhasnuti LED0 (neni-li stisknuto tlacitko)
    }
    while (TLACITKO==0); //test tlacitka pripojeneho na RA0 - podminka
    LED=1;            //rozsviceni LED0 (je-li stisknuto tlacitko), ale i LED=0}
}                    // konec nekonečného cyklu while
```

Nejprve se LED zhasne ($LED=0$), až poté se testuje stav tlačítka ($TLACITKO==0$). Dokud tlačítko není stisknuté, LED nesvítí ($LED=0$). Po stisku tlačítka dojde k rozsvícení LED, problém ale je ten, že v těle nekonečného cyklu nejprve dochází v rychlém sledu za sebou ke zhasnutí LED ($LED=0$) a poté k rozsvícení LED ($LED=1$), což se projeví nižší intenzitou svitu LED diody.

Cyklus for

Cyklus *for* se používá pro zadaný počet opakování příkazu nebo bloku příkazu. Má následující syntaxi.

```
for(inicializace; test-podminky; inkrementace) příkaz;
```

Část inicializace se používá pro zadání počáteční hodnoty proměnné, která řídí průběh cyklu. Tato hodnota se obvykle označuje jako řídicí proměnná cyklu. Část inicializace se provádí pouze jednou před začátkem cyklu. Část test-podmínky testuje řídicí proměnnou cyklu na koncovou hodnotu. Je-li test podmínky vyhodnocen jako pravdivý, cyklus se opakuje. Je-li nepravdivý, cyklus se ukončí a zpracování programu pokračuje příkazem následujícím za cyklem. Test podmínky se provádí na začátku cyklu neboli před každým opakováním těla cyklu. Část inkrementace cyklu *for* se provádí na konci cyklu. To znamená, že část inkrementace se provádí poté, co se provede příkaz nebo blok, který tvoří tělo cyklu. Účelem inkrementace je zvyšovat (nebo snižovat) řídicí proměnnou cyklu o určitou hodnotu.

Část inkrementace je nepovinná. Pokud neuvedeme testovací výraz test-podmínky, použije překladač hodnotu jedna, a tedy bude provádět nekonečný cyklus.

Příklad:

```
// ukazka funkce casove smycky

void cas(void){                //funkce casu

    for( i=0; i<50000; i++) { }    //cyklus pro funkci casu

}
```

K volání funkce slouží příkaz *cas()*, následně bude vykonána časová smyčka, která má za úkol zaměstnat procesor na dobu, která je dána nastavením cyklu *for (i=0; i<50000; i++) { }*. Část inicializační obsahuje hodnotu 0 (*i=0*), k této hodnotě je každým průchodem přičítána hodnota 1 (*i++*) a vše se opakuje tak dlouho, dokud platí podmínka (*i<50000*).

4.3.6 Funkce

Funkce jsou základním stavebním kamenem jazyka C. Každý program v jazyce C minimálně jednu funkci obsahuje - *main()*. Nyní se podíváme na tvoření funkcí a na to, jak je volat.

Definice funkce je následující:

```
navratovy_typ jmeno ([parametry])

{

    telo funkce

}
```

}

Jméno funkce slouží k její identifikaci. Při volání funkce v programu musíme uvést za funkcí kulaté závorky, a to i tehdy, když funkce nemá žádné argumenty.

Volání funkce vypadá pak takto:

```
Vyraz (seznam_skutecných_parametru);
```

Výraz může být identifikátorem funkce (její jméno) nebo adresa vzniklá jeho vyhodnocením.

Parametry funkce jsou očekávaná data, která bude funkce zpracovávat. Každý parametr má své jméno a musí být určitého datového typu. Pokud funkce nemá žádné parametry, uvádí se v závorkách slůvko *void*. Pokud je jich více než jeden, oddělují se čárkou.

Funkce může mít jednu návratovou hodnotu. Její typ se uvádí před jménem funkce (návratový_typ).

Po ukončení funkce (ať již příkazem *return*, nebo tím že se vykonají všechny příkazy v jejím těle) se pokračuje v provádění kódu za místem kde byla funkce volána.

Návratová hodnota funkce *main()* se vrací operačnímu systému. Zaběhnutá praxe je, že návratová hodnota 0 (tj. *return 0*) znamená úspěšné ukončení programu, jakákoliv jiná hodnota značí chybu.

Příklad:

```

/*****

// Rozblikani LED diody pripojene na RB0 anodou

*****/

//promenne

    int i=0;

    sbit LED at RB0_bit; // LED je pripojena na pinu RB0 - prostředí MikroC

//vedlejsi funkce - casova smycka

void cas(void){                                //funkce casu

    for( i=0; i<50000; i++) { }                //cyklus pro funkci casu

}

//hlavni funkce

void main(void){                               //zacatek hlavni funkce

    TRISA=0b11111111;                          //nastaveni portu A jako vstupy ( 1=vstup , 0=vystup )

    TRISB=0b11111110;                          //nastaveni portu B jako vyvstupy ( 1=vstup , 0=vystup )

    PORTB=0b00000000;                          //zhasnuti LED na portu B

    while(1){                                  //zacatek nekonecneho cyklu while (nekonecny = 1 )

        LED=1;                                //rozsviceni LED pripojene na RB0

        cas();                                //volani funkce casu

        LED=0;                                //zhasnuti LED připojené na RB0

        cas();                                //volani funkce casu

    } //konec cyklu while

} //konec hlavni funkce

```

Poznámka: Ve vývojovém prostředí Mikroelektronika MikroC se pro časové smyčky nevyužívá cyklu for, ale vestavěných funkcí, např. *delay_ms ()* nebo *delay_us ()*.

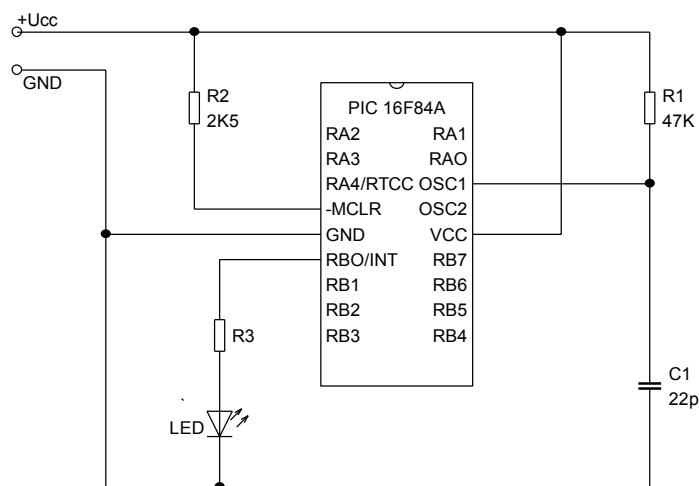
4.4 Řešené úlohy

4.4.1 Nastavení I/O bran

4.4.1.1 Rozsvícení LED diody

Zadání úlohy

Rozsviďte LED diodu připojenou na RB0. LED dioda je připojena na RB0 dle schématu (obr. 4.04).



Obr. 4.04: Schéma zapojení

Výukový účel úlohy

Naučit se pracovat se vstupy / výstupy mikrokontroléru.

Teoretický rozbor úlohy

K tomu, abychom rozsvítili LED diodu, která je připojená na pin RB0 PIC16F84A anodou (katoda je potom zapojena na svorku GND), musíme nejdřív nastavit vstupně-výstupní (I/O) vývod RB0 jako výstupní. To provedeme na Stránce 1 paměti RAM a Speciálních funkčních registrů, konkrétně v registru TRISB na adrese 86h.

TRISB 86h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
1	1	1	1	1	1	1	0

Tab. 4.13: Registr TRISB

V/V piny RB1 - RB7 ponecháme jako vstupní, z RB0 uděláme výstup.

Zapišeme-li do odpovídajícího bitu těchto registrů hodnotu 1 (I – Input, vstup), I/O pin se stane vstupem (což je také stav po zapnutí obvodu), zapišeme-li nulu (0 – Output, výstup), I/O pin se stane výstupem. Když je pin vstupní, můžeme ho číst, když je výstupní, můžeme do něj zapsat buď logickou 0 nebo 1. To uděláme zápisem do registru PORTB na adrese 06h.

PORTB 06h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
							1

Tab. 4.14: Registr PORTB

Logickou 1 zapišeme na pozici RB0 v registru PORTB na adrese 06h. To nám zajistí, že se na RB0 objeví napětí +Ucc (to odpovídá log. 1).

Program

```
//*****
// rozsviceni LED diody, která je připojena na RB0 anodou
//*****
// promenne
    sbit LED at rb0_bit;
int main(void){          //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISB=0b11111110;    //RB0 nastaven jako vystupni
    LED=0b00000001;      //LED na RB0 připojena anodou sviti
return 0;
}
```


vývody registru PORTA (RA5 až RA7 nejsou implementovány, proto hodnotu těchto bitů nemůžeme změnit).

TRISA 85h							
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RB0
1	1	1	1	1	1	1	1

Tab. 4.16: Registr TRISA

Zapíšeme-li do odpovídajícího bitu těchto registrů hodnotu 1, I/O pin se stane vstupem (což je také stav po zapnutí obvodu), zapíšeme-li nulu, I/O pin se stane výstupem. Když je pin vstupní, můžeme ho číst, když je výstupní, můžeme do něho zapsat buď logickou 0 nebo 1. To provedeme zápisem do registru PORTB na adrese 06h.

PORTB 06h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
							1

Tab. 4.17: Registr PORTB

Logickou 1 zapíšeme na pozici RB0 v registru PORTB na adrese 06h. To nám zajistí, že se na RB0 objeví napětí +Ucc (to odpovídá log. 1).

Nyní k programu v jazyce C:

Podmínka if je používána pro větvení programu. Syntaxe vypadá následovně:

```
if(podminka) prikaz1;
else prikaz2;
```

Podmínka je výraz, který je vyhodnocen jako pravdivý nebo nepravdivý. Je-li pravdivý, je vykonán příkaz1, je-li nepravdivý, vykoná se větev else, tedy příkaz2.

Program

```
//*****
// stiskem tlacitka na RA0 se rozsviti LED pripojena anodou na RB0
//*****
// promenne
    sbit TLACITKO at ra0_bit;    //TLACITKO je na RA0
    sbit LED at rb0_bit;        //LED je na RB0

int main(void){                //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111111;          //PORTA nastaven jako vstupni
    TRISB=0b11111110;          //RB0 nastaven jako vystupni
    PORTB=0b00000000;          //uvodni konfigurace registru PORTB
while(1){                      //zacatek nekonecneho cyklu while (nekonecny = 1 )
    if(TLACITKO==1) {           //test tlacitka pripojeneho na RA0
        LED=1;                  //rozsviceni LED0 (je-li stisknuto tlacitko)
    }
    else {
        LED=0;                  //zhasnuti LED0 (neni-li stisknuto tlacitko)
    }
}                               // konec cyklu while
return 0;                      // kontrola, ze fce probehla
}
```

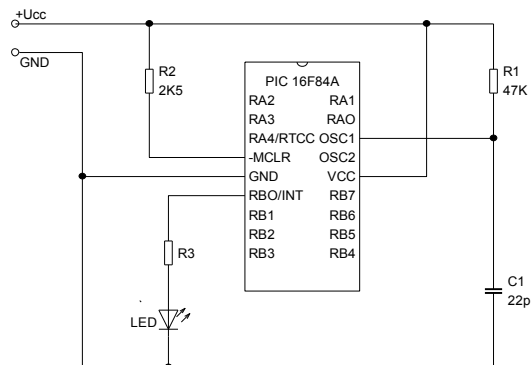
Hodnoty pro tlačítko můžeme nastavovat v okně **Watch Values** přímým zápisem do pole **Value** (hodnota), stačí tedy jen přepsat hodnotu nula za jedna pro stisknuté tlačítko, vrátit hodnotu nula pro nestisknuté tlačítko.

4.4.2 Časové smyčky

4.4.2.1 Blikání LED diody

Zadání úlohy

Rozblikujte LED diodu připojenou na RB0 s frekvencí 1 Hz. LED dioda je připojena na RB0 dle schématu (obr. 4.06).



Obr. 4.06: Schéma zapojení

Výukový účel úlohy

Naučit se pracovat s časovými smyčkami.

Frekvence použitého oscilátoru je 4 miliony kmitů za jednu sekundu. Počet instrukčních cyklů = $4\,000\,000/4 = 1\,000\,000$ instrukčních cyklů za 1 s, a odtud $T = 1/f$, tj. $1/1\,000\,000$ a to odpovídá 1 mikrosekundě.

Teoretický rozbor úlohy

Při použití RC oscilátoru dle obrázku 4.06 zjistíme, že frekvence použitého oscilátoru je 4 miliony kmitů za jednu sekundu. Počet instrukčních cyklů = $4\,000\,000/4 = 1\,000\,000$ instrukčních cyklů za 1 s, a odtud $T = 1/f$, tj. $1/1\,000\,000$ a to odpovídá 1 mikrosekundě.



Obr. 4.07: Instrukční cyklus

Ve vlastním programu můžeme pro časové smyčky použít přímo funkci *delay_ms()*, kde v závorce uvedu čas v milisekundách nebo *delay_us()*, kde do závorek udáme čas v mikrosekundách.

Celý program vypadá následovně

```

/*****
// Rozblikani osmi LED diod pripojenych na PORTB anodou
*****/

void main(){
    TRISA = 0xFF;          // nastaveni PORTA jako vstup
    TRISB = 0;             // nastaveni PORTB jako vystup
    PORTB = 0;             // uvodni inicializace PORTB
    do {                   // nekonecna smycka
        PORTB = 0b1111111; // rosvit vsechny LED diody na PORTB
        Delay_ms(500);      // cekej 0,5 sekundy
        PORTB = 0b00000000; // zhasni vsechny LED diody
        Delay_us(500000);   // cekej 0,5 sekundy
    } while(1);
} //konec hlavni funkce

```

Poznámka: funkci *delay* je možné použít pouze v prostředí MikroC. Ve všech prostředích bude určité funkční časová smyčka napsaná pomocí cyklu *for*, jak ukazuje další příklad.

```

/*****
// Rozblikani osmi LED diod pripojenych na PORTB anodou
*****/

//promenne
int i=0;

//vedlejsi funkce
void cas(void){           //funkce casu
    for( i=0; i<50000; i++) { } //cyklus pro funkci casu
}

//hlavni funkce
void main(void){          //zacatek hlavni funkce
    TRISA=0b11111111; //nastaveni portu A jako vstupy ( 1=vstup, 0=vystup)
    TRISB=0b11111110; //nastaveni portu B jako vyvstupy ( 1=vstup, 0=vystup)
    PORTB=0b00000000;    //zhasnuti LED na portu B
    while(1){             //zacatek nekonecneho cyklu while (nekonecny = 1 )
        PORTB=0;          //rozsviceni LED1
        cas();             //funkce casu
        PORTB=1;          //zhasnuti LED1
    }
}

```

```

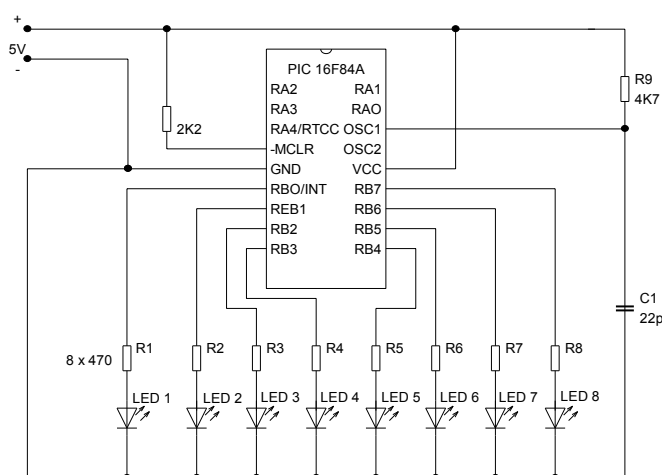
        cas();           //funkce casu
    } //konec cyklu while
} //konec programu

```

4.4.2.2 Běžící světlo

Zadání úlohy

Napište program pro zařízení zapojené dle obrázku (obr. 4.08), který imituje pohyb světelného bodu – bod se odráží mezi dvěmi krajními polohami s pevně nastavenou rychlostí 0,1s. Po dosažení krajní hodnoty RB je "pohyb" vždy vykonáván zpět.



Obr. 4.08: Běžící světlo

Výukový účel úlohy

Procvičit použití časových smyček, instrukce rotace, práce s porty.

Teoretický rozbor úlohy

Naše úloha využívá 8 LED diod připojených na PORTB mikrokontroléru. Po nadefinování používaných registrů a bitů musíme správně nastavit všechny piny PORTu B jako výstupní (jsou zde přece připojeny LED diody). Dále začneme postupným zápisem do PORTB rozsvěcovat jednotlivé LED diody.

Vzhledem k tomu, že nastavení RC oscilátoru odpovídá 4 MHz, jeden instrukční cyklus pak trvá 1 mikrosekundu, musíme vždy mezi jednotlivými volbami použít časovou smyčku – na 0,1s musíme zaměstnat mikroprocesor, aby se navenek jakoby nic nedělo.

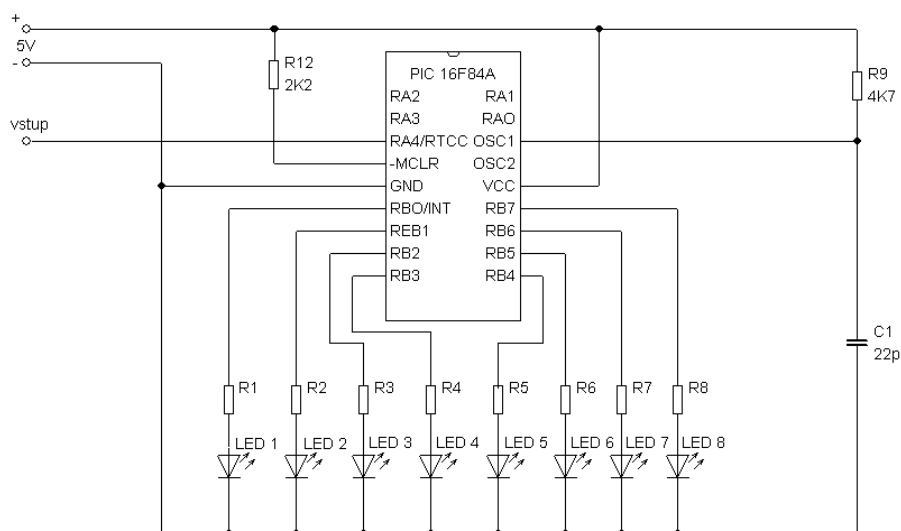
Kompletní zdrojový kód aplikace

```
/******  
// bezíci světlo s osmi LED diodami připojenými na PORTB anodou  
/******  
  
int main(void){          //zacatek hlavni funkce  
    TRISA=0b11111111; // POTRA jako vstup  
    TRISB=0b00000000; // PORTB jako vystup  
  
    while(1) {  
        PORTB = 0b00000001; // nahraj na PORTB prvni hodnotu  
        Delay_ms(100);      // cekej 100 ms  
        PORTB = 0b00000010;  
        Delay_ms(100);  
        PORTB = 0b00000100;  
        Delay_ms(100);  
        PORTB = 0b00001000;  
        Delay_ms(100);  
        PORTB = 0b00010000;  
        Delay_ms(100);  
        PORTB = 0b00100000;  
        Delay_ms(100);  
        PORTB = 0b01000000;  
        Delay_ms(100);  
        PORTB = 0b10000000;  
        Delay_ms(100);  
        PORTB = 0b01000000;  
        Delay_ms(100);  
        PORTB = 0b00100000;  
        Delay_ms(100);  
        PORTB = 0b00010000;  
        Delay_ms(100);  
        PORTB = 0b00001000;  
        Delay_ms(100);  
        PORTB = 0b00000100;  
        Delay_ms(100);  
        PORTB = 0b00000010;  
        Delay_ms(100);  
    }          // konec cyklu while  
    return 0;  
}              // konec programu
```

4.4.3 Čítač, časovač

Zadání úlohy

Mikroprocesor nyní využijeme jako čítač dle schématu (obr. 4.09), kdy na RA4 přivedete čítané pravoúhlé impulsy, na portu B je pak zapojeno 8 LED diod, které budou signalizovat stav čítače.



Obr. 4.09: Schéma zapojení čítače

Výukový účel úlohy

Naučit se pracovat s čítačem/časovačem.

Teoretický rozbor úlohy

Čítač / časovač má pro svou práci vyčleněn 8 bitový **registr TMR0** a předděličku s dělicím poměrem až 1:256. Rozdíl mezi čítačem a časovačem je ve zdroji signálu. Signál pro časovač pochází z hodinového signálu oscilátoru [16]. Frekvence načítání je tedy stejná jako frekvence vykonávání strojových cyklů tedy frekvence oscilátoru vydělená čtyřmi. Signál pro čítač pochází z externího zdroje signálu.

Režim čítače / časovače volíme nastavením nebo nulováním pátého bitu **registru OPTION**, který má název **TOCS**. Pokud je bit nulován ($TOCS = 0$), je zvolen režim časovače. Pokud je bit nastaven ($TOCS = 1$), tedy je v logické jedna, máme zvolen režim čítače.

OPTION 81h							
RPBU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

Tab. 4.18: Registr OPTION

Jak jsem již jsme zmínil, čítač reaguje na externí signál přivedený na vstup RA4. Ovšem máme zde nějaké podmínky. Na externím signálu totiž dochází ke zpoždění od aktivní hrany signálu do inkrementování obsahu registru. Doba trvání tohoto zpoždění jsou dva strojové cykly. Této doby tedy nemůžeme využít k přivedení dalšího impulzu, který by PIC byl schopen zaregistrovat. Čtvrtým bitem registru OPTION s názvem **T0SE** se určuje, zda bude čítač reagovat na sestupnou nebo vzestupnou hranu. Pokud je vynulován, probíhá inkrementace s náběžnou hranou a pokud je nastaven, probíhá s hranou sestupnou. Vstup je opatřen Schmittovým klopným obvodem.

Dalším důležitým obvodem je **předdělička**. Je-li použita předdělička můžeme na mikroprocesor přivádět i impulzy s vyšší frekvencí než bez ní. Musí ale být dodržena minimální doba trvání impulzu, která je pro L i H minimálně 10 ns. Předdělička je nastavována bity **PS0**, **PS1**, **PS2** v registru OPTION. Různými kombinacemi lze dosáhnout dělicího poměru 1:1 až 1:256. Předděličku před č/č zařazujeme vynulováním bitu **PSA** v registru OPTION. Dělicí poměry nastavuje podle tabulky na následující straně [16].

Pokud je předdělička předřazena obvodu WATCHDOG, jsou volitelné dělicí poměry v rozsahu 1:1 (nastavení PS2,PS1, PS0: 000) až 1:128 (111). Pro spolupráci s čítačem / časovačem jsou dělicí poměry dvojnásobné, tedy 1:2 (000) až 1:256 (111).

Kompletní zdrojový kód aplikace

```

/*****
// citac s PIC16F84A
*****/

void main(void){      // zacatek hlavni funkce

    trisb=0b00000000; // PORTB vystupni
    trisa=0b11111111; // PORTA vstupni
    ps0_bit=0; // nastaveni predelicky
    ps1_bit=0; // delici pomer 1:1
    ps2_bit=0;
    psa_bit=1; //volba pouziti preddelicky - citac
    t0se_bit=0; // volba aktivni hrany - inkrementace nabeznou hr.
    t0cs_bit=1; //volba pouziti preddelicky - citac
    portb=0; // vynulovani PORTB

    while (1) {

        portb=tmr0; // kopirovani obsahu TMR0 na PORTB

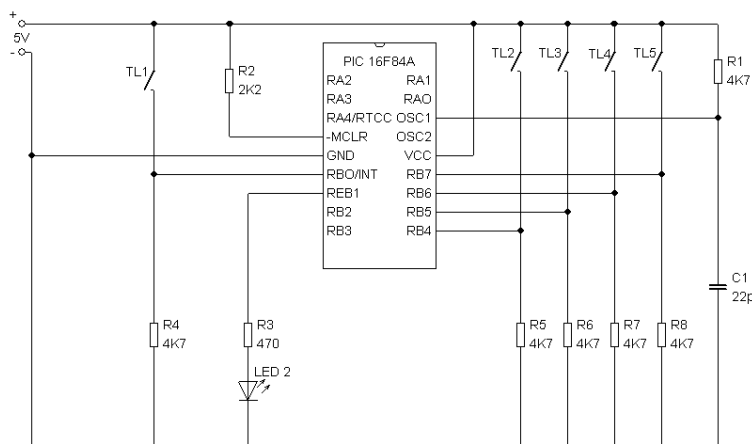
    }      // konec cyklu while
}      // konec programu

```

4.4.4 Přerušení

Zadání úlohy

Mikroprocesor je zapojen dle schématu (obr. 4.10), kdy na RB0 a RB4 až RB7 jsou připojená tlačítka, na RB1 je pak zapojena LED dioda. Rozsviťte LED diodu pomocí vnějšího přerušení na tlačítku připojeném na RB0, v druhé části využijte jedno z tlačítek RB4 až RB7, kdy s pomocí vnějšího přerušení na čtyřech vyšších bitech registru PORTB LED diodu **zhasnete**.



Obr. 4.10: Schéma zapojení obvodu

Výukový účel úlohy

Naučit se pracovat s vnějším přerušením na vývodu RB0/INT.

Teoretický rozbor úlohy

PIC16F84A má čtyři zdroje přerušení, a to dva vnitřní (přerušení přetečením čítače TMR0 a od ukončení zápisu do EEPROM), a dva vnější zdroje (přerušení na vývodu RB0/INT a přerušení změnou na čtyřech vyšších bitech PORTB, tedy pinů RB4 – RB7). O přerušení se stará registr INTCON, který je opět osmibitový a nachází se jak na stránce 0 (0x0B), tak na stránce 1 (0x8B) paměti RAM.

INTCON 0Bh; 8Bh

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
7	6	5	4	3	2	1	0

Tab. 4.19: Registr OPTION

Význam bitů registru INTCON:

GIE – Globální povolení přerušení (1 – všechna přerušení povolena / 0 - zakázána)
EEIE – Přerušení od dokončení zápisu (1 – povoleno / 0 - maskováno)
TOIE – Přerušení od přetečení čítače/časovače (1 – povoleno / 0 - maskováno)
INTE – Přerušení od vnějšího signálu na RB0 (1 – povoleno / 0 - maskováno)
RBIE – Přerušení změnou RB4 – RB7 (1 – povoleno / 0 -maskováno)
TOIF – Příznak přetečení čítače/časovače (1 – došlo / 0- nedošlo)
INTF – Příznak vnějšího přerušení na RB0 (1 – došlo / 0- nedošlo)
RBIF – Příznak přerušení změnou RB4 – RB7 (1 – došlo / 0- nedošlo)

Příznak, který jen nastaven při přerušení od dokončení zápisu do EERPOM, je v registru EECON1 a jmenuje se EEIF.

Zatím jsem se zde zmiňoval jen o jakémsi registru INTCON a čtyřech zdrojích přerušení. K čemu ale je přerušení dobré? Takže mějme nějaký program, který se nerušeně v pohodě a v poklidu vykonává [16], např. točící se servomotor, kdy v průběhu vykonávání programu kdosi stiskne nějakou klávesu, např. maticové klávesnice. Najednou ale vznikne potřeba tento běh programu nějak upravit nebo přerušit. Právě k tomu nám může soužit ono přerušení.

Máme tedy nějaký hlavní běh programu a najednou se objeví požadavek na přerušení. Vykoná se poslední rozpracovaná instrukce, opustí se vykonávání hlavního programu a započne zpracování přerušení. Tomuto zpracování říkáme **obsluha přerušení**. A po obslužení přerušení se program vrátí tam, kde skončil a “jakoby nic“ pokračuje dále tam, kde před přerušením skončil (tedy pokud není dáno jinak).

Poté, co je vyvoláno nějakým způsobem přerušení, dojde k zakázání ostatních přerušení a to tak, že se vynuluje registr GIE (jsou zakázána všechna další přerušení). Uloží se adresa místa, na kterém bylo vykonáno přerušení (ta se ukládá proto, aby zde program mohl znovu pokračovat). A vykonávání programu se přesune na adresu 0004h. Ta je pro přerušení pevně vyhrazena. Nyní v případě, že je povoleno několik zdrojů přerušení, testujeme bity RBIF, INTF, EEIF a T0IF a zjišťujeme tak, ze kterého zdroje ono přerušení pochází a podle toho se vykoná programem (programátorem) určená činnost. Na závěr softwarově vymažeme příznaky přerušení, aby po dokončení obsluhy nenastalo přerušení nové.

Navíc je nutné se zmínit o jednom z problémů, který musíme při obsluze přerušení řešit, a tím je obsah registrů, především se jedná o pracovní registr W a registr STATUS. Ty je nutné před započítáním obsluhy přerušení někde zazálohovat a před jejím ukončením opět obnovit.

V jazyce C je vypořádání přerušení poměrně jednoduché, přerušení a s tím související činnosti popsané výše vypořádává za vývojáře funkce `interrupt()`, jak je vidět z následujících programů.

Program verze 1

Program na demonstraci přerušení vykonává neustále nekonečný cyklus (řádek `while()`). Po stisku tlačítka na RB0 (náběžnou hranou – šestý bit s názvem `INTEDG` registru `OPTION`) dojde k rozsvícení LED pomocí přerušení (tedy automatickému vykonání obsluhy přerušení):

```
//*****
// ukazka vnejsiho preruseni - stiskem tlacitka na RB0 se rozsviti LED
// pripojena anodou na RB0
//*****
// promenne
sbit LED at rb1_bit;    //LED je na RB1
sbit TLACITKO at rb0_bit;    //TLACITKO je na RB0

void interrupt() {
    if(INTCON.INTF) {        //test tlacitka pripojeneho na RA0
        LED=1;              //rozsviceni LED0 (bylo-li stisknuto tlacitko)
        INTF_bit = 0;
    }
}

int main(void){            //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111111;      // PORTA nastaven jako vstupni
    TRISB=0b11111001;      // RB0 nastaven jako vystupni
    PORTB=0b00000000;      // nastaveni pocatecni hodnoty na PORTB
    INTE_bit = 1;          // povoleni preruseni od RB0
    GIE_bit = 1;           // globalni povoleni preruseni
    INTEDG_bit = 0;        // preruseni nastane nabeznou hranou

    while(1);              // nekonečný cyklus

    return 0;
}                          // konec programu
```

Program verze 2

Program na demonstraci přerušení v první části vykonává tu samou činnost, co program verze 1, navíc však povoluje přerušení od čtyř vyšších bitů registru `PORTB`. Po stisku tlačítka

na RB0 dojde k rozsvícení LED pomocí přerušení na vývodu RB0/INT, po stisku jakéhokoliv z tlačítek RB4 až RB7 dojde k zhasnutí LED diody opět pomocí vnějšího přerušení změnou stavu čtyř vyšších bitů registru PORTB:

```
//*****
// ukazka vnejsiho preruseni - stiskem tlacitka na RB0 se rozsviti LED
// pripojena anodou na RB0, stiskem RB4-RB7 se LED dioda zhasne
//*****

// promenne
sbit LED at rb1_bit;    //LED je na RB1
sbit TLACITKO at rb0_bit;    //TLACITKO je na RB0

void interrupt() {
    if(INTCON.INTF) {        //test tlacitka pripojeneho na RA0
        LED=1;              //rozsviceni LED (bylo-li stisknuto tlacitko)
        INTF_bit = 0;
    }
    if(INTCON.RBIF) {        //test tlacitka pripojeneho na RA0
        LED=0;              //zhasnuti LED (bylo-li stisknuto tlacitko)
        RBIF_bit = 0;
    }
}

int main(void){            //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111111;      // PORTA nastaven jako vstupni
    TRISB=0b11111001;      // RB0 nastaven jako vystupni
    PORTB=0b00000000;      // nastaveni pocatecni hodnoty na PORTB
    INTE_bit = 1;          // povoleni preruseni od RB0
    RBIE_bit = 1;          // povoleni preruseni od RB4 az RB7
    GIE_bit = 1;           // globalni povoleni preruseni
    INTEDG_bit = 0;        // preruseni nastane nabeznou hranou

    while(1);              // nekonečný cyklus

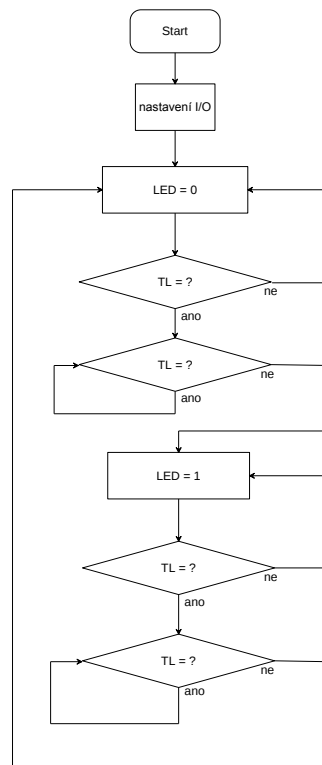
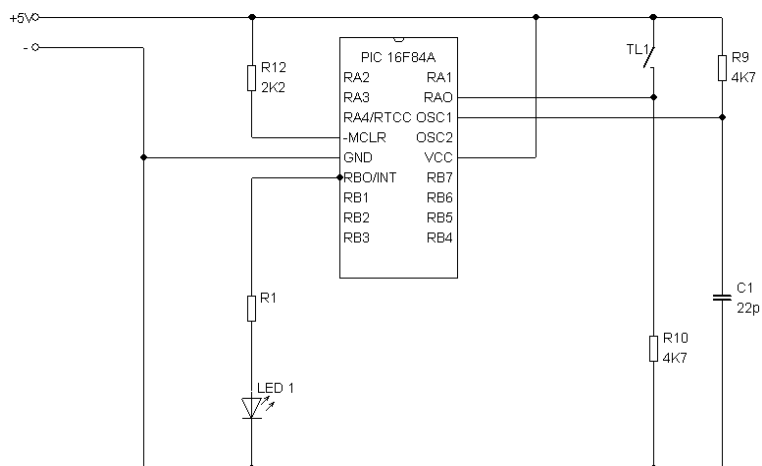
    return 0;
}                          // konec programu
```

4.4.5 Tlačítka

4.4.5.1 Multifunkční tlačítko

Zadání úlohy

Napište aplikaci, která po stisku tlačítka připojeného na RA0 rozsvítí LED diodu připojenou na RB0. Dioda zůstane svítit. LED diodu zhasne až další stisk tlačítka. A opět zůstane zhasnutá. Vše se bude neustále opakovat.



Obr. 4.11: Schéma zapojení obvodu a vývojový diagram

Výukový účel úlohy

Procvičení algoritmů s tlačítky, procvičení práce s vývojovými diagramy.

Teoretický rozbor úlohy

Na obr. 4.11 je vývojový diagram aplikace. Není možné testovat stisknutí tlačítka (vzhledem k velkému počtu instrukcí, které proběhnou za 1 s, při 4 MHz taktovacím kmitočtu trvá 1 instrukční cyklus 1 mikrosekundu), ale je docela dobře možné vyhodnotit, kdy tlačítko pustíme.

Zdrojový kód aplikace

```

//*****
// multifunkční tlačítko rozsvítí LED připojenou anodou na RB0 po stisku jednoho z tlačítek
// připoj. na RA0 ci na RA1
//*****
// promenne
int main(void){          //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111111;    //PORTA nastaven jako vstupni
    TRISB=0b00000000;    //RB0 nastaven jako vystupni
    PORTB=0b00000000;

    while(1){
        if(RA0_bit==1)    //test tlacitka pripojeneho na RA0
        {
            //provede se, je-li stisknuto tlacitko
            Delay_ms(100);

            if (RA0_bit==0) {
                PORTB = ~PORTB; // negace hodnot PORTu B
            }
        }

        // konec cyklu while
    }

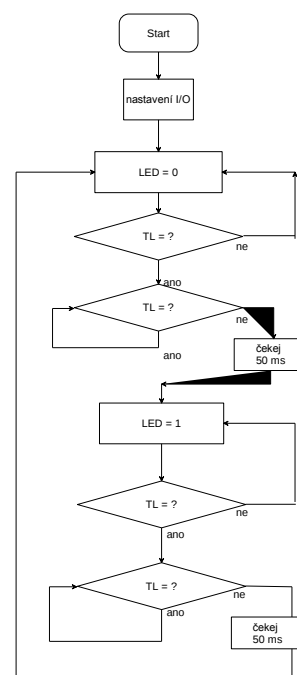
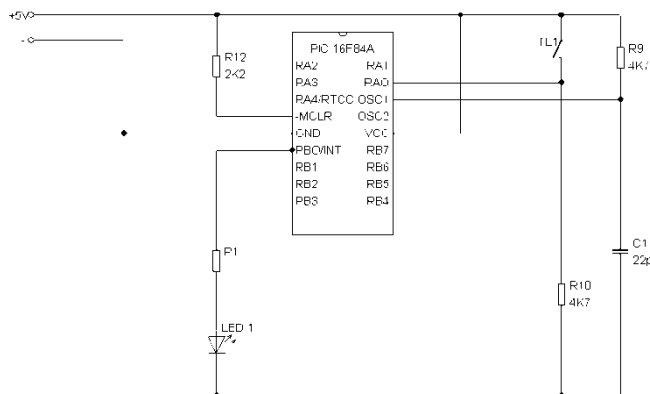
    return 0;
}

```

4.4.5.2 Multifunkční tlačítko s ošetřením zákmitů

Zadání úlohy

Napište aplikaci, která po stisku tlačítka připojeného na RA0 rozsvítí LED diodu připojenou na RB0. Dioda zůstane svítit. LED diodu zhasne až další stisk tlačítka. A opět zůstane zhasnutá. Vše se bude neustále opakovat. Vypořádejte ošetření zákmitů.



Obr. 4.12: Schéma zapojení obvodu a vývojový diagram

Výukový účel úlohy

Naučit se pracovat s ošetřením zákmitů tlačítka.

Teoretický rozbor úlohy

Nyní se dostáváme k řešení zákmitů kontaktů. Ve všech předchozích případech nám zákmity kontaktů řešila hardwarová konstrukce tlačítka. Máme-li tlačítko, na kterém jsou zákmity, je vhodné softwarové řešení, kdy jde o to, že v době, kdy zákmity probíhají, se vykonává program a mikroprocesor nesmí v této době provádět test zakmitávajícího tlačítka. To se dá vyřešit např. vhodně umístěnou časovou smyčkou. Na obr. 4.12 je vývojový diagram, který řeší danou aplikaci.

Pro ošetření zákmitů můžeme použít speciální rutinu vývojového prostředí MikroC:

```
unsigned short Button(unsigned short *port, unsigned short pin_mikroprocesoru,  
unsigned short cas_zpozdeni, unsigned short aktivni_hrana);
```

port – specifikuje port, na kterém je tlačítko připojeno, např. &PORTA

pin_mikroprocesoru – specifikuje konkrétní pin portu, na kterém je tlačítko připojeno (rozsah 0-7)

cas_zpozdeni – doba potlačení odskoku tlačítka v milisekundách, např. 1

aktivni_hrana – určuje, kdy je rutina aktivní (0 nebo 1)

Zdrojový kód aplikace

```
//*****  
// stiskem tlacitka na RA0 se rozsviti LED pripojena anodou na RB0  
//*****  
// promenne  
    sbit TLACITKO at ra0_bit;    //TLACITKO je na RA0  
    sbit LED at rb0_bit;        //LED je na RB0  
    int  oldstate;  
  
int main(void){    //zacatek hlavni funkce  
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]  
    TRISA=0b11111111;    //PORTA nastaven jako vstupni  
    TRISB=0b11111110;    //RB0 nastaven jako vystupni  
    PORTB=0b00000000;  
do {  
    if (Button(&PORTA, 0, 1, 1)) oldstate = 1;
```

```

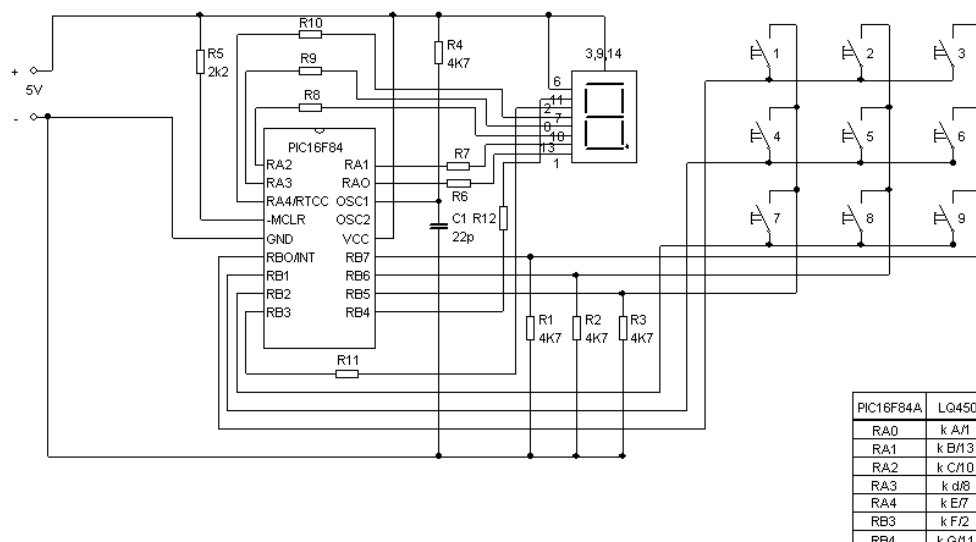
if (oldstate && Button(&PORTA, 0, 1, 0)) {
    PORTB = ~PORTB;
    oldstate = 0;
}
} while(1);
return 0;
}

```

4.4.5.3 Maticová klávesnice

Zadání úlohy

Napište program, který v zajistí, že se na LED sedmisegmentovém zobrazovači budou zobrazeny čísla právě stisknuté klávesy. Při stisku klávesy č.1 bude na displeji svítit hodnota číslo 1, při stisku klávesy č. 2 dvojka atd. Schéma zapojení je na obrázku 4.13.



Obr. 4.13: Schéma zapojení obvodu

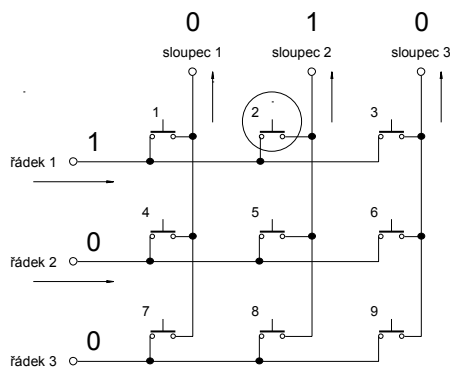
Výukový účel úlohy

Naučit se pracovat s maticovou klávesnicí a procvičení práce s podmíněným operátorem if.

Teoretický rozbor úlohy

Naše úloha využívá dvě vstupně/výstupní zařízení: jsou to maticová klávesnice a LED sedmisegmentový zobrazovač. Po nadefinování používaných bitů musíme správně nastavit připojené vstupně/výstupní piny: PORTA nastavíme jako výstupní (je zde připojen

sedmisegmentový zobrazovač, výstupy budou i na RB3 a RB4 – zde jsou katody f a g), RB0-RB2 nastavíme jako výstupy a RB5-RB7 jako vstupy (obr. 4.13). Z pinů RB0-RB2 budeme postupně vysílat vždy na jeden ze tří pinů log.1 a na RB5-RB7 budeme testovat, zda se nám log.1 neobjevila.



Obr. 4.14: Maticová klávesnice

V našem případě máme např. stisknuté tlačítko 2 (obr. 4.14), proto, když pošleme log. 1 do prvního řádku, přes stisknuté tlačítko 2 projde log.1 na sloupec 2, kde ji můžeme detekovat.

Ukázka části zdrojového kódu pro test prvního řádku

```
radek_1=1; // test prvního sloupce, tj. 1-3
radek_2=0;
radek_3=0;
if(sloupec_1==1) { //test sloupce_1
    jedna(); //rozsviceni cisla 1
}

if(sloupec_2==1) { //test sloupce_2
    dva(); //rozsviceni cisla 2
}

if(sloupec_3==1) { //test sloupce_3
    tri(); //rozsviceni cisla 3
}
```

Takto jsme schopni vyhodnotit, že byla stisknuta určitá klávesa, např. klávesa č.2. Pokud byla např. stisknuta klávesa s hodnotou 2, tak zavoláme funkci dva(), která zajistí nahrání požadovaných hodnot na sedmisegmentový zobrazovač. Nemí-li stisknuta žádná klávesa, testování bude probíhat neustále dokola.

Zdrojový kód celé aplikace

```
//*****
// maticova klavesnice + sedmisegmentovka
//*****

// promenne
    sbit radek_1 at rb0_bit;    //Radek 1 maticove klavesnice
    sbit radek_2 at rb1_bit;    //Radek 2 maticove klavesnice
    sbit radek_3 at rb2_bit;    //Radek 3 maticove klavesnice
    sbit sloupec_1 at rb5_bit;   //Sloupec 1 maticove klavesnice
    sbit sloupec_2 at rb6_bit;   //Sloupec 2 maticove klavesnice
    sbit sloupec_3 at rb7_bit;   //Sloupec 3 maticove klavesnice
    sbit katoda_F at rb3_bit;    //katoda F sedmisegmentovky
    sbit katoda_G at rb4_bit;    //katoda G sedmisegmentovky

//podprogramy pro rozsviceni cisel na sedmisegmentovce

void jedna(void){                //rozsviti cislo 1 na sedmisegmentovce
    PORTA=0b11111001;
    katoda_F=1;
    katoda_G=1;
}

void dva(void){                  //rozsviti cislo 2 na sedmisegmentovce
    PORTA=0b00000100;
    katoda_F=1;
    katoda_G=0;
}

void tri(void){                  //rozsviti cislo 3 na sedmisegmentovce
    PORTA=0b01010000;
    katoda_F=1;
    katoda_G=0;
}

void ctyri(void){                //rozsviti cislo 4 na sedmisegmentovce
    PORTA=0b00011001;
    katoda_F=0;
    katoda_G=0;
}

void pet(void){                  //rozsviti cislo 5 na sedmisegmentovce
    PORTA=0b00010010;
    katoda_F=0;
    katoda_G=0;
}

void sest(void){                 //rozsviti cislo 6 na sedmisegmentovce
    PORTA=0b00000010;
    katoda_F=0;
```

```

    katoda_G=0;
}

void sedm(void){           //rozsviti cislo 7 na sedmisegmentovce
    PORTA=0b00011000;
    katoda_F=1;
    katoda_G=1;
}

void osm(void){           //rozsviti cislo 8 na sedmisegmentovce
    PORTA=0b00000000;
    katoda_F=0;
    katoda_G=0;
}

void devet(void){        //rozsviti cislo 9 na sedmisegmentovce
    PORTA=0b00010000;
    katoda_F=0;
    katoda_G=0;
}

int main(void){          //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11100000;    //PORTA nastaven jako vstupni
    TRISB=0b11100000;    //RB0 nastaven jako vystupni

    PORTA=0b11111111;
    PORTB=0b00011000;
    katoda_F=1;
    katoda_G=0;

while(1){                //zacatek nekonecneho cyklu while  (nekonecny = 1 )
    radek_1=1; // test prvniho sloupce, tj. 1-3
    radek_2=0;
    radek_3=0;
    if(sloupec_1==1) {    //test sloupce_1
        jedna();          //rozsviceni cisla 1
    }

    if(sloupec_2==1) {    //test sloupce_2
        dva();             //rozsviceni cisla 2
    }

    if(sloupec_3==1) {    //test sloupce_3
        tri();             //rozsviceni cisla 3
    }

    radek_1=0; // test druhého sloupce, tj. 4-6
    radek_2=1;

```

```

radek_3=0;

if(sloupec_1==1) {      //test sloupce_1
    ctyri();           //rozsviceni cisla 4
}

if(sloupec_2==1) {      //test sloupce_2
    pet();             //rozsviceni cisla 5
}

if(sloupec_3==1) {      //test sloupce_3
    sest();            //rozsviceni cisla 6
}

radek_1=0; // test tretho sloupce, tj. 7-9
radek_2=0;
radek_3=1;

if(sloupec_1==1) {      //test sloupce_1
    sedm();            //rozsviceni cisla 7
}

if(sloupec_2==1) {      //test sloupce_2
    osm();             //rozsviceni cisla 8
}

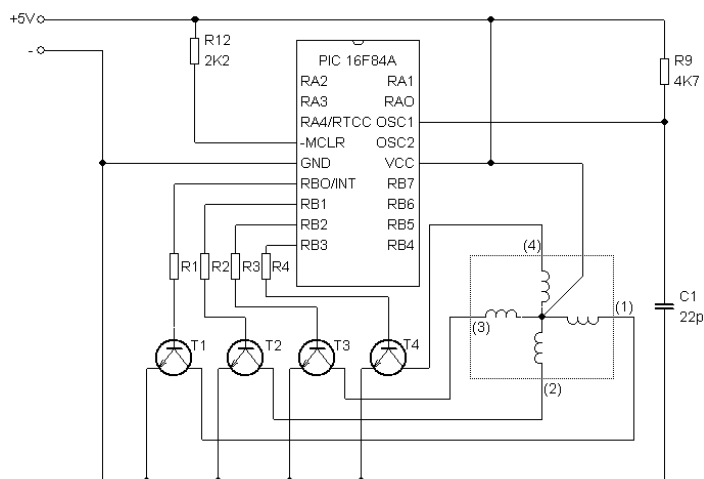
if(sloupec_3==1) {      //test sloupce_3
    devet();           //rozsviceni cisla 9
}
}
}

```

4.4.6 Krokový motor

Zadání úlohy

Ovládejte krokový motor zapojený na RB0 – RB3 mikroprocesoru dle schématu (obr. 4.15).



Obr. 4.15: Schéma zapojení obvodu

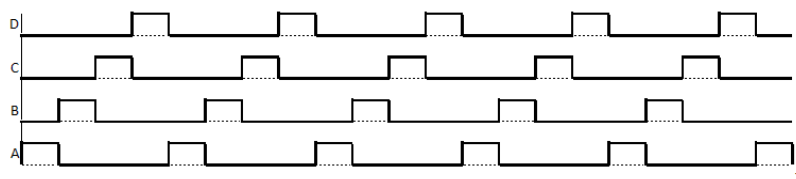
Výukový účel úlohy

Naučit se pracovat s krokovým motorem.

Teoretický rozbor úlohy

Základní princip krokového motoru je jednoduchý. Proud procházející cívkou statoru vytvoří magnetické pole, které přitáhne opačný pól magnetu rotoru. Vhodným zapojováním cívek dosáhneme vytvoření rotujícího magnetického pole, které otáčí rotorem. V našem zapojení je krokový motor připojen na vývody RB0 až RB3 (vzhledem k velkým proudům odebírajících cívek jsou použity tranzistory), proto tyto I/O piny nastavíme jako výstupní. Dále budeme postupně spínat cívky 1 až 4 pomocí unipolárního jednofázového řízení s plným krokem (obr. 4.16), nebo podle unipolárního řízení s polovičním krokem (obr. 4.17).

Unipolární řízení krokového motoru s plným krokem

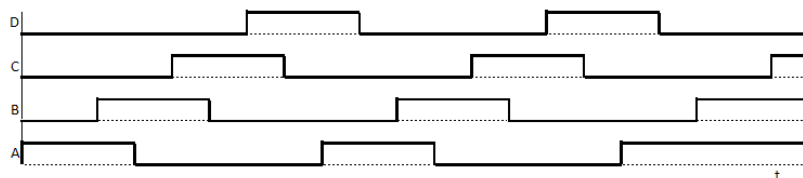


Obr. 4.16: Unipolární jednofázové řízení s plným krokem

Ukázka části programu:

```
while (1){  
    portB=0b00000001;  
    Delay_ms(50);  
    portB=0b00000010;  
    Delay_ms(50);  
    portB=0b00000100;  
    Delay_ms(50);  
    portB=0b00001000;  
    Delay_ms(50);  
}
```

Unipolární řízení krokového motoru s polovičním krokem



Obr. 4.17: Unipolární jednofázové řízení s polovičním krokem

Ukázka části programu:

```
while (1){  
    portB=0b00000001;  
    Delay_ms(50);  
    portB=0b00000011;  
    Delay_ms(50);  
    portB=0b00000010;  
    Delay_ms(50);  
    portB=0b00000110;  
    Delay_ms(50);  
    portB=0b00000100;  
    Delay_ms(50);  
}
```

```

    Delay_ms(50);
    portB=0b00001100;
    Delay_ms(50);
    portB=0b00001000;
    Delay_ms(50);
    portB=0b00001001;
    Delay_ms(50);
}

```

Řízení s plným krokem znamená, že na jednu otáčku je potřeba přesně tolik kroků, kolik zubů má stator daného motoru.

Řízením s polovičním krokem dosáhneme dvojnásobné přesnosti. Technicky se jedná o střídání kroků s jedno a dvoufázovým řízením.

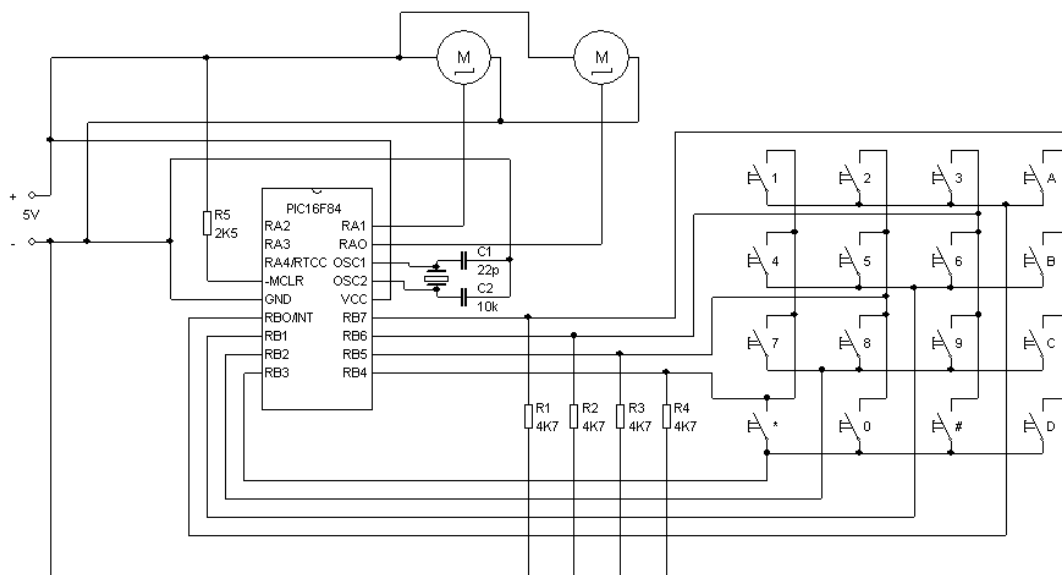
Směr otáčení můžeme změnit otočením sledu spínaných fází.

Rychlost otáčení je dána délkou použité časové smyčky, čím kratší je časová smyčka, tím rychlejší je otáčení krokového motoru. Zkrátíme-li časovou smyčku příliš (méně než cca 20 ms), krokový motor začne ztrácet kroky.

4.4.7 Servomotor

Zadání úlohy

Ovládejte servomotor zapojený na RA0 mikroprocesoru dle schematu (obr. 4.18).



Obr. 4.18: Schéma zapojení obvodu

Výukový účel úlohy

Naučit se pracovat s modelářským servem.

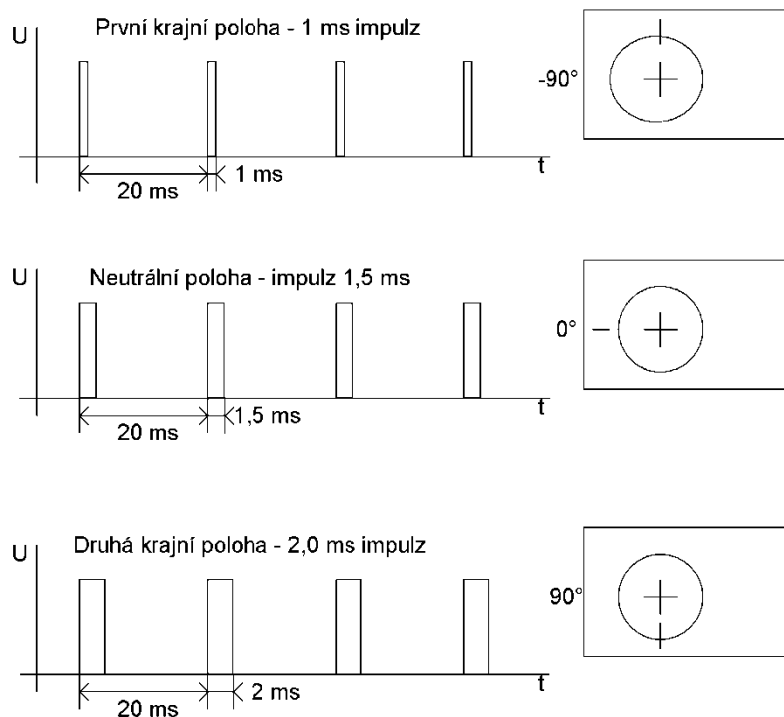
Teoretický rozbor úlohy

Modelářské servo se používá jako levné a dostupné polohové jednotky pro natočení např. u robotů. Servo obsahuje stejnosměrný motorek, převodovku a s výstupní hřídelí spřažený miniaturní potenciometr, připojený přímo na výstupní hřídel. U serv vyšších kategorií je připojen přes zvláštní převod (tzv. nepřímý náhon), který velmi účinně chrání před přenosem vibrací.

Žádaná hodnota natočení výstupní hřídele je ve formě šířkově modulovaného signálu (s úrovní TTL) s periodou 20 ms a šířkou pulzu mezi cca 1 a 2 ms. (obr. 4.19).

Do střední polohy natočíme servo tedy tak, že na příslušný výstup mikroprocesoru budeme posílat po určitou minimální dobu impuls 1,5 ms v rámci 20 ms dlouhém (tj. 1,5 ms log. 1 a 18,5 ms log. 0). Nastačí jen jednou, ale je třeba určitého počtu těchto impulsů, aby se servo

mělo čas do požadované polohy dotočit. Je třeba si uvědomit, že servo potřebuje cca 0,8 s na to, aby se dostalo z jedné do druhé krajní polohy. To odpovídá 40 rámcům po 20 ms.



Obr. 4.19: Ukázka ovládání servomotorů

První program provádí nastavení servomotoru do střední polohy. V těle cyklu jen musíme zajistit 1,5 ms log. 1 a po zbytek do 20 ms – tedy na 18,5 ms (20 ms - 1,5 ms = 18,5 ms) log. 0.

Program č.1

```
//*****
// natoceni servomotoru pripojeneho na RA0 do stredni polohy
//*****
void main() {          //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111100;    //RA0 a RA1 jako vystupy
    TRISB=0b11110000;    //RB0-RB3 jako vystupy, RB4-RB7 jako vstupy
    PORTB=0b00000000;

    while(1){
        RA0_bit = 1;
        Delay_us(1500);
        RA0_bit = 0;
        Delay_us(18500);
    }                    // konec cyklu while
}
```

Úplně analogicky můžeme servo natočit do jakékoliv polohy o 90 stupňů na jednu či druhou stranu od střední polohy (jak ukazuje obr. 4.19), nebo jakékoliv polohy mezi dvěma krajními polohami.

Druhý program provádí pohyb servomotoru mezi dvěma krajními polohami. Počet opakování nám zajišťuje cyklus for. Výsledkem cyklu bude 0,8 s natáčení z jedné krajní polohy do druhé, jak zjistíme z jednoduchého výpočtu 20 ms dlouhý rámec x 40 opakování.

Program č.2

```

/*****
// servomotor - pohyb mezi krajními polohami
*****/

// promenne
int i = 1;
int n;

void main() {          //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111100;    //RA0 a RA1 jako vystupy
    TRISB=0b11110000;    //RB0-RB3 jako vystupy, RB4-RB7 jako vstupy
    PORTB=0b00000000;

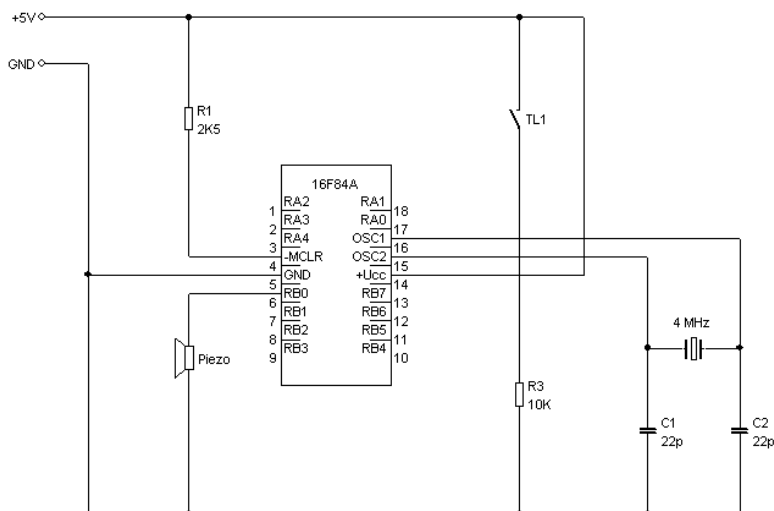
    while(1){
        for ( i = 1; i <= 40 i++ ) {
            RA0_bit = 1;
            RA1_bit = 1;
            Delay_us(1000);
            RA0_bit = 0;
            RA1_bit = 0;
            Delay_us(19000);
        }
        for ( i = 1; i <= 40; i++ ) {
            RA0_bit = 1;
            RA1_bit = 1;
            Delay_us(2000);
            RA0_bit = 0;
            RA1_bit = 0;
            Delay_us(18000);
        }
    }
    // konec cyklu while
}

```

4.4.8 Piezoměnič

Zadání úlohy

Na připojeném piezo reproduktoru na RB0 dle schematu (obr. 4.20) přehrajte v Morseově abecedě zašifrované volání o pomoc – SOS. Tečka bude reprezentována krátkým pípnutím 0,2 s a čárka delším pípnutím 0,6 s.



Obr. 4.20: Schéma zapojení obvodu

Výukový účel úlohy

Naučit se pracovat s piezoměničem.

Teoretický rozbor úlohy

V nastavení I/O nastavíme RB0 jako výstup, protože je zde připojen piezoměnič. Je potřeba si uvědomit, že pokud má piezo pískat, je potřeba vygenerovat na výstup RB0 signál o určité frekvenci (dle použitého piezo reproduktoru), v našem případě 2,8 kHz.

Pro vlastní úlohu je výhodné využít podprogramů (z důvodu přehlednosti programu).

Přehrávaný kód v Morseově abecedě vypadá následovně:

... _ _ _ .
S O S

Přehled znaků Morseovy abecedy

A . _
B _ . . .
C _ . _ .
D _ . .
E .
F . . _ .
G _ _ .
H
CH _ _ _ _
I . .
J . _ _ _
K _ . _
L . _ . .
M _ _
N _ .
O _ _ _
P . _ _ .
Q _ _ . _
R . _ .
S . . .
T _
U . . _
V . . . _
X _ . . _
Y _ . _ _
W . _ _
Z _ _ . .
, _ _ . . _ _

Vlastní program bude vypadat následovně

```

/*****

// program SOS

*****/

// promenne

int    i=0;

sbit   piezo    at  RB0_bit; //piezo je na RB0


// funkce

void    tecka() {
for    (i=0; i<1250; i++) {

    piezo=1;

    Delay_us (160);

    piezo=0;

    Delay_us (160);

    }

    Delay_ms (200);}


void    carka() {
for    (i=0; i<3750; i++) {

    piezo=1;

    Delay_us (160);

    piezo=0;

    Delay_us (160);

    }

    delay_ms (600);}


// hlavni funkce

void main(void){          //zacatek hlavni funkce

// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]

trisB=0b11111110;

trisA=0b11111111;

portB=0b00000000;
```

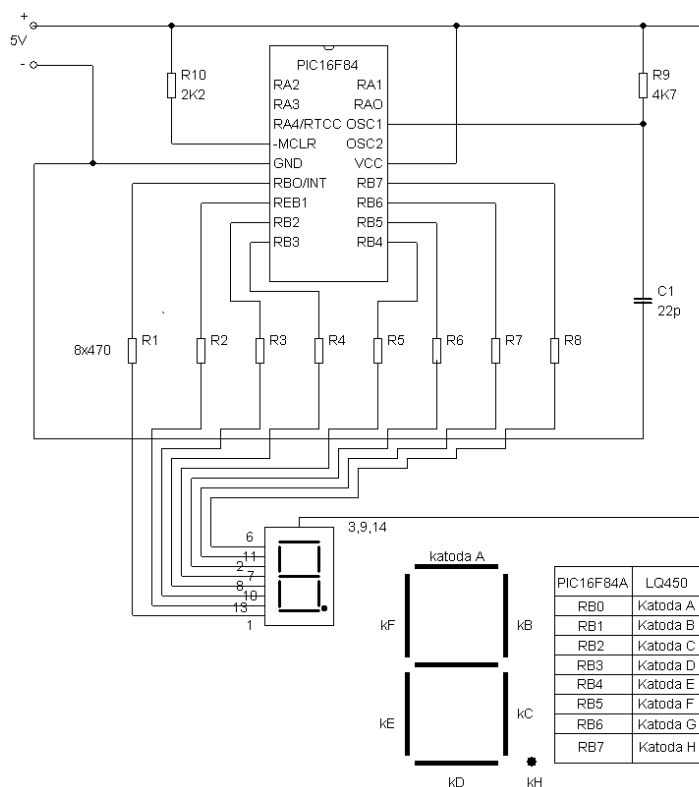
```
while (1){  
    tecka(); // S  
    tecka();  
    tecka();  
    carka(); // 0  
    carka();  
    carka();  
    tecka(); // S  
    tecka();  
    tecka();  
    } // konec nekonecneho cyklu  
} // konec programu
```

4.4.9 Zobrazovače

4.4.9.1 Sedmisegmentový zobrazovač

Zadání úlohy

Na sedmisegmentovém zobrazovači se společnou anodou připojeném na RB0 až RB7 dle schématu (obr. 4.21) zobrazte číslo 1.



Obr. 4.21: Schéma zapojení sedmisegmentového zobrazovače

Výukový účel úlohy

Naučit se pracovat se sedmisegmentovým zobrazovačem.

Teoretický rozbor úlohy

Po celou dobu jsme používali pro zobrazení výstupní hodnoty LED diody. Nyní použijeme sedmisegmentový zobrazovač, což není nic jiného než opět LED diody, které jsou vyrobeny ve tvaru segmentu, který nám umožňuje zobrazovat čísla 0 až 9.

Sedmisegmentové zobrazovače se prodávají v provedení se společnou anodou (náš případ), kdy anody jsou spojeny (u nás vývody 3,9 a 14) a připojeny na kladné napájecí napětí a jednotlivé katody jsou vyvedeny a v našem případě připojeny na PORTB mikroprocesoru. Sedmisegmentové zobrazovače se prodávají také v provedení se společnou katodou, kdy jsou katody spojeny a ovládány z mikroprocesoru jednotlivé anody.

V nastavení I/O nastavíme RB0 jako výstupní, protože je zde připojen sedmisegmentový zobrazovač. Je potřeba si uvědomit, že pro zobrazení čísla 1 je potřeba rozsvítit segment B a C, tj. na připojené katody kB a kC poslat log. 0. Ostatní segmenty nebudou svítit, proto pošleme log. 1 (nastavení registru PORTB je vidět v tab. 4.20).

PORTB 06h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
1	1	1	1	1	0	0	1

Tab. 4.20: Nastavení registru PORTB

Vlastní program bude vypadat následovně

```

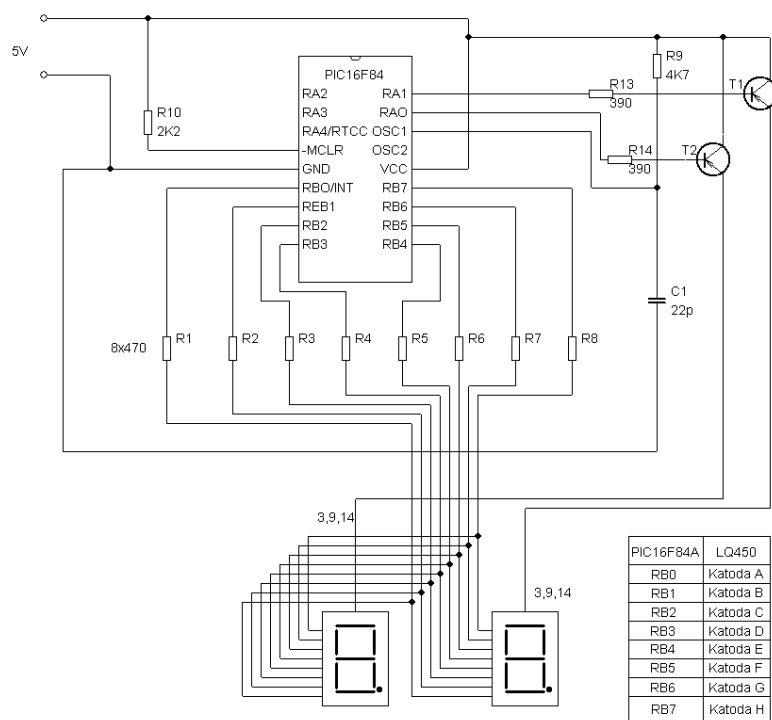
/*****
// rozsvícení čísla jedna na sedmisegmentovém zobrazovací
/*****
void main(void){          //zacatek hlavni funkce
// nastavení vstupu a výstupu, 1 - vstup(input), 0 / výstup(output]
    TRISB=0b00000000;      //PORTB nastaven jako výstupní
    PORTB=0b11111001;      //rozsvítí číslo 1 na sedmisegmentovce
}

```

4.4.9.2 Sedmisegmentový zobrazovač – multiplexní řízení

Zadání úlohy

Na dvojici sedmisegmentových zobrazovačů se společnou anodou připojeném na RB0 až RB7 dle schématu (obr. 4.22) zobrazte číslo 12.



Obr. 4.22: Schéma zapojení obvodu

Výukový účel úlohy

Naučit se pracovat s více sedmisegmentovými zobrazovači za použití multiplexního řízení.

Teoretický rozbor úlohy

Opět jako v předcházející úloze máme sedmisegmentové zobrazovače se společnou anodou, opět budou vývody 3,9 a 14 připojeny na kladné napájecí napětí, ale tentokrát ne přímo, ale přes tranzistory PNP připojené na RA0 a RA1 mikroprocesoru.

Abychom mohli naráz rozsvítit dvě různá čísla, musíme použít multiplexní řízení displeje, tj. musíme zajistit přepínání jednotlivých sedmisegmentových jednotek s určitou frekvencí. Tuto frekvenci lze odhadnout ze schopnosti lidského oka odlišit od sebe jednotlivé, rychle se měnící děje. Z kinematografie víme, že již 24 snímků za sekundu by mělo stačit. Lepší je však použít frekvenci minimálně 50Hz. Maximální frekvence také není libovolná. Nesmíme

překročit povolený frekvenční rozsah použitých součástek (LED, tranzistory) a musíme myslet na to, že čím větší frekvence, tím větší úbytky budeme mít na řídicím prvku (tranzistoru) a tím méně bude LED displej svítit. Ale to je kritické až od určité frekvence.

Abychom ještě více podpořili optický klam, budeme sedmisegmentovky rozsvěcet v pořadí 1, 2.

Vlastní program bude vypadat následovně

```
//*****
// multiplexni rizeni sedmisegmentoveho zobrazovace
//*****

// promenne
    sbit Segment1 at ra0_bit;    //Tranzistor 1
    sbit Segment2 at ra1_bit;    //Tranzistor 2

//podprogramy pro rozsviceni cisel na sedmisegmentovce

void jedna(void){                //rozsviti cislo 1 na sedmisegmentovce
    PORTB=0b11111001;
}

void dva(void){                  //rozsviti cislo 2 na sedmisegmentovce
    PORTB=0b10100100;
}

int main(void){                  //zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISA=0b11111100;           //RA0 RA1 jako vystupy, zbytek PORTA vstupni
    TRISB=0b00000000;           //PORTB nastaven jako vystupni

    PORTB=0b11111111;

    while(1){                    //zacatek nekonecneho cyklu while   (nekonecny = 1 )

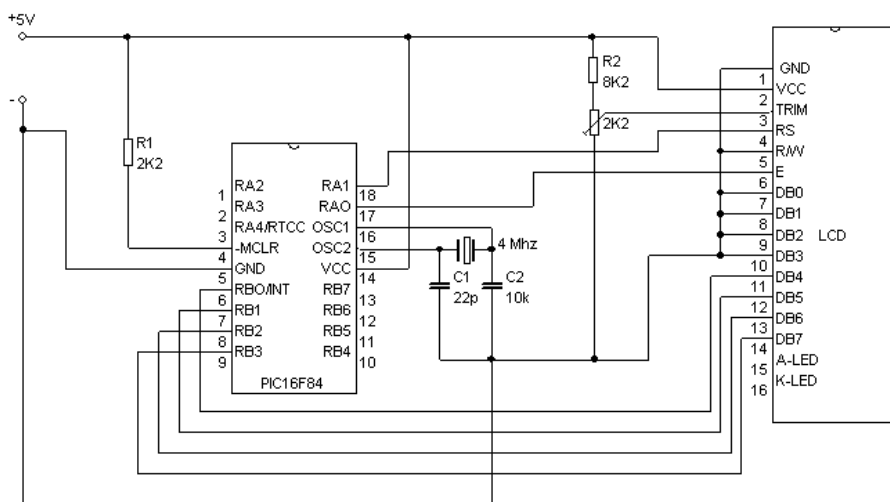
        Segment1=0;              //zapnout pravy zobrazovac - baze NPN tranzistoru
        Segment2=1;              //vypni levy zobrazovac
        jedna();                  //zobrazeni cisla 1
        delay_ms(5);             //cekej 5 ms

        Segment1=1;              //vypni pravy zobrazovac
        Segment2=0;              //zapni levy zobrazovac
        dva();                    //zobrazeni cisla 2
        delay_ms(5);             //cekej 5 ms
    }
}
```

4.4.9.3 LCD zobrazovač

Zadání úlohy

Na LCD zobrazovači připojeném k PIC16F84A dle schématu (obr. 4.23) zobrazte postupně krátké texty „SOSE, COP, Hluboka nad Vlt.“ a „Hluboka nad Vlt., navstivte nas“. Pro oživení použijte efekt, který bude postupně znaky posouvat na jednu a druhou stranu.



Obr. 4.23: Schéma zapojení obvodu

Výukový účel úlohy

Naučit se pracovat s LCD zobrazovačem.

Teoretický rozbor úlohy

LCD zobrazovač je připojen k PIC16F84A, konkrétně na RA0 Enable, na RA1 RS, pro vlastní datovou komunikaci realizovanou čtyřbitově jsou využity nižší čtyři piny PORTu B, tedy RB0 až RB3.

Použitý LCD zobrazovač je LCD displej se 16 znaky v každém ze dvou řádků, které jsou tvořeny maticí 5x8 bodů. Pomocí mikrokontroléru komunikujeme se standardním řadičem HD44780, který využívá mnoho výrobců.

Pro komunikaci s LCD displejem je nutné na PIC připojit 6 až 11 vývodů. V našem případě se jedná o čtyřbitovou komunikaci se 6 připojenými vývody (výstupy mikrokontroléru), kdy pomocí RS a R/W (Read/Write není nastavováno – log. 0 znamená pouze zápis, kdy data pouze

posíláme, ale nečteme) a datových vodičů přenášíme do LCD data pro zobrazení. Pro potvrzení zápisu dat se používá enable.

Vlastní program bude vypadat následovně

```
// pripojeni LCD modulu
sbit LCD_RS at RA1_bit;
sbit LCD_EN at RA0_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISA1_bit;
sbit LCD_EN_Direction at TRISA0_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

// definovani zobrazovaneho textu
char txt1[] = "SOSE, COP";
char txt2[] = "Hluboka nad Vlt";
char txt3[] = "Hluboka nad Vlt";
char txt4[] = "navstivte nas";

// promenne
char i;

void Move_Delay() { // funkce pro efekt
    Delay_ms(500); // rychlost efektu
}

void main(){ // zacatek hlavni funkce
    PORTB = 0xFF;
    TRISB = 0xFF; // definice vstupu a vystupu

    Lcd_Init(); // inicializace LCD

    while(1) { // nekonecna smycka

        Lcd_Cmd(_LCD_CLEAR); // vymazani displeje
        Lcd_Cmd(_LCD_CURSOR_OFF); // vypnuti kurzoru
        Lcd_Out(1,1,txt3); // zapis prvnioho radku - txt3
        Lcd_Out(2,1,txt4); // zapis druheho radku - txt4
        Delay_ms(2000);
```

```

Lcd_Cmd(_LCD_CLEAR);          // vymazani displeje

Lcd_Out(1,1,txt1);            // zapis prvnio radku - txt2
Lcd_Out(2,1,txt2);            // zapis druheho radku - txt2
Delay_ms(2000);

// presun textu
for(i=0; i<2; i++) {          // presun textu o ctyri znaky doprava
    Lcd_Cmd(_LCD_SHIFT_RIGHT);
    Move_Delay();
}

for(i=0; i<2; i++) {          // presun textu o 7 pozic doleva
    Lcd_Cmd(_LCD_SHIFT_LEFT);
    Move_Delay();
}

for(i=0; i<2; i++) {          // // presun textu o 7 pozic doprava
    Lcd_Cmd(_LCD_SHIFT_RIGHT);
    Move_Delay();
}

} // konec nekonecne smycky
} // konec programu

```

4.4.10 Paměť EEPROM

Zadání úlohy

Uložte zadanou hodnotu D'100' do paměti EEPROM.

Výukový účel úlohy

Naučit se pracovat s EEPROM pamětí.

Teoretický rozbor úlohy

Až do této doby jsme tvořili programy, které po odpojení napájení neukládali žádné hodnoty či nastavení. Chceme-li uchovat nějakou hodnotu pro pozdější použití, musíme jí uložit do EEPROM paměti, která má kapacitu 64 osmibitových čísel, tedy 64 bajtů.

Paměť EEPROM není přímo adresovatelná (nelze použít přímé adresování), musíme použít nepřímé adresování při využití čtyř registrů: EECON1, EECON2, EEDATA a EEADR.

Registr EEDATA je osmibitový pracovní registr pro přístup do paměti EEPROM, který obsahuje data pro zápis či čtení. Registr EEADR je ukazatel na buňku paměti EEPROM, se kterou se bude pracovat. Adresa je z rozsahu 0h až 3Fh. Práce s pamětí EEPROM probíhá po osmi bitech, tedy celých bajtech, přičemž automaticky dochází k vymazání dříve uložené hodnoty v příslušné paměťové buňce.

EECON 88h							
-	-	-	EEIF	WRERR	WREN	WR	RD

Tab. 4.21: Registr EECON

Bity registru EECON:

Bity 7 až 5 nejsou použity. EEIF je příznakový bit vnitřního přerušení od dokončení zápisu do paměti EEPROM (1-zápis byl dokončen). WRERR je příznakový bit chyby při zápisu do paměti EEPROM (1-zápis nebyl ukončen řádně). Bit WREN 1-povoluje/0-zakazuje zápis do paměti EEPROM. Zatímco WR je řídicí bit pro zápis dat do EEPROM, tak RD je řídicí bit pro čtení paměti EEPROM.

Kompletní program

```

//*****
// ukazka prace s EEPROM pameti
// Program pro zapis hodnoty 100 do pametove bunky 0x02
//*****

void main(void){          // zacatek hlavni funkce
// nastaveni vstupu a vystupu, 1 - vstup(input), 0 / vystup(output]
    TRISB=0b00000000;      // RB0 nastaven jako vystupni
    PORTB=0b00000000;      // uvodni inicializace

    while(1){              //zacatek nekonecneho cyklu while  (nekonecny = 1 )

        EEPROM_Write(0x02,100);    // Zapise data 100 na adresu 0x02
        Delay_ms(20);              // Mezi zapisem a ctenim min 20 ms prodleva
        PORTB = EEPROM_Read(0x02); // Precete data z adresy 0x02 a zobrazi na PORTB
        Delay_ms(20);
    }    // konec nekonecneho cyklu while

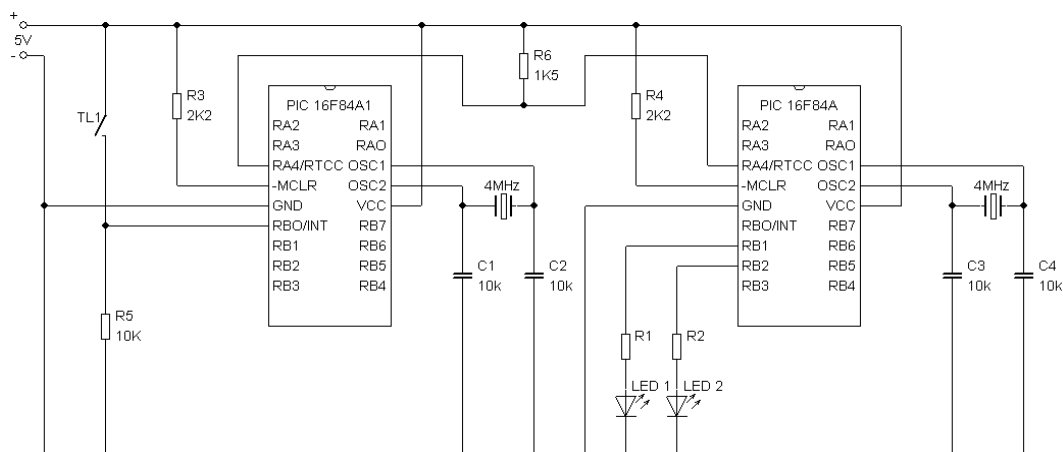
}    // konec programu

```

4.4.11 Komunikace PIC16F84A pomocí 1-vodičové sběrnice DALLAS

Zadání úlohy

Pro dvojici PIC16F84A zapojených dle schématu (obr. 4.24) napište program, kdy bude komunikovat první PIC16F84A pomocí 1-vodičové sběrnice Dallas s druhým PIC16F84A. Po stisku tlačítka připojeného k PIC1 se rozsvítí příslušný LED2 dioda zapojená k PIC2, nebude-li tlačítko stisknuté, bude svítit LED1.



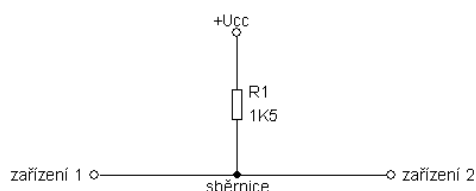
Obr. 4.24: Schéma zapojení

Výukový účel úlohy

Vyzkoušet komunikaci mikroprocesorů po 1-vodičové Dallas sběrnici.

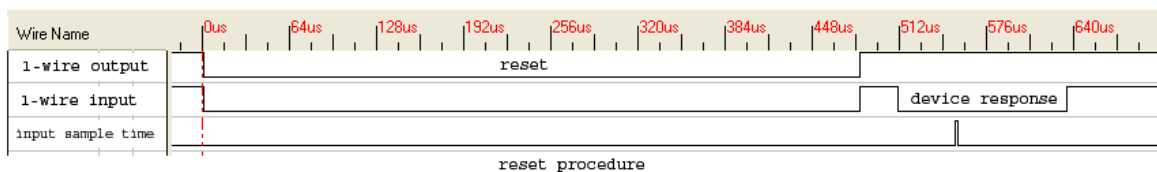
Teoretický rozbor úlohy

Sběrnici ke komunikaci používá řada obvodů, např. teploměr DS18B20. Neskládá se, jak by se mohlo zdát jen z jediného vodiče, ale z vodičů dvou (obr. 4.25). Jedním z nich, zemí, jsou zařízení přímo propojené a druhý, současně slouží pro napájení obou zařízení. Tento typ sběrnice je možné použít až do vzdálenosti 300 m.



Obr. 4.25: Schéma zapojení 1-wire sběrnice

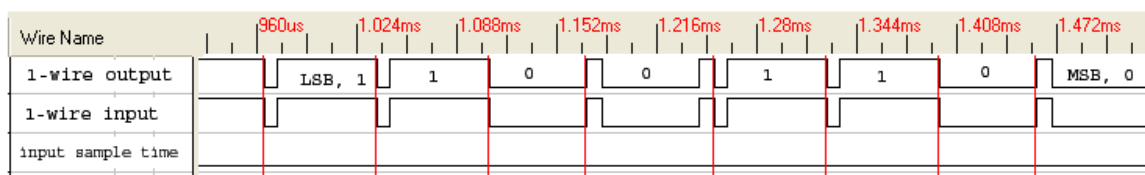
Vlastní komunikace začíná vysláním pulzu RESET:



Obr. 4.26: puls RESET / PRESENCE [17]

První zařízení vyšle 480 μ s dlouhý pulz RESET a je-li na stejné sběrnici připojeno další zařízení Dallas, odpoví do 60 μ s pulzem PRESENCE. Po tuto dobu první zařízení naslouchá dění na sběrnici. Poté může začít další komunikace.

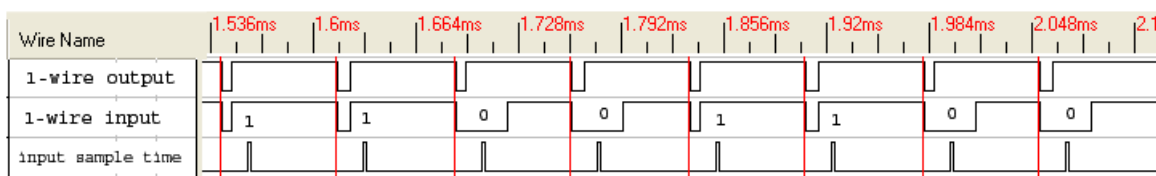
Zápis (zaslání) bajtu:



Obr. 4.27: zápis bajtu o dekadické hodnotě 33, v bin. soustavě 0b00110011 [17]

Pro zápis „nuly“ se využívá 60 μ s dlouhá úroveň L. Další komunikace může pokračovat nejméně po 1 μ s. Zápis „jedničky“ je realizován zápisem 5 μ s dlouhé úrovně L a poté může další komunikace pokračovat po nejméně 56 μ s.

Čtení (příjem) bajtu:



Obr. 4.28: čtení bajtu o dekadické hodnotě 33, v bin. soustavě 0b00110011 [17]

Čtení je přesně inverzní operace zápisu, proto platí to, co již bylo uvedeno výše:

Pro čtení „nuly“ se využívá 60 μ s dlouhá úroveň L. Další komunikace (čtení) může pokračovat nejméně po 1 μ s.

Čtení „jedničky“ je realizováno 5 μ s dlouhou úrovní L, poté může další komunikace pokračovat po nejméně 56 μ s.

Tyto zdrojové kódy demonstrují komunikaci dvou jednočipových mikrokontrolérů s využitím 1-wire sběrnice pro přenos informace z PIC1 (vlevo) do PIC2 (vpravo). Mikrokontroléry jsou přímo spojeny piny RA4.

Kompletní programy

Vlastní program pro 1. PIC – vysílač bude vypadat takto:

```

/*****
// one wire sběrnice - vysílač
*****/

// proměnné
sbit TLACITKO at rb0_bit;

void main() {          // začátek hlavní funkce
    TRISB=0b11111111;   // PORTB nastaven jako výstupní
    do { // začátek nekonečné smyčky
        Ow_Reset(&PORTA, 4); // 1-wire resetovací signal
        if (TLACITKO==1) {
            Ow_Write(&PORTA, 4, 0b00000100); // na 1-wire zápis hodnoty 00000100
            Delay_us(120);
        }
        else {
            Ow_Write(&PORTA, 4, 0b00000010); // na 1-wire zápis hodnoty 00000010
            Delay_us(120);
        }
    } while (1); // konec nekonečné smyčky
}

```

Na straně 2. PIC – přijímače bude tento zdrojový kód:

```

/*****
// 1-wire sběrnice - přijímač
*****/

void main() {          //začátek hlavní funkce

    TRISB=0b11111001;   //RB1 a RB2 nastaveny jako výstupy
    PORTB=0b00000000;   //zhasnutí LED diod
    do {
        Ow_Reset(&PORTA, 4); // 1-wire resetovací signal
        PORTB = Ow_Read(&PORTA, 4); // zaslana hodnota na PORTB
    } while (1);
}

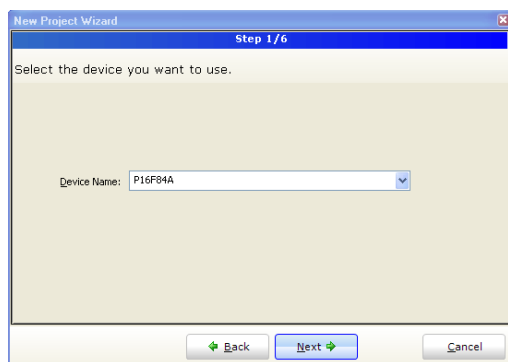
```

4.5. Příloha – Prostředí Mikroelektronika MikroC

Mikroelektronika - Založení nového projektu v MikroC for PIC

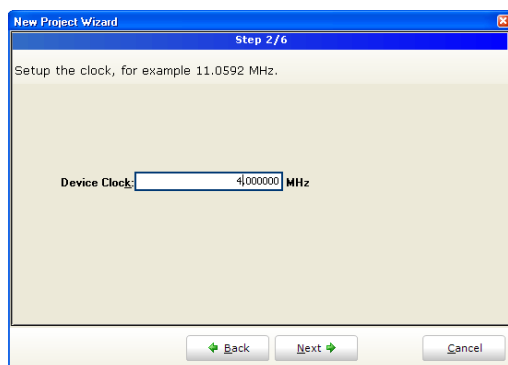
Pro psaní projektů v jazyce C použijeme nyní vývojové prostředí od společnosti Mikroelektronika s názvem MikroC for PIC. Prvním krokem bude založení nového projektu.

Nový projekt založíme pomocí průvodce, který se automaticky spustí po kliknutí v menu **Project/New Project (Shift+Ctrl+n)**. Dále projdeme průvodce. Po prvním přivítacím okně, kde jen klikneme na tlačítko **Další**, následuje okno s názvem **Step One: Select Device**, kde vybereme typ součástky, se kterou budeme pracovat, v našem případě to bude PIC16F84A (obr. 4.29).



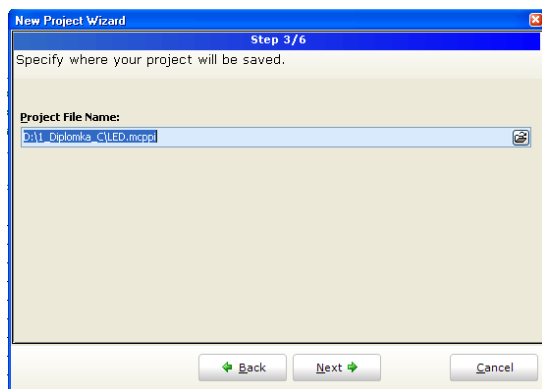
Obr. 4.29: Krok 1 – výběr součástky

Pokračujeme stiskem tlačítka **Další (Next)**. Ve druhém kroku (obr. 4.30) **nastavíme taktovací frekvenci**, která v našem případě bude 4 MHz.



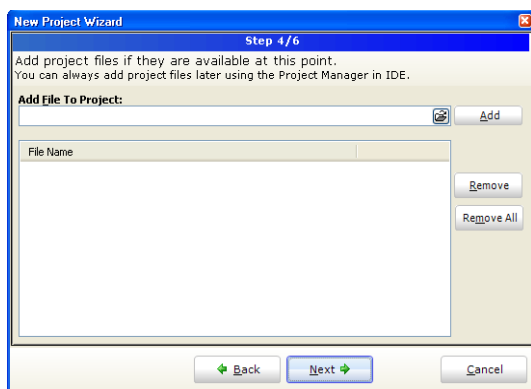
Obr. 4.30: Krok 2 – nastavení taktovací frekvence

Ve třetím kroku průvodce (obr. 4.31) **zvolíme jméno** pro náš nový projekt (např. LED – ve jméně nepoužíváme českou diakritiku ani mezery a tečky), a zadáme cestu k adresáři, kde bude náš projekt uložen (k tomu využijeme tlačítko Browse).



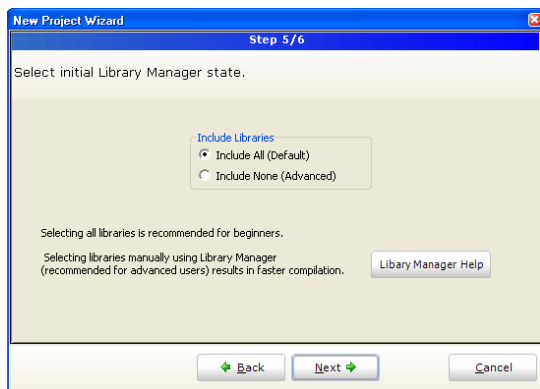
Obr. 4.31: Krok 3 – výběr umístění pro uložení projektu

Ve čtvrtém kroku (obr. 4.32) můžeme do projektu **přidat zdrojový soubor**, pokud již existuje. Neexistuje-li, pokračujeme pouze kliknutím na tlačítko Další.



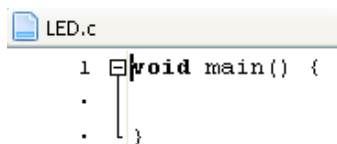
Obr. 4.32: Krok 4 – vložení existujících zdrojových souborů do projektu

V předposledním okně (obr. 4.33) volíme, které knihovny připojíme do projektu. Pro začátečníky je doporučena implicitně nastavená volba **Include All** (tedy všechny knihovny).



Obr. 4.33: Summary – rekapitulace zadaných údajů

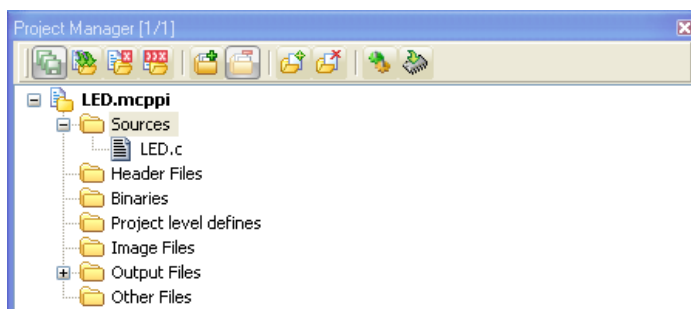
Poslední šestý krok průvodce jen oznámí úspěšné založení nového projektu. Po kliknutí na tlačítko **Dokončit (Finish)** se průvodce uzavře. Tímto máme založen nový projekt se zdrojovým souborem, kde se nachází prázdná hlavní funkce `main{}` (obr. 4.34), do které můžeme psát zdrojový kód.



Obr. 4.34: Hlavní funkce `main()` pro psaní zdrojového kódu

Vložení zdrojového souboru do projektu

Před překladem hotového zdrojového souboru musí být tento zdrojový soubor vložen v projektu. Vložení zdrojového souboru probíhá sice automaticky v době zakládání nového projektu, ale někdy může vzniknout potřeba do projektu zahrnout jiný zdrojový soubor. Toto je možné dosáhnout v okně Project Manageru, který (obr. 4.35) zobrazíme z menu **View/ Project Manager**.



Obr. 4.35: Okno Project Manageru

Pro vložení zdrojového souboru do projektu označíme v okně projektu **Sources** a po kliknutí pravým tlačítkem myši se zobrazí nabídka, kde zvolíme **Add Files To Project**, a v následně zobrazeném okně zvolíme zdrojový soubor, který chceme vložit do projektu. Zdrojový soubor z projektu vymažeme tak, že ho označíme a zvolíme z kontextového menu **Remove File From Project**.

Překlad (kompilace) zdrojového souboru

Z menu **Project / Build all** (nebo stisk klávesy Ctrl+F9). V okně Output se objeví hlášení překladače, pokud je všechno v pořádku, objeví se **FINISHED SUCCESFULLY** a pokud jsou někde nějaké chyby, překladač napíše **FINISHED (WITH ERRORS)** a navíc červeně jsou vypsány čísla řádků a typ chyby, které se ve zdrojovém kódu nacházejí (obr. 4.36).

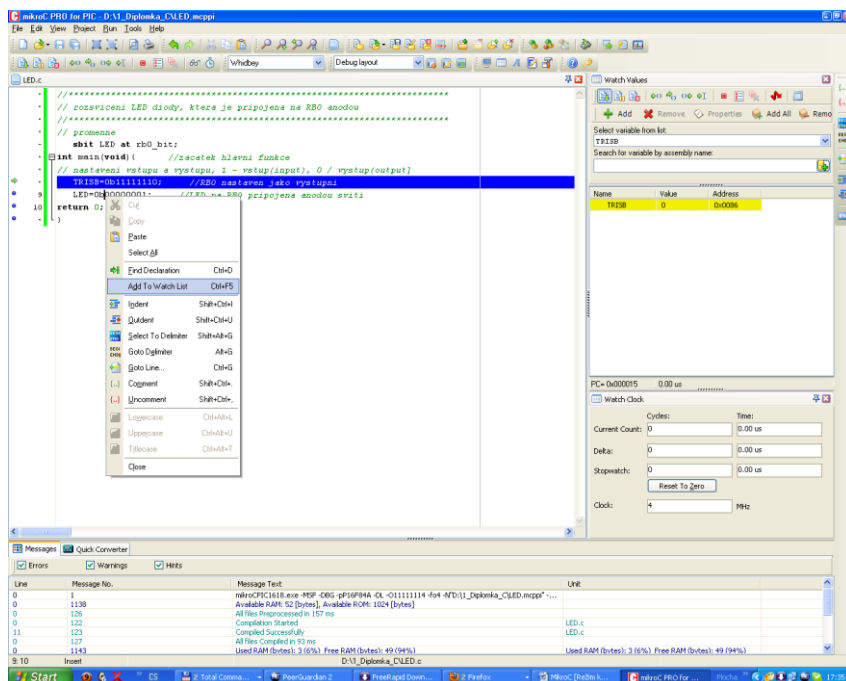
Line	Message No.	Message Text	Unit
0	1	mikroC PIC1618.exe -MSF -DBG -pP16F84A -DL -O111111114 -fo4 -N"D:\1_Diplomka_C...	
0	1138	Available RAM: 52 [bytes], Available ROM: 1024 [bytes]	
0	126	All files Preprocessed in 157 ms	
0	122	Compilation Started	LED.c
11	123	Compiled Successfully	LED.c
0	127	All files Compiled in 93 ms	
0	1143	Used RAM (bytes): 3 (6%) Free RAM (bytes): 49 (94%)	Used RAM (bytes): 3 (6%) Free RAM (bytes): 49 (94%)
0	1143	Used ROM (program words): 26 (3%) Free ROM (program words): 998 (97%)	Used ROM (program words): 26 (3%) Free ROM (progra...
0	125	Project Linked Successfully	LED.mcppi
0	128	Linked in 188 ms	
0	129	Project 'LED.mcppi' completed: 547 ms	
0	103	Finished successfully: 08 XI 2009, 17:30:56	LED.mcppi

Obr. 4.36: Překlad zdrojového kódu

Simulace

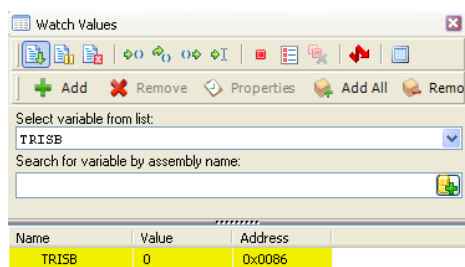
MikroC Pro podobně jako MPLab nám umožňuje zobrazit obsah Speciálních funkčních registrů, či nadefinovaných proměnných. Pomocí simulace si můžeme otestovat program, zda plní správně svoji funkci. MPLAB simulátor spustíme z horního menu **Run / Start Debugger**.

Nesmíme zapomenout, že zdrojový soubor musí být před spuštěním simulace zkompilován (Project / Build all nebo stisk klávesy Ctrl+F9).



Obr. 4.37: Simulátor MikroC Pro

Speciální funkční registry či proměnné vkládáme buď výběrem ze seznamu **Select variable from list** a poté klikneme na zelené plus v okně **Watch Values** (obr. 4.38) nebo přímo ze zdrojového kódu tak, že klikneme na příslušnou proměnnou ve zdrojovém kódu a po kliknutí na pravé tlačítko z kontextového menu vybereme položku **Add To Watch List** (obr. 4.37).



Obr. 4.38: Okno Watch Values

Horní řada ikon nám pak dovoluje simulaci programu. Najdeme zde podobná tlačítka, jako tomu bylo např. v MPLabu, ty nejběžnější, které se využívají při simulaci, si nyní popíšeme:

Step Into (F7) - při ladění programu provede jeden řádek (příkaz) programu. Pokud je na řádku volána programová rutina, provede se její otevření a přesun na první příkaz.

Step Over (F8) - při ladění programu provede jeden řádek (příkaz) programu. Pokud je na řádku volána programová rutina, provede se celá.

Step Out (CTRL+F8) - při opuštění programové rutiny.

Start Debugger (F9) - vykoná reset procesoru.

Run / Pause Debugger (F6) – rozběhnutí programu až do nalezení dalšího breakpointu / zastavení programu.

Breakpoint – značka, která způsobí zastavení běžícího programu, breakpointy obvykle vkládáme na významné řádky, kde chceme zjistit hodnotu proměnné.

Rychlý přehled

Založení nového projektu: v menu Project/New Project

(nový zdrojový soubor je založen automaticky při založení nového projektu – přípona .c)

Zobrazení Project Manageru, z menu View/ Project Manager.

Přidání zdrojového souboru do projektu: V Project Manageru Sources / pravé tlačítko myši - Add Files To Project.

Překlad zdrojového souboru: Z menu Project / Build all (nebo stisk klávesy Ctrl+F9).

Simulace: spuštění simulátoru z horního menu Run / Start Debugger (F9).

Zobrazení hodnot SFR či proměnných: označíme příslušnou proměnnou ve zdrojovém kódu a po kliknutí na pravé tlačítko z kontextového menu vybereme položku Add To Watch List.

Otázky k procvičování

1. Jaká architektura je použita u PIC16F84A?
2. Kolik zdrojů přerušení má PIC16F84A a jaké?
3. Jaký je rozdíl mezi proměnnou a konstantou v jazyce C?
4. Jaký je rozdíl mezi cykly while a do-while?
5. Uveďte, co je potřeba udělat pro to, aby LED dioda zapojená na RB4 anodou po připojení napájení svítila?
6. Jak dlouho trvá provedení 1 instrukčního cyklu při použití krystalu s frekvencí 4 MHz?

7. Jakým způsobem je možné ošetřit zákmity tlačítka?
8. Je možné k PIC16F84A přímo připojit krokový motor?
9. Popište způsob ovládání modelářského serva pomocí jednočipového mikropočítače?
10. K čemu slouží u PIC16F84A paměť EEPROM?

Odpovědi na otázky za textem

1. PIC16F84A používá *Harvardskou architekturu*, to znamená, že má navzájem oddělenou paměť programu (velikost 1024 B) a paměť dat (RAM 68 B a EEPROM 64 B).

2. PIC16F84A má 4 zdroje přerušení: dva vnější - vstup RB0/INT a čtyři vyšší bity PORTu B <7:4> a dva vnitřní - přetečení od časovače TMR0 a ukončení zápisu dat do paměti EEPROM). Tyto zdroje mohou vyvolat přerušení na základě vnější události.

3. *Hodnota proměnné může být během výpočtu měněna*. Tím se proměnné zásadně liší od konstant, které mají po celou dobu chodu programu hodnotu stejnou - *konstantní*.

4. Podmínka cyklu *while* se testuje před průchodem cyklu, což znamená, že cyklus (příkaz) tedy vůbec nemusí proběhnout.

```
while (výraz)
    příkaz
```

U cyklu *do-while* je podmínka testována až po průchodu tělem cyklu, což zajišťuje alespoň jedno provedení těla cyklu (příkazu).

```
do
    příkaz
while (výraz);
```

5. K tomu, abychom rozsvítili LED diodu, která je připojená na pin RB0 PIC16F84A anodou, *musíme nejdřív nastavit vstupně-výstupní vývod RB4 jako výstupní*. To provedeme konkrétně v registru TRISB na adrese 86h následujícím zápisem:

TRISB 86h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
1	1	1	0	1	1	1	1

Z RB4 uděláme výstup, ostatní I/O piny Portu B ponecháme jako vstupní.

PORTB 06h							
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			1				1

Logickou 1 zapíšeme na pozici RB4 v registru PORTB na adrese 06h. To nám zajistí, že se na RB4 objeví napětí +Ucc (to odpovídá úrovni log. 1).

6. *1 mikrosekundu*, protože frekvence použitého oscilátoru je 4 miliony kmitů za jednu sekundu. Počet instrukčních cyklů = $4\,000\,000/4 = 1\,000\,000$ instrukčních cyklů za 1 s, a odtud $T = 1/f$, tj. $1/1\,000\,000$ a to odpovídá právě 1 mikrosekundě.

7. Máme-li tlačítko, na kterém jsou zákmity, je vhodné softwarové řešení, kdy jde o to, že v době, kdy zákmity probíhají, se vykonává program a mikroprocesor nesmí v této době provádět test zakmitávajícího tlačítka. To se dá vyřešit např. vhodně umístěnou časovou smyčkou. Délka časové smyčky může být okolo 50 ms (obr. 4.12).

8. Ne, přímé připojení není možné. Vzhledem k velkým proudům odebírajících cívek jsou použity tranzistory (obr. 4.15). Maximální hodnota pro výstup u PIC16F84A je pouze 20 mA na jeden výstup.

9. Žádaná hodnota natočení výstupní hřídele je ve formě šířkově modulovaného signálu (s úrovní TTL) s periodou 20 ms a šířkou pulzu mezi cca 1 a 2 ms. (obr. 4.19).

Do střední polohy natočíme servo tedy tak, že na příslušný výstup mikroprocesoru budeme posílat po určité minimální dobu impuls 1,5 ms v rámci 20 ms dlouhém (tj. 1,5 ms log. 1 a 18,5 ms log. 0). Nestačí jen jednou, ale je třeba určitého počtu těchto impulsů, aby se servo mělo čas do požadované polohy dotočit. Je třeba si uvědomit, že servo potřebuje cca 0,8 s na to, aby se dostalo z jedné do druhé krajní polohy. To odpovídá 40 rámcům po 20 ms.

10. Paměť EEPROM slouží pro uchování hodnot pro pozdější použití. Tyto hodnoty zůstanou uloženy v paměti EEPROM i po odpojení napájení. Uchovat můžeme až 64 osmibitových čísel, tedy 64 bajtů – kapacita EEPROM.

Test: základy elektroniky

1) Které z těchto prvků jsou lineární

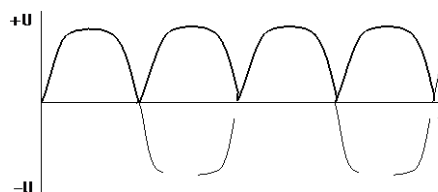
- a) odpor
- b) dioda
- c) tranzistor
- d) led dioda

2) Které z těchto prvků jsou pasivní součástky

- a) skříň
- b) vodič
- c) tranzistor
- d) kondenzátor

3) Který z usměrňovačů má průběh uvedený na obrázku

- a) jednocestný
- b) řízený
- c) dvoucestný
- d) trojcestný

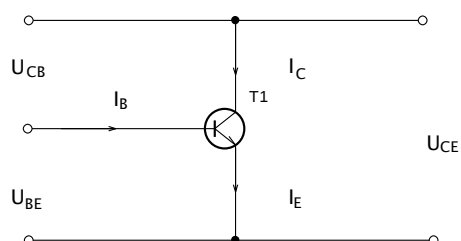


4) Trojúhelníček u schématické značky diody značí

- a) bázi
- b) kolektor
- c) anodu
- d) katodu

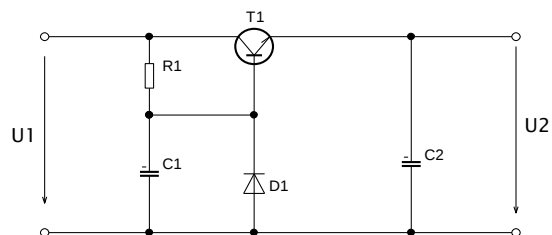
5) Šipka u schématické značky tranzistoru značí

- a) bázi
- b) emitor
- c) kolektor
- d) označení směru proudu elektronů



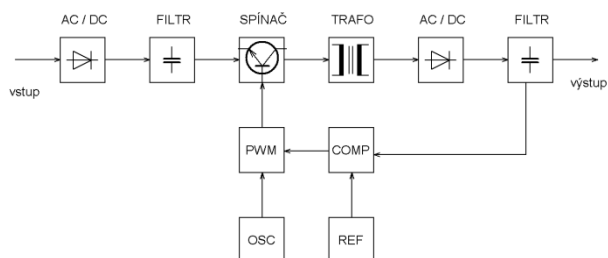
6) Na uvedeném obrázku se jedná o

- a) zesilovač
- b) stabilizátor napětí sériový
- c) filtrační obvod
- d) paralelní stabilizátor



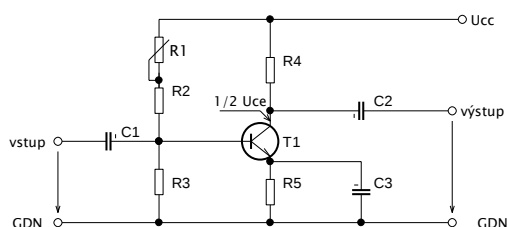
7) O jaký typ zdroje se jedná na následujícím obrázku

- a) lineární zdroj
- b) pulzně kvantový zdroj napětí
- c) výkonový zesilovač
- d) spínaný zdroj



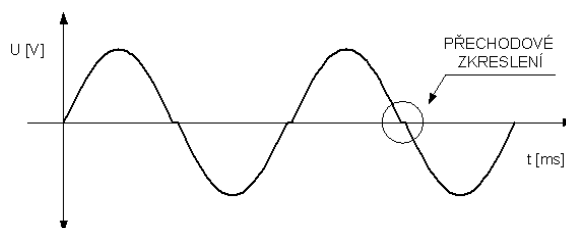
8) O jaký typ zesilovače se jedná na následujícím obrázku

- a) jednostupňový zesilovač se společnou bází třída B
- b) jednostupňový zesilovač třída A zapojení se společným emitorem
- c) jednostupňový zesilovač třída A zapojení se společným kolektorem
- d) jednostupňový tvarovač signálu



9) U jakého typu koncového zesilovače dochází k přechodovému zkreslení na obrázku

- a) třída A
- b) třída B
- c) třída D
- d) třída C

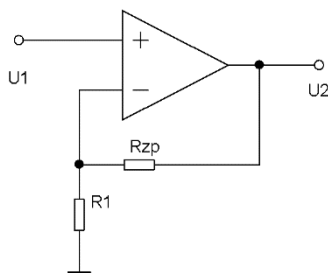


10) Jaký typ zdroje použijete pro napájení operačního zesilovače

- a) zdroj + 15 V
- b) zdroj 0 V
- c) zdroj -10 V
- d) zdroj symetrický ± 15 V

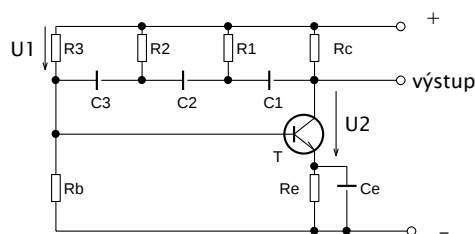
11) O jaký typ zapojení operačního zesilovače se jedná

- a) invertující zesilovač
- b) neinvertující zesilovač
- c) integrační zesilovač
- d) sumační zesilovač



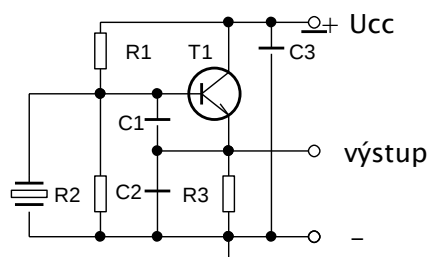
12) O jaké zapojení uvedené na obrázku se jedná

- a) zesilovač s kapacitní vazbou
- b) oscilátor RC
- c) oscilátor LC
- d) koncový stupeň zesilovače



13) O jaký typ oscilátoru se jedná

- a) LC oscilátor
- b) RC oscilátor
- c) krystalový oscilátor
- d) krystalový zesilovač



Řešení testu:

1 a

2 d

3 c

4 c

5 b

6 b

7 d

8 b

9 b

10 d

11 b

12 b

13 c

Test: číslicová technika

1. Sedmisegmentová zobrazovací jednotka LED při zobrazení číslice 2 odebírá proud 100 mA. Zobrazovací jednotka může odebírat maximálně:

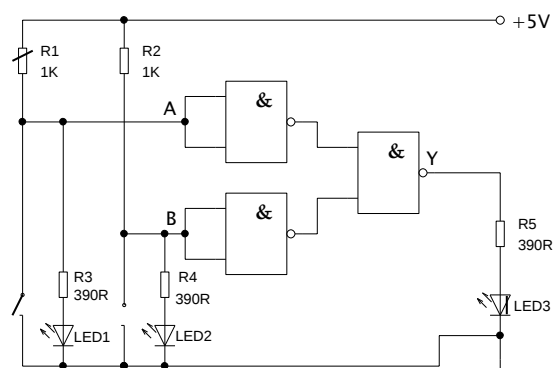
- a) 70 mA
- b) 100 mA
- c) 160 mA
- d) 140 mA

2. Určete, jaké tvrzení je správné:

- a) Nejjednodušším kombinačním obvodem je RS klopný obvod.
- b) Kombinační logické obvody neobsahují paměťové členy, výstupní hodnoty jsou určeny okamžitými hodnotami na vstupech.
- c) Mezi kombinační logické obvody patří dekodéry, čítače a posuvné registry.
- d) Kombinační logické obvody mají paměť o velikosti 1 bitu a řadíme sem např. RS, D a J-K klopné obvody.

3. Zapojení složené z trojice hradel NAND dle obrázku odpovídá logickému členu:

- a) AND
- b) OR
- c) NAND
- d) NOR



4. Vztah $\overline{A+B} = \overline{A}.\overline{B}$ se nazývá:

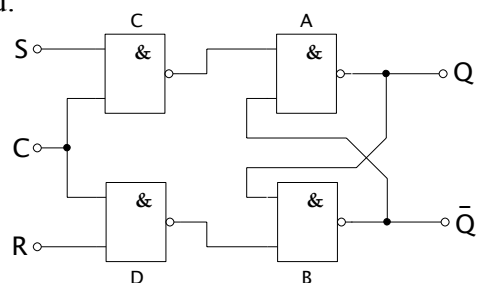
- a) zákon asociativní
- b) zákon disociativní
- c) pravidlo absorpce
- d) Teorém De Morganův

5. Pokud je kdekoliv na vstupu úroveň log. 1, je na výstupu také úroveň log. 1. Jedná se o hradlo:

- a) OR
- b) NOR
- c) AND
- d) NAND

6. Níže uvedené schéma představuje klopný obvod:

- a) RS
- b) RST
- c) D
- d) J-K



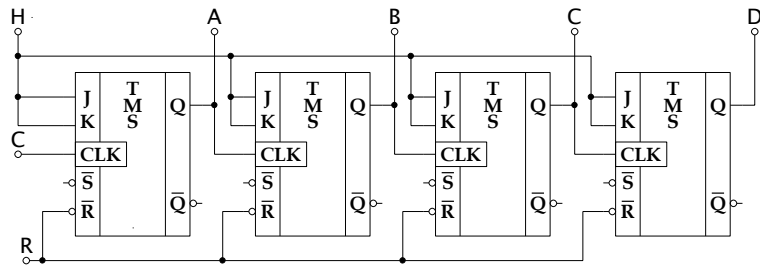
7. Výsledná funkce Y po minimalizaci má hodnotu:

- a) $Y = B * C + \overline{B} * \overline{C} + B * \overline{D}$
- b) $Y = B + \overline{B} * \overline{C} + \overline{C} * \overline{D}$
- c) $Y = A * B + \overline{B} * \overline{C} + B * \overline{D}$
- d) $Y = A + \overline{B} * \overline{C} + \overline{C} * \overline{D}$

		B			
		A			
C	D	1	1	1	1
		0	0	1	1
		0	0	1	1
		1	1	0	0

8. Na obrázku je:

- a) asynchronní čítač
- b) synchronní čítač
- c) posuvný registr
- d) demultiplexer



9. Komparátor je:

- a) je logický obvod, který umožňuje přepínat jednu vstupní proměnnou na několik výstupů;
- b) je logický obvod, který umožňuje přepínat několik vstupních proměnných na jeden výstup;
- c) kombinační logický obvod, který porovnává dvě čísla na vstupu a zároveň generuje výstupní signál o rovnosti či nerovnosti;
- d) sekvenční logický obvod, který porovnává dvě čísla na vstupu a zároveň generuje výstupní signál o rovnosti či nerovnosti.

10. Skutečná velikost 1 kB paměti je:

- a) $2^{10} = 1024\text{B}$
- b) $2^{20} = 1048576\text{ B}$
- c) $2^{30} = 1073741824\text{ B}$
- d) $2^3 = 8\text{ B}$

Řešení testu:

1 d

2 b

3 b

4 d

5 a

6 b

7 a

8 a

9 c

10 a

Test: zabezpečovací technika

1. Citlivý na infračervené záření je:
 - a) optický požární detektor
 - b) PIR detektor pohybu
 - c) detektor rozbití skla
 - d) magnetický detektor

2. Magnetický detektor má v klidu kontakt:
 - a) rozepnutý
 - b) sepnutý
 - c) záleží na nastavení ústředny
 - d) nemá kontakt

3. Dělený systém neumožňuje:
 - a) postupné zajišťování objektu
 - b) použití více klávesnic
 - c) nezávislé zajištění dvou částí objektu
 - d) použití jedné ústředny

4. Detektor kouře JA-80S je:
 - a) ionizační
 - b) radioaktivní
 - c) optický
 - d) mechanický

5. Hodnota vyvažovacího odporu drátové smyčky OASISu je:
 - a) 10 k
 - b) 2k2
 - c) 1k
 - d) 4k

6. Označení IW znamená:
- a) mikrovlnné čidlo
 - b) vnitřní siréna
 - c) tísňové tlačítko
 - d) ultrazvukový detektor
7. Doba aktivace sirény je max.:
- a) 12 minut
 - b) 8 minut
 - c) 5 minut
 - d) 1 minuta
8. Pro nastavení ústředny JA-82K pomocí počítače slouží program:
- a) OLink
 - b) GDLink
 - c) ComLink
 - d) GSMLink
9. Bezdrátový systém OASIS pracuje na frekvenci:
- a) 466 MHz
 - b) 2,4 GHz
 - c) 868 MHz
 - d) 27 MHz
10. Maximální délka antény systému OASIS může být přibližně:
- a) 1,4 m
 - b) 6 cm
 - c) 17 cm
 - d) 86 cm

Řešení testu:

1 b

2 b

3 a

4 c

5 c

6 b

7 c

8 a

9 c

10 c

Test: programování jednočipových mikropočítačů

1. Paměť programu u PIC16F84A má kapacitu:

- a) 1 K x 14 bitů
- b) 64 K x 14 bitů
- c) 1 K x 8 bitů
- d) 64 K x 8 bitů

2. PIC16F84A má celkem:

- a) 14 I/O bran rozdělených do 2 portů
- b) 13 I/O bran rozdělených do 2 portů
- c) 12 I/O bran rozdělených do 2 portů
- d) 11 I/O bran rozdělených do 2 portů

3. Cyklus for:

- a) se používá pro jednoduché větvení programu
- b) se používá pro vícenásobné větvení programu
- c) se používá pro předem zadaný počet opakování příkazu nebo bloku příkazů
- d) se používá pro testování stavu tlačítka

4. Jediným z cyklů, který zajišťuje alespoň jedno provedení těla cyklu, tedy příkazu, který je typicky blokem, je cyklus:

- a) switch
- b) for
- c) while
- d) do-while

5. V identifikátoru je možné použít:

- a) jen písmena a .. z a A .. Z a číslice 0 .. 9
- b) písmena a .. z a A .. Z, číslice 0 .. 9 a znak podtržítka (_)
- c) jen písmena a .. z a A .. Z a znak podtržítka (_)
- d) písmena včetně znaků české znakové sady, číslice 0 .. 9 a speciální znaky

6. Pro konfiguraci vstupů/výstupů pro port B musíme zapsat do registru:

- a) PORTB
- b) TRISB
- c) PORTA
- d) TRISA

7. Režim čítače / časovače volíme nastavením nebo nulováním:

- a) třetího bitu registru OPTION, který má název PSA
- b) čtvrtého bitu registru OPTION, který má název TOSE
- c) pátého bitu registru OPTION, který má název TOCS
- d) sedmého bitu registru OPTION, který má název RPBU

8. Globální povolení přerušení se nastavuje pomocí:

- a) sedmého bitu registru INTCON, který má název GIE
- b) šestého bitu registru INTCON, který má název EEIE
- c) pátého bitu registru INTCON, který má název TOIE
- d) čtvrtého bitu registru INTCON, který má název INTE

9. Pro nastavení servomotoru do střední polohy se používá signál s periodou 20 ms:

- a) a šířkou pulzu pro úroveň log. 1 - 1 ms
- b) a šířkou pulzu pro úroveň log. 1 – 1,5 ms
- c) a šířkou pulzu pro log. úroveň 1 - 2 ms
- d) a šířkou pulzu pro log. úroveň 1 - 3 ms

10. Pro připojení maticové klávesnice s 12 tlačítky k PIC16F84A potřebujeme:

- a) 5 volných vstupně výstupních bran
- b) 6 volných vstupně výstupních bran
- c) 7 volných vstupně výstupních bran
- d) 8 volných vstupně výstupních bran

Řešení testu:

1 a

2 b

3 c

4 d

5 b

6 b

7 c

8 a

9 b

10 c

Použitá literatura

- [1] Kolektiv autorů: Dílenská příručka - Elektronika I., Vydalo SOUE a U. Plzeň
- [2] Miroslav Bezděk: Elektronika, Kopp 2002, ISBN 80-7232-171-4
- [3] Ivan Uhlíř a kol.: Elektronické obvody a Elektronika, ČVUT 2002 ISBN 80-01-02466-0
- [4] Jan Kesi: Elektronika I-analogová technika, BEN 2007, ISBN 80-7300-074-1
- [5] J. Stach a kolektiv: Československé integrované obvody, 2. vydání, SNTL, Praha 1975, 04-513-75
- [6] Jan Kesi: Elektronika III – Číslicová technika, BEN – Technická literatura, Praha 2003, ISBN 80-7300-075-X
- [7] M. Antořová, V. Davídek: Číslicová technika (učebnice), KOPP, České Budějovice 2003, ISBN 80-7232-206-0
- [8] V. Malina: Digitální technika, KOPP, České Budějovice 2002, ISBN 80-7232-157-9
- [9] Ing. M. Dosedla: Architektura počítačů, PF Masarykovy univerzity 2007
- [10] R. Fojtík: Architektura počítačů, Universita Ostraviensis, Ostrava 2006
- [11] Kolektiv autorů: Dílenská příručka III: Elektronika I, SOUE a U Plzeň, 2004
- [12] V. Malina: Poznáváme elektroniku VIII. – KOPP České Budějovice 2006, ISBN 80-7232-271-0
- [13] Ing. P. Olivka: Architektura počítačů, katedra informatiky FEI VŠB-TU Ostrava, Ostrava 2010, <http://poli.cs.vsb.cz/edu/arp/down/archpoc.pdf>
- [14] Stanislav Křeček a kol.: Příručka zabezpečovací techniky, Blatná, 2003
- [15] ČSN EN 50 131 – 1/Z1 Poplachové systémy – Elektrické zabezpečovací systémy – Všeobecné požadavky – Změn Z1
- [16] Výukové materiály a montážní listy firmy Jablotron
- [17] Microchip: Katalogový list PIC16F84A: <http://ww1.microchip.com/downloads/en/DeviceDoc/35007b.pdf>
- [18] V. Čebiš , J. Kadlecová: Překlad originální dokumentace PIC16F84A, 1998
- [19] Doveda Boys: Stránky procesorů PIC: <http://www.cmail.cz/doveda/>

- [20] ASIX: PIC krok za krokem, komentované příklady programů pro PIC
- [21] Jazyk C – přednáška: <http://homel.vsb.cz/~rep75/proglang/prednasky/prednaska1.htm>
- [22] Metodika výuky mikroprocesorů PIC v oborech elektro, výpočetní a automatizační technika na SPŠ a SOŠ s využitím PC: <http://www.spshk.cz>
- [23] Herbert Schildt: Nauč se sám C, Praha: nakladatelství SFTPRESS, 2001, ISBN 80-86497-16-X
- [24] Dalibor Kačmář: Jazyk C: Učebnice pro střední školy, Praha: nakladatelství CP Books, a.s. 2000, ISBN 80-7226-295-5
- [25] MikroElektronika: <http://www.mikroe.com/>
- [26] Václav Kadlec: Učíme se programovat v jazyce C, Brno: nakladatelství CP Books, a.s., 2005, ISBN 80-7226-715-9
- [27] Burkhard Mann: C pro mikrokontroléry, Praha: nakladatelství Ben – technická literatura, 2000, ISBN 3-7723-4154-3
- [28] Jiří Hrbáček: Komunikace mikrokontroléru s okolím, Praha: nakladatelství Ben – technická literatura, 1999, ISBN 80-86056-42-2
- [29] Jiří Hrbáček: Moderní učebnice programování jednočipových mikrokontrolérů PIC 1, BEN 2004, ISBN 80-7300-136-5
- [30] Jiří Hrbáček: Moderní učebnice programování jednočipových mikrokontrolérů PIC 2, BEN 2007, ISBN 80-7300-137-3
- [31] Elektronika E-ZIN: <http://elektronika.ezin.cz/>
- [32] Wikipedia: 1-Wire: <http://en.wikipedia.org/wiki/1-Wire>
- [33] Václav Vacek: Učebnice programování PIC, BEN 2000, ISBN 80-86056-87-2

5 Přílohy

5.1 Zásady BOZP

5.1.1 Obecné zásady bezpečnosti práce

Při práci používáme pracovní obuv a oděv. Při práci na točivých strojích musí být vlasy zajištěny proti vtažení do stroje a oděv musí být zapnutý.

Při práci s chemickou látkou je nutné věnovat pozornost etiketě. Pro odlití použít jenom předepsané nádoby, každou chemikálii označit. Nikdy nepoužívat nádoby od požívatin.

Použité chemikálie nevylévat do odpadu, ale do k tomu určených sběrných nádob. Obdobně zacházet i ostatním odpadem.

Na pracovišti je zakázáno jíst a kouřit. Nápoje je možné mít v uzavřených lahvích.

Je zakázáno zasahovat do elektrických rozvodů. V případě zjištění poškození rozvodu nebo přívodu nahlásit věc vyučujícímu nebo vedoucímu pracovníkovi a postupovat podle jeho pokynů.

5.1.2 Zásady ochrany před antistatickou elektřinou

Při práci na elektronických zařízeních postupovat tak, aby nedocházelo ke vzniku elektrického náboje nebo byl jeho vznik minimalizován. Sledovat označení součástek, které tento jev může poškodit.

Je-li to možné, pracovat na pracovišti, které je proti vzniku elektrostatického náboje speciálně vybaveno. Je nutno dodržovat specifické postupy pro takto vybavená pracoviště.

5.1.3 Obecné zásady bezpečnosti práce na školním pracovišti

Práce probíhá pouze na pracovištích k tomu určených a činnosti se zde provádějí dle pokynů vyučujících.

Na pracovišti může pracovat pouze určený počet žáků pod vedením učitele odborného výcviku, který vykonává dozor.

Žáci jsou seznámeni s náplní práce, zařízením pracoviště a s možnými riziky úrazu.

Při zapojování se nezabýváme jinou činností, než příslušnou prací, která souvisí se zapojováním, zapojování přístrojů se provádí zásadně ve vypnutém stavu, tj. bez napětí, nedotýkáme se otáčející hřídele elektromotoru vrtačky.

Každá elektrotechnická dílna, která je určena pro výuku žáků, musí být vybavena ochranou před úrazem elektrickým proudem - ochrana pospojováním, použití proudového chrániče. Z bezpečnostních důvodů jsou dílny vybaveny tlačítkem CENTRAL STOP, kterým lze v případě potřeby vypnout všechna pracoviště současně.

5.1.4 Povinnosti žáků

- řídit se pokyny vyučujícího učitele odborného výcviku
- vstupovat pouze do učeben vyčleněných k činnosti
- při práci žáci nesmí mít volně vlající oděv
- je zakázáno v průběhu pracovních činností používat nevhodné a nebezpečné ozdobné předměty jako jsou např. kovové náramky, náušnice, ozdobné kroužky, náhrdelníky
- dbát pokynů učitele odborného výcviku a neopouštět pracoviště bez jeho vědomí
- nepoužívat mobilní telefony a jinou audiovizuální techniku
- používat pracovní obuv a oděv
- práce pod napětím je zakázána
- jakýkoli úraz nebo nevolnost ihned ohlásit učiteli
- chovat se tak abych nezpůsobil úraz sobě ani druhému
- nepoužívat poškozené nářadí, udržovat pořádek a čistotu na pracovišti
- v případě nějaké závady je nutno vše hlásit učiteli odborného výcviku nebo dozoru
- po skončení úlohy oznámí žák učiteli splnění zadaného úkolu

5.1.5 Obecné zásady požární bezpečnosti

Každý je povinen si počínat tak, aby nezavdal příčinu ke vzniku požáru, neohrozil život a zdraví osob a majetek. V případě požáru provést nutná opatření pro záchranu ohrožených osob,

uhasit požár, jestliže je to možné nebo provést nutná opatření k zamezení jeho šíření a ohlásit jej neodkladně učiteli a hasičskému záchrannému sboru na telefonní číslo 150.

5.1.6 První pomoc při úrazu elektrickým proudem

1. vypni přívod elektrického proudu
2. okamžitě poskytni první pomoc – nepřímou masáž s a dýchání z úst do úst
3. přivolej lékařskou pomoc a svého nadřízeného pracovníka

5.2 VF obvody

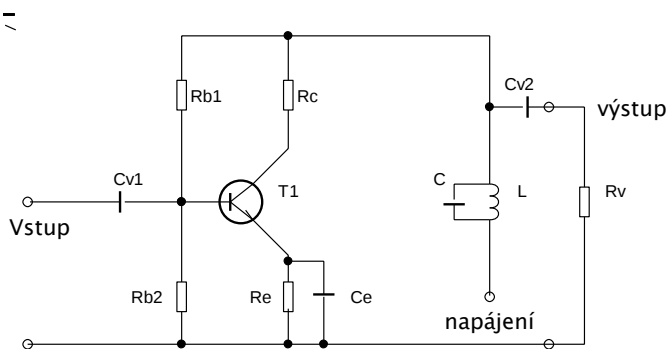
5.2.1 Vysokofrekvenční zesilovače

Vysokofrekvenční zesilovače zesilují elektrické signály jen v určitém, přesně stanoveném kmitočtovém pásmu. Signály o kmitočtu mimo toto pásmo jsou potlačeny. Ve vysokofrekvenčních zesilovačích se využívá vlastností paralelního rezonančního obvodu.

5.2.2 Vysokofrekvenční zesilovač s jednoduchým laděným obvodem

Paralelní rezonanční obvod LC na obr. 3.1 tvoří selektivní zátěž naladěnou na střední frekvenci zesilovaného frekvenčního pásma. Stejnoseměrný pracovní bod je nastaven rezistory Rb1, Rb2, a Re. Rv je vstupní odpor následujícího obvodu.

Impedance laděného obvodu je frekvenčně závislá, ve stavu rezonance nabývá maxima a má reálný charakter. Zesílení zesilovače určuje jednak zesílení tranzistoru T1, ale i rezonanční impedance laděného obvodu LC. Činitel jakosti tohoto obvodu určuje šířku přenášeného pásma.

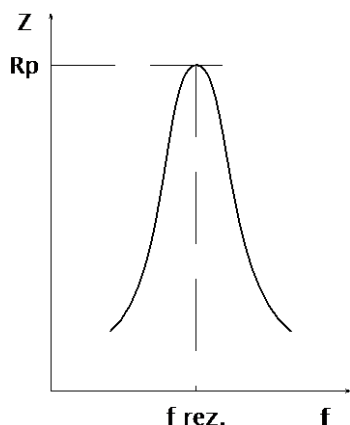


Obr. 3.1 Vysokofrekvenční zesilovač

Zesílení je největší ve středu přenášeného pásma, tj. pro rezonanční frekvenci obr. 3.2, jejíž velikost lze spočítat pomocí Thomsonova vzorce

$$f = \frac{1}{2\pi\sqrt{LC}}$$

Pro nižší a vyšší frekvence než je f_{rez} zesílení klesá. Šířka pásma je dána poklesem zesílení o -3 dB na každou stranu od rezonanční frekvence.



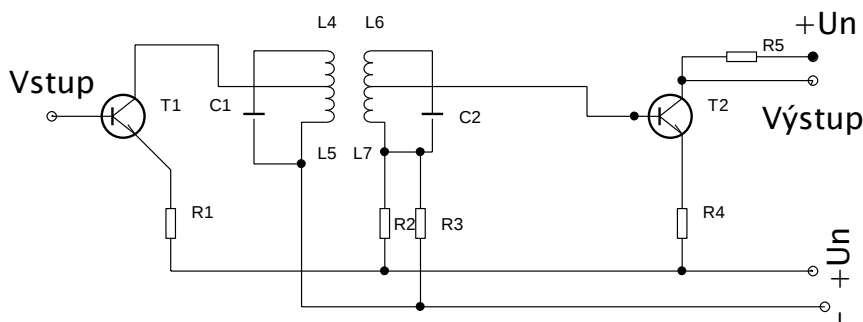
Obr. 3.2 Rezonanční křivka paralelního rezonančního obvodu

Popsaný zesilovač se liší od ideálního stavu. Při velkém činiteli jakosti je malá šířka přenášeného pásma a při malém činiteli jakosti má nevyhovující selektivitu.

5.2.3 Vázané rezonanční obvody

Z důvodu požadavku jakostnějších vysokofrekvenčních zesilovačů používají zesilovače s vázanými rezonančními obvody, obr. 3.3.

Je to vysokofrekvenční zesilovač s větší šířkou pásma. Přenos energie ze vstupního do výstupního obvodu určuje činitel vazby mezi rezonančními obvody, které lze nastavit vzájemnou indukčností M . Stupeň vazby upravuje tvar rezonanční křivky na tzv. vazbu kritickou, nadkritickou a podkritickou. Největší šířku pásma má vazba nadkritická. Vlastnosti zesilovače jsou dány zesílením tranzistorů, ale i vázanými laděnými obvody. Pomocí odboček na vinutí laděného obvodu se zmenšuje tlumící vliv tranzistoru na laděný obvod.



Obr. 3.3 V_f zesilovač s vázaným rezonančním obvodem

5.3.4 Širokopásmové zesilovače

Širokopásmové zesilovače jsou takové, u kterých je šířka přenášeného pásma zesilovaných frekvencí větší než u běžných zesilovačů. Dělí se na zesilovače s nosnou frekvencí a zesilovače bez nosné frekvence. Požívají se v měřicí technice a případně jako obrazové zesilovače v televizní technice.

5.4 Spínací, impulsní a tvarovací obvody

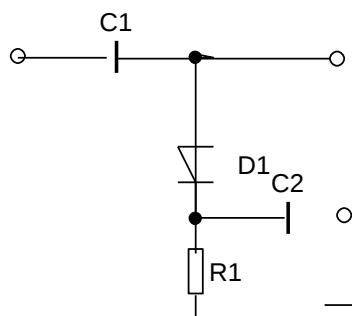
5.4.1 Spínací obvody

Spínací obvod se užívá ke spínání signálu, ve dvou stavech 0 a 1, neboli zapnuto, vypnuto. Rostoucí nároky na rychlost a spolehlivost různých ovládacích a měřicích zařízení nutí konstruktéry nahrazovat mechanické spínače elektronickými spínači. Dalším hlediskem, ze kterého je možno nahlížet na spínací prvky je jejich tepelná ztráta. Při zmíněném dvoustavovém provozu je právě jejich tepelná ztráta minimální, což například opodstatňuje použití obvodově složitějších napájecích zdrojů, či regulátorů výkonu.

5.4.2 Spínací diody

Spínací diody se používají v obvodech pro spínání elektrických signálů. Pracují obvykle se signály obdélníkového průběhu a přecházejí ze stavu vodivého do nevodivého. Použitelnost diod pro spínání je omezena setrvačností diody při přepnutí z vodivého do nevodivého stavu. Příčinou je nahromadění menšinových nosičů v blízkosti přechodu PN.

Chceme-li pomocí diody spínat střídavé signály, musíme prostřednictvím stejnosměrného napětí posunout pracovní bod do lineární oblasti propustné části charakteristiky. Jedná se tedy o spínání signálů malých úrovní.

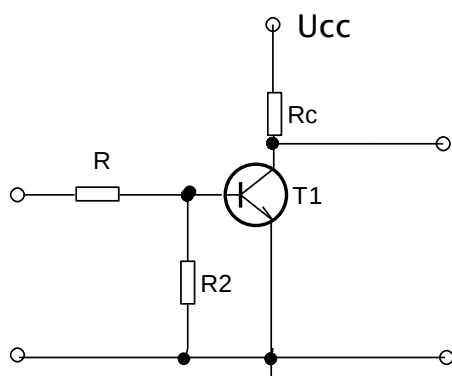


Obr. 4.2 Dioda jako spínač

5.4.3 Spínací tranzistor

Provozujeme-li tranzistor jako spínač, tak podle velikosti signálu v bázi může tranzistorem procházet buďto poměrně velký kolektorový proud, nebo proud tranzistorem téměř neprochází.

Tyto dva stavy - propustný nebo blokovací - daly tranzistoru při tomto použití název spínač. V zapnutém stavu tedy představuje tranzistor pro kolektorový proud malý odpor, ve vypnutém stavu pak velmi velký odpor. Tranzistor se tedy chová jako spínač. Protože nic není dokonalé, protéká v zavřeném stavu proud I_{co} a v otevřeném stavu je na tranzistoru úbytek napětí U_{ce-sat} . Dále je nutná vysoká spínací rychlost tranzistoru, spínací časy se pohybují v řádu nanosekund. Tepelné ztráty závisí na četnosti spínání, poměru sepnuto - vypnuto a velikosti zátěže a je nutno je respektovat. Pro tuto aplikaci je obvyklé zapojení se společným emitorem, které je uvedeno na obr 4.2. Rezistor R_c v kolektoru představuje spínanou zátěž, rezistor R_2 zajišťuje uzavření tranzistoru.



Obr. 4.2 Tranzistor jako spínač

5.4.4 Tvarovací a impulsní obvody

Tvarovací obvody se používají, jak už jejich název napovídá, pro změnu tvaru elektrického signálu. Jednak se jedná o tvarování z obecných signálů na signály pravoúhlé, např. zkreslení dlouhým vedením, odskoky kontaktů, analogově – číslicové převodníky atd., které se pak dále mohou zpracovávat číslicovou technikou. Na druhé straně ovšem i signály trojúhelníkové potřebujeme převést na sinusový tvar, který pak dále zpracováváme obvody analogovými.

Tvarovací obvody se dělí na dvě základní skupiny podle použití součástek:

- s pasivními součástkami
- s aktivními součástkami

Protože vstupy a výstupy těchto obvodů tvoří impulsy, nebo jejich sledy, je nutno nejprve tento pojem definovat:

Impuls je definován jako jednorázový jev konečné délky. Je definován svým tvarem (pravoúhlý, jehlový, trojúhelníkový atd.), amplitudou, dobou trvání a okamžikem vzniku. Vydeme-li z reálného tvaru impulsu, lze stanovit následující důležité pojmy.

Velikost impulsu – můžeme definovat hodnotu střední, efektivní, okamžitou, zpravidla však udáváme hodnotu maximální, kterou nazýváme amplitudou.

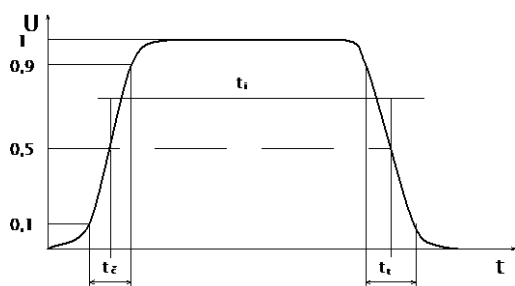
Doba trvání impulsu – (aktivní délka) je rozpětí mezi 50 % hodnotou přední a zadní hrany amplitudy impulsu (čela a týlu). Náběh a doběh je časové rozpětí mezi 10% a 90% amplitudy.

Sled impulsů – pravidelně se opakující impulsy.

Frekvence – počet impulsů za sekundu.

Perioda – doba daná součtem doby trvání impulsu a mezery mezi nimi.

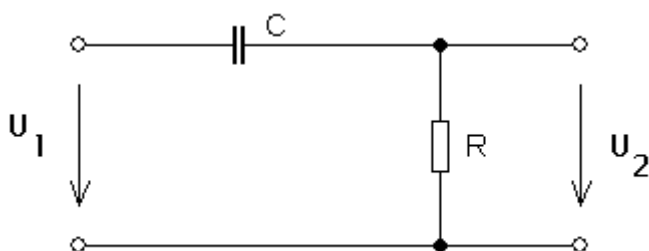
Střída – poměr doby trvání impulsu k délce periody.



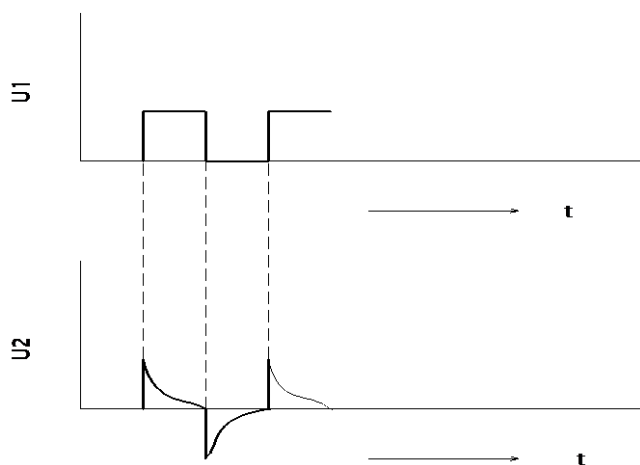
Obr. 4.3 Tvar impulsu

5.4.5 Tvarování impulsu pomocí pasivních součástek

Derivační článek používají k vytvoření derivačního impulsu, v číslicové technice k řízení dvouhodnotových spínacích systémů, jestliže má převládat dynamická složka řízení nad statickou. Pokud statickou složku nechceme zcela potlačit, kombinujeme derivační člen s děličem napětí. Popsané derivační chování členu RC platí nejen pro pravoúhlé napěťové impulsy, nýbrž i pro jiné tvary signálu, např. pro sinusové napětí. Z hlediska tvarování se jedná o změnu z pravoúhlého průběhu na exponenciální.



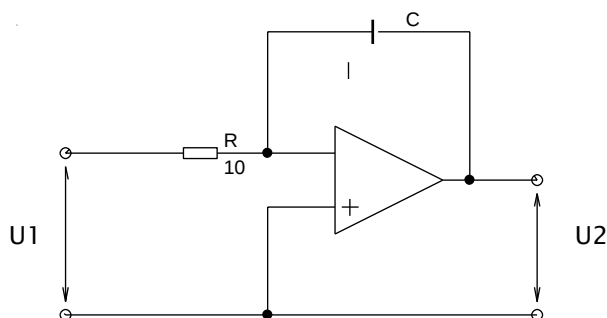
Obr. 4.4 Derivační článek



Obr. 4.5 Průběh výstupního napětí v závislosti na vstupním pulsu

5.4.6 Tvarování impulsu pomocí aktivních součástek

Nedostatky použití pasivních součástek pro tvarování impulsů odstraňuje použití aktivních součástek – operačních zesilovačů, které optimalizují požadované průběhy napětí.



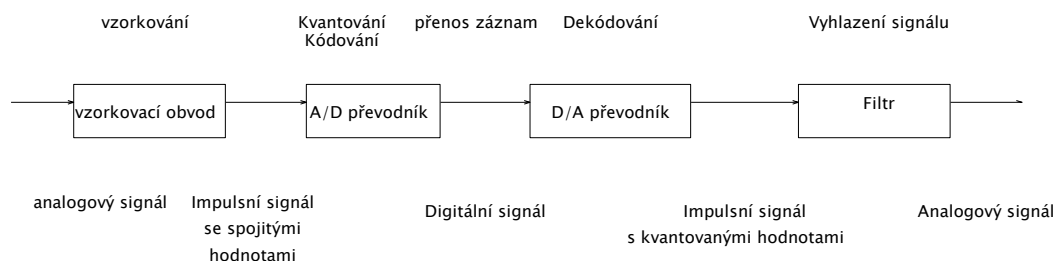
Obr. 4.6 Tvarovací obvod s operačním zesilovačem

Uvedený zpětnovazební obvod je poměrně jednoduchý. Obzvláště je toto tvarování výhodné pro velmi nízké kmitočty, protože zejména LC prvky vycházejí pro nižší kmitočty neúnosně veliké. Další častou úlohou je vytvarování v podstatě jakéhokoli průběhu na pravoúhlý. Pro řešení této úlohy se používají obvody, které spínají na určitou úroveň vstupního napětí. Jejich typickým představitelem je Schmidův klopný obvod, který je tvořen dvěma tranzistory se společným emitorovým odporem. Právě tato vazba umožňuje rychlé překlopení obvodu z jednoho stavu do druhého. V praxi se používá pro tvarování impulsů nebo pro vytváření impulsového průběhu z průběhu měnicího se spojitě. Obvod překlápí v okamžiku, v němž vstupní signál převyší určitou napěťovou úroveň. Praktická realizace je uvedena v kapitole číslicová technika.

5.5 Převodníky A/D a D/A

5.5.1 Převodníky

Analogově-číslicové (dále jen A/D) a číslicově-analogové (dále jen D/A) převodníky (také ADC a DAC) nacházejí uplatnění všude tam, kde je třeba analogový signál číslicově zpracovat nebo analogový signál z číslicového vytvořit. Je třeba si uvědomit, že většina veličin má analogový charakter (teplota, tlak, pohyb, lidský hlas apod.). Na druhou stranu má však číslicové zpracování analogových signálů řadu výhod, které jsou podpořeny ještě relativní dostupností a nízkou cenou technického vybavení pro zpracování číslicových signálů, tj. logických kombinačních a sekvenčních obvodů, mikroprocesorů, pamětí aj. Oba druhy převodníků mohou být realizovány buďto výhradně technickými nebo kombinací technických a programových prostředků.



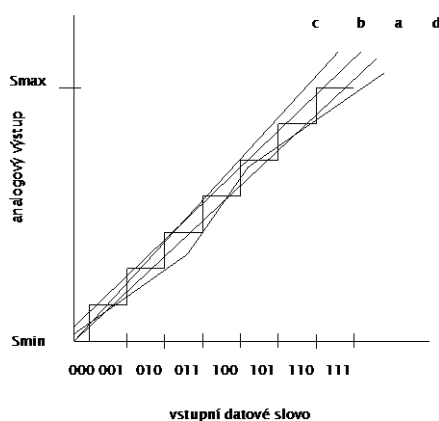
Obr. 2.1 Funkce převodníků

5.5.2 Funkce D/A převodníků

D/A převodníky zajišťují převod vstupní číslicové informace (datového slova) na výstupní analogový signál, obvykle na odpovídající hodnotu elektrického napětí (méně často elektrického proudu). Na výstupu převodníku však nemůžeme nastavit libovolnou hodnotu analogového signálu, výstupní signál je schodovitý, jeho hodnoty mohou nabývat pouze diskrétních hodnot. Chyba způsobená diskrétními úrovněmi výstupního signálu se nazývá **kvantizační chyba**. Maximální nepřesnost, tj. rozdíl mezi požadovanou a nastavenou hodnotou výstupního signálu je dán polovinou přírůstku výstupního signálu, odpovídajícímu nejnižšímu bitu vstupního datového slova (LSB). Dalším důležitým parametrem je **rozlišovací schopnost**

(kvantizační krok) Q . Je vyjadřována počtem diskretních stupňů výstupního analogového signálu a je v přímé souvislosti s počtem bitů vstupního datového slova.

Základní aplikací D/A převodníků ve spojení s mikroprocesorem, resp. počítačem je generování různých hodnot, popř. různých průběhů výstupního napětí. Toto napětí může být buď přímo použito pro řízení připojených akčních členů, nebo může sloužit jako vstupní napětí pro připojený převodník napětí na jinou elektrickou nebo neelektrickou veličinu.



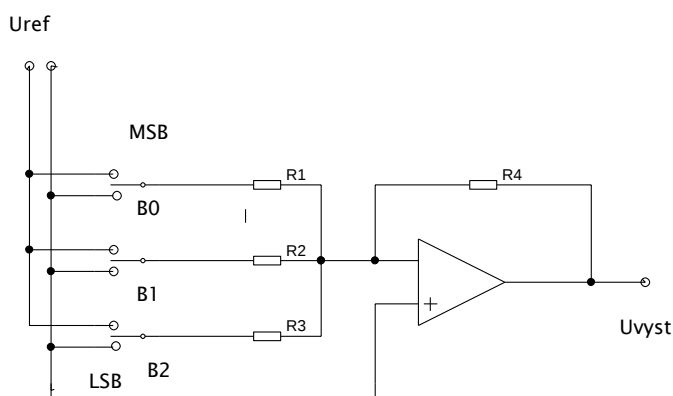
Obr. 2.2 Převodní charakteristika převodníku

5.5.3 Typy D/A převodníků

Typy D/A převodníků nejčastěji dělíme na **přímé** a **nepřímé**. U přímých převodníků je vstupní datové slovo přímo převedeno na výstupní napětí, příp. proud. Tyto převodníky jsou nejčastěji řešeny pomocí příčkové nebo váhové struktury odporové sítě. Základní charakteristikou nepřímých D/A převodníků je rozdělení převodu na dvě části – v první části je číslicová veličina převedena na pomocný diskretní signál (na šířku pulsu nebo počet pulsů za jednotku času), který je ve druhé části převeden na výstupní analogový signál.

5.5.4 princip D/A převodníku váhovou strukturou odporové sítě

Princip D/A převodníku váhovou strukturou odporové sítě je znázorněn na obr. 2.3. Řídicí veličinou je vstupní datové slovo, předávané prostřednictvím datové sběrnice. Jednotlivé bity vstupního slova s váhou $2_0, 2_1, 2_2, \dots, 2_n$ ovládají jednotlivé elektrické přepínače, které mají v sérii odpory o takových hodnotách, že každá další hodnota je vždy dvojnásobkem předcházející. V případě obr.2, kde je ukázána realizace 3bitového D/A převodníku, mají zmiňované odpory hodnoty $4R, 2R$ a R (počínaje přepínačem s nejnižší vahou (odpovídající bitu B_2)). Při spínání jednotlivých přepínačů se mění celkový přenos výstupního zesilovače (princip součtového zesilovače). Při realizaci jsou kladeny velké nároky na přesnost odporů.



Obr. 2.3 Tříbitový D/a Převodník s váhovou odporovou sítí

5.5.5 Vlastnosti A/D převodníků

A/D převodníky zajišťují převod vstupního analogového signálu na výstupní číslicový signál. Vstupním signálem je nejčastěji napětí, výstupem pak datové slovo o stanoveném počtu bitů. Převod spojitého analogového signálu na diskrétní číslicový tvar se u většiny převodníků provádí ve dvou krocích. Analogový se nejprve periodicky **vzorkuje**, tj. získává se sled úzkých impulsů, jejichž amplitudy odpovídají analogovému signálu v příslušných časových okamžicích. Ve druhém kroku jsou amplitudy jednotlivých signálů převáděny tzv. **kvantováním** na číslicový tvar. Kvantizace přiřazuje jednotlivým vzorkům diskrétní hodnotu (výstupní datové slovo). Přesnost převodu není teoreticky omezena, ale musíme dodržet následující zásady:

- vzorkování analogového signálu je prováděno alespoň s dvojnásobným opakovacím kmitočtem než je nejvyšší harmonická složka snímaného analogového napětí (Nyquistův teorém),

- vzorkovací impulsy jsou dostatečně úzké,

- kvantování vzorkovacích impulsů je dostatečně „jemné“, tj. číslo vyjadřující amplitudu má dostatečný počet řádů, tedy výstupní datové slovo má dostatečný počet bitů.

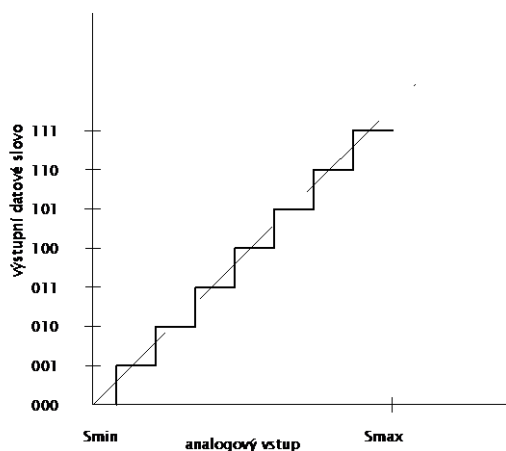
Uvedené procesy mají své technické možnosti. Rychlost změny analogového signálu, kterou je možno zachytit a digitalizovat, má své meze v rychlosti a rozlišovací schopnosti kvantovacího obvodu A/D převodníku.

Rychlost vzorkování vstupního signálu patří mezi nejvýznamnější parametry A/D převodníků. Musí být dostatečně vysoká vzhledem k nejvyšší kmitočtové složce vstupního analogového napětí – jak již bylo uvedeno, je nutné přenést více jak dva body amplitudy nejvyšší kmitočtové složky sledovaného signálu.

Pokud nás naopak některá vyšší harmonická složka nezajímá nebo způsobuje chybu v následném zpracování dat (např. šumový signál), je možno ji odstranit vhodnou dolní propustí.

Rychlost převodu je u A/D převodníků obvykle shodná s rychlostí vzorkování, resp. naopak, rychlost vzorkování vyplývá z nejkratší možné doby převodu. **Doba převodu** může být určena jako doba, která uplyne od okamžiku přivedení vstupního analogového napětí na vstup převodníku až do doby, kdy je výstupu převodníku k dispozici platné výstupní datové slovo. Může být rovněž vyjádřena počtem úplných převodů za jednotku času nebo počtem bitů

za jednotku času. Obdobně jako u D/A převodníku je definována **rozlišovací schopnost** – je určena počtem úrovní, na něž je rozdělen rozsah dovoleného vstupního napětí. Je třeba si uvědomit, že čím je větší rozlišovací schopnost, tím je nižší rychlost převodu.



Obr. 2.4 Převodní charakteristika A/D převodníku

Rozlišovací schopnost se současně rovná **kvantizačnímu kroku Q** . Chybu vzniklou kvantováním shodně nazýváme (jako u D/A převodníků) **kvantizační chyba**. Může dosahovat maximální hodnoty rovné polovině hodnoty změny napětí, která by vyvolala změnu výstupního datového slova o jeden bit nejnižšího řádu (LSB). Maximální chyba převodu je tedy u ideálního převodníku $\pm Q/2$. Na obr.2.4 je znázorněna ideální převodní charakteristika A/D převodníku. Změny převodní charakteristiky od ideálního tvaru jsou dalším zdrojem chyb. Tyto chyby jsou analogické chybám, o kterých jsme se zmiňovali u D/A převodníků (chyby způsobené napěťovým posunem, změnou zisku převodníku a nelinearitou převodníku). Schodovitý průběh převodní charakteristiky způsobuje odchylku od ideálního průběhu.

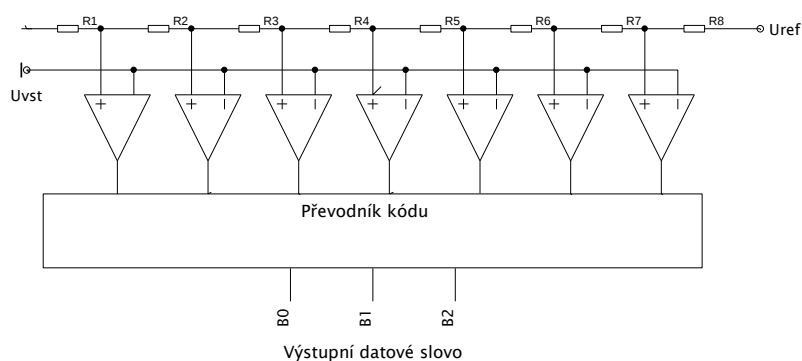
5.5.6 Typy A/D převodníků

A/D převodníky můžeme dělit podle různých kritérií. Podle způsobu činnosti dělíme převodníky na synchronní a asynchronní. U **synchronních převodníků** probíhá převod analogového napětí na výstupní datové slovo v určitém počtu kroků, které se uskutečňují synchronně s hodinovými (taktovacími) impulsy, u **asynchronních převodníků** může být převod rovněž uskutečněn v několika krocích, ovšem doba trvání těchto kroků závisí výhradně na časové odezvě dílčích obvodů převodníku a na jejich zpoždění.

Jiné rozdělení A/D převodníků je podle vstupního signálu na přímé a nepřímé. **Přímé převodníky** převádějí přímo vstupní analogové napětí na výstupní slovo, u **nepřímých**

převodníků se vstupní analogové napětí nejprve převádí určitým obvodem na jinou analogovou veličinu (např. na dobu trvání impulsu) a dalším obvodem je teprve tato veličina převedena na výstupní datové slovo.

Paralelní A/D převodník je nejrychlejší a současně principiálně nejjednodušším typem přímého A/D převodníku. Princip převodníku je znázorněn na obr. 2.5. Vstupní analogové napětí je přiváděno současně na vstupy soustavy napět'ových komparátorů. Na těchto komparátorech se toto napětí porovnává s určitým referenčním napětím (pro každý komparátor rozdílným, daným odporovým děličem) a výstup jednotlivých komparátorů překlápí v případě, že převaděč kódu pak převede výstupy z napět'ových komparátorů na výstupní datové slovo. Doba převodu paralelního převodníku je určena přenosovým zpožděním, resp. dobou ustálení napět'ových komparátorů a přenosovým zpožděním v převodníku kódu.



Obr. 2.5 Tříbitový paralelní převodník

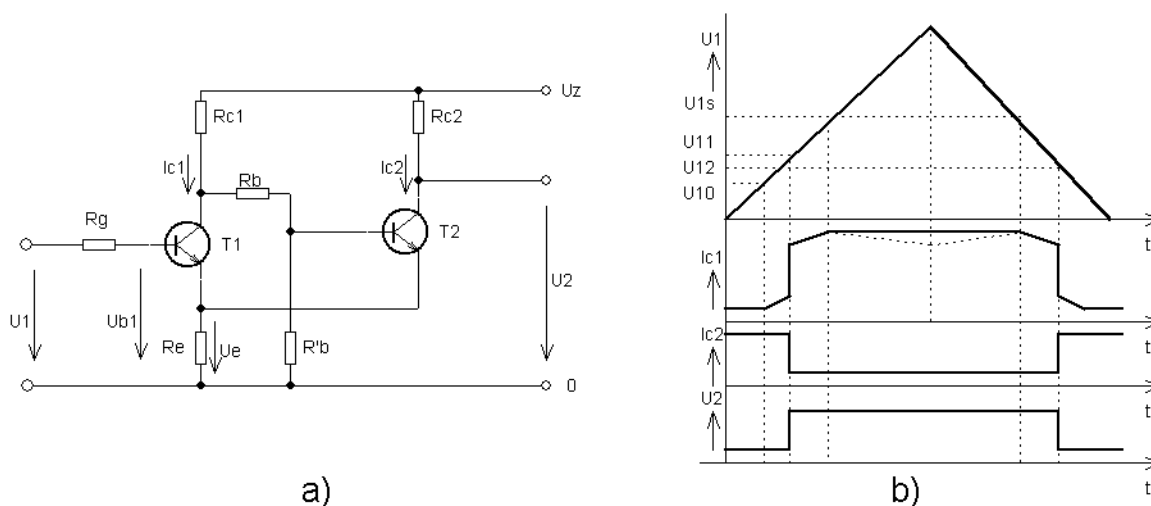
5.6 Tvarovací obvody

5.6.1 Schmittův klopný obvod

Schmittův klopný obvod se používá v celé řadě konstrukcí, a to nejen v číslicové technice. Určitě se s ním setkáme i v zapojeních s operačními zesilovači.

V číslicové technice slouží Schmittův klopný obvod k úpravě tvaru impulsů přiváděných na vstupy logických členů. Připomeňme jen, že signál přiváděný na vstupy číslicových obvodů musí mít pravoúhlý tvar se strmým čelem a týlem. Tuto podmínku přiváděné signály mnohdy nesplňují, proto je nutné signály upravit.

Impuls se na výstupu Schmittova klopného obvodu objeví tehdy, pokud vstupní signál převyší určitou napěťovou úroveň.



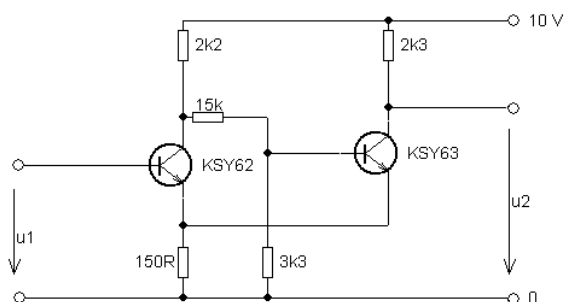
Obr. 1: Schmittův klopný obvod

Příklad obvodu pro úpravu signálu najdeme na obr. č. 1a. Předpokládejme nejprve, že odpor R_g je malý a napětí U_{b1} sleduje průběh napětí U_1 . Vyjdeme-li ze stavu, v němž napětí U_{b1} je nulové, je tranzistor T_1 uzavřen a T_2 vede, protože na jeho bázi je přiváděno napětí děličem $R_b R'_b$ z kolektoru tranzistoru T_1 . Popsaná situace se nezmění, pokud je napětí U_{b1} nižší než napětí U_e na rezistoru R_e . Překročí-li napětí U_{b1} velikost napětí U_e , začne se tranzistor T_1 otevírat, napětí na jeho kolektoru klesne a tím se zmenší buzení tranzistoru T_2 . Při dalším růstu napětí U_{b1} se tranzistor T_1 otevře T_2 naopak uzavře, napětí na výstupu U_2 bude maximální (objeví se log. 1). K zpětnému překlopení dojde, jestliže napětí na vstupu

poklesne natolik, že napětí na kolektoru tranzistoru T1 vzroste na velikost, při které dělič $R_b R'_b$ začne otevírat tranzistor T2.

Na obr. č. 1b je pro objasnění nakreslen časový průběh kolektorového proudu tranzistoru T₁, tranzistoru T₂ a průběh na kolektoru T₂ pro případ, že na vstup je přivedeno napětí s trojúhelníkovým časovým průběhem. Při napětí U_{10} se začíná otevírat tranzistor T₁, při napětí U_{11} se dostává T₁ do nasycení. Ke zpětnému překlopení dochází při vstupním napětí U_{12} . Rozdíl mezi U_{11} a U_{12} se nazývá hystereze obvodu.

Příklad navrženého Schmittova klopného obvodu je na obr. 2.



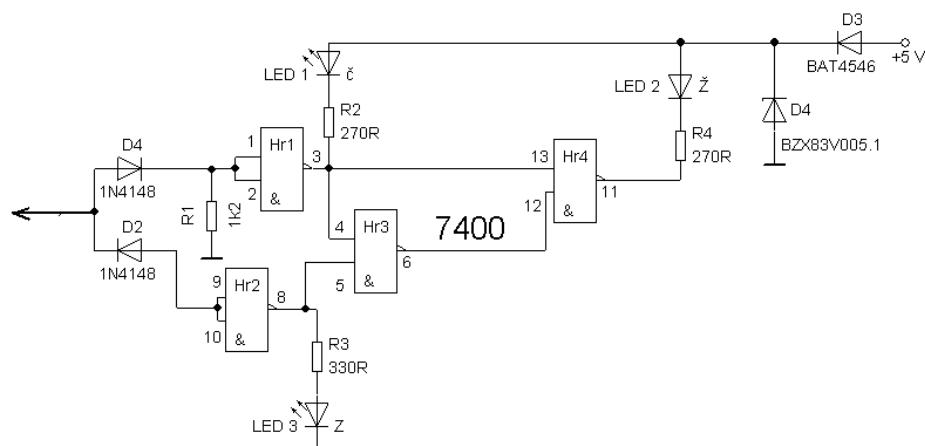
Obr. 2: Navržený Schmittův klopný obvod

5.6.2 Měření, oprava a diagnostika číslicových obvodů

Pro diagnostiku číslicových obvodů můžeme použít stejnosměrný voltmetr, kdy v obvodu měříme napěťové úrovně, tj. napětí menší než 0,8 V odpovídá log. 0 (úroveň L), napětí větší než 2 V odpovídá log. 1 (úroveň H). Nikdy v číslicovém obvodu nesmí nastat tzv. zakázaný stav, tj. hodnota napětí v rozmezí 0,8 až 2 V. Hodnoty pro TTL byly popsány v kapitole 2.3.2 s názvem Základní logický člen integrovaných obvodů TTL a shrnuty jsou v tabulce 3 v téže kapitole.

Některé dražší multimetry mají v sobě zabudovanu funkci logické sondy, kdy přímo na displej vypisují logické úrovně (tedy hodnoty log. 0, log. 1, neurčité pásmo).

Speciální měřicí přístroj sloužící pro měření číslicových signálů se nazývá **logická sonda**. Jedno z možných zapojení je na obrázku č. 3. Hlavní součástí je integrovaný obvod 7400, který obsahuje čtyři hradla NAND. K zapojení potřebujeme ještě několik diod, rezistorů a tři LED diody.



Obr. 3: Schéma logické sondy

Jak logická sonda pracuje?

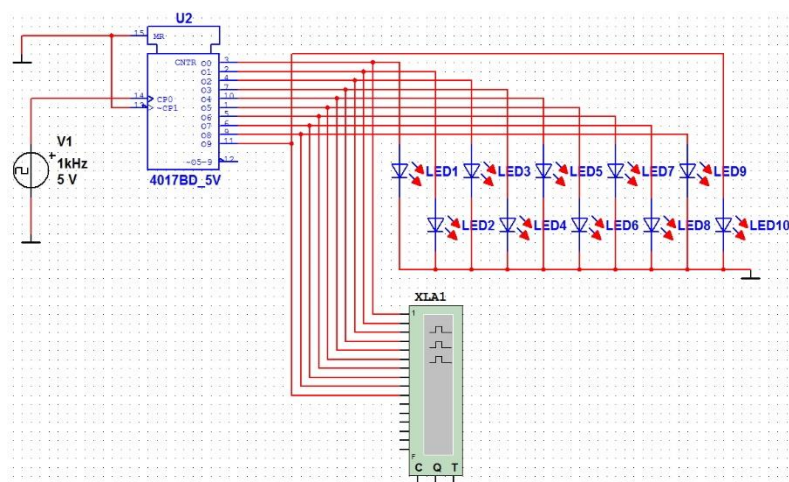
Úroveň H registruje hradlo 1. Dioda D_1 stojí v cestě signálu je pólována v propustném směru, zatímco současně vedlejší D_2 je v závěrném směru. Na výstupu tohoto hradla - invertoru je úroveň L, jestliže je vstup na H. Červená LED_1 se rozsvítí, protože je zapojená druhým vývodem proti kladnému napětí. Rezistor R_1 ovlivňuje nejnižší kladné napětí, od kterého invertor pracuje. Kdyby LED dioda částečně prosvěcovala bez připojení hrotu k úrovni H pak odpor R_1 zmenšíme.

Úroveň L registruje hradlo 2. Před ním vložená dioda D_2 je pro vstupní úroveň L pólována v propustném směru, zatímco současně D_1 v závěrném směru. V takovém případě je výstup invertoru na H. Tuto úroveň zobrazuje zelená LED_3 , zapojená proti zemi. Rezistor R_3 omezuje proud tekoucí světelnou diodou.

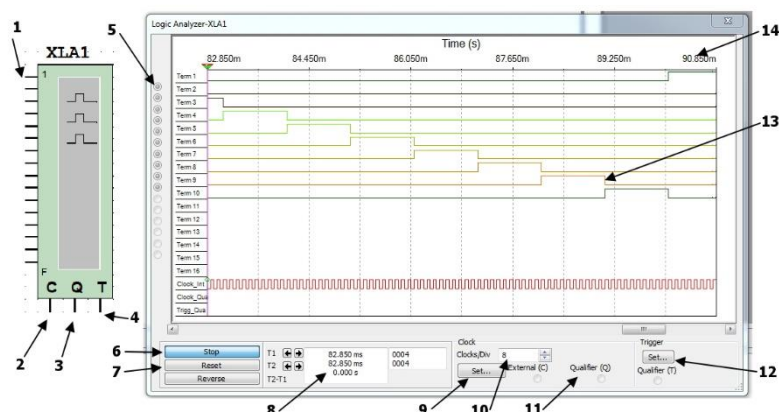
Zakázané pásmo zobrazuje žlutá LED_2 . Kdy se rozsvítí? Na vstupy hradla 3 (NAND) přichází informace z invertorů 1 a 2. Není-li hrot sondy ve styku s úrovní H nebo L, pak vstup hradla (pin č. 4) připojený k výstupu druhého invertoru je na úrovni L. Naproti tomu z výstupu prvního invertoru přichází současně úroveň H, protože jeho vstup je přes R_1 uzemněn. Výstup hradla 3 má proto úroveň H. A jak se zachová hradlo 4 (NAND)? Za této situace je na obou vstupech úroveň H, a proto na výstupu zjišťujeme L. LED_2 zapojená odtud proti kladnému napětí se rozsvítí. Přestane svítit okamžitě, jakmile se změní některá ze vstupních úrovní hradla 4. Jinak řečeno zhasne, jestliže hrot přiložíme k místu s úrovní L nebo H. Rezistor R_4 omezuje proud žluté LED_2 , stejně jako R_2 omezuje proud červené LED_1 . Změnou odporu vyrovnáváme případné rozdíly ve svítivosti jednotlivých diod.

V praxi se potom můžeme setkat s dalším zařízením, které se nazývá **logický analyzátor**. Logický analyzátor je zařízení, které má mnoho vstupů, pomocí kterých, poté co je spuštěno, si zapamatuje posloupnost logických stavů řady číslicových signálů - průběhů na výstupech jednotlivých hradel.

Logický analyzátor slouží k názorné demonstraci a měření parametrů digitálních signálů. Umožňuje rychlé sledování logických stavů společně s jejich analýzou v čase (posloupnosti). Princip měření logickým analyzátozem si ukážeme v programu NI Multisim (obr. č. 4).



Po dvojkliku na ikonu logického analyzátoru se objeví okno s průběhy číslicových signálů (obr. 5).



Obr. 5: Zapojení běžícího světla s LED diodami s obvodem CMOS 4017 pro měření log.

Význam jednotlivých ovládacích prvků:

- 1 – vstupy 16 kanálů
- 2 – vnější hodinový vstup
- 3 – filtrace hodinového vstupu
- 4 – filtrace spouštění
- 5 – indikace připojených kanálů
- 6 – zastavení simulace
- 7 – vymazání průběhu
- 8 – časové údaje pro polohu kurzorů
- 9 – nastavení hodinového vstupu
- 10 – volba počtu zobrazených period hodinového signálu (měřítko pro osu x)
- 11 – indikace připojených řídicích signálů
- 12 – nastavení spouštění
- 13 – zobrazení průběhů signálů v připojených kanálech
- 14 – časová základna

5.7 Průmyslová televize – kamerové systémy

Průmyslová televize je významnou součástí signálně informačních systémů, neboť vizuální informace je nenahraditelná všude tam, kde vyhodnocování informací a případný zásah provádí člověk. Miniaturizace a příznivý vývoj cen kamer dovolují projektovat rozsáhlé systémy o mnoha desítkách kamer. V rozsáhlých systémech jsou kladeny vysoké požadavky na centrální ústřednu, která musí umožňovat připojení velkého počtu video vstupů a výstupů a jejich nezávislé přepínání s možností řízení na více úrovních.

Přenosové cesty systému průmyslové televize jsou uzpůsobeny pro přenos nejen videosignálů, ale i kombinace audio - video. Šířka pásma je dostatečná pro černobílý i barevný TV signál.

Řízení systému průmyslové televize zahrnuje řízení přepínání ústředny vstupy - výstupy na více úrovních a řízení dalších zařízení jako jsou polohovací hlavice a zoomy kamer, videorekordéry, videosenzory a selektory. Komunikace mezi jednotlivými zařízeními je sériová, na standardních rozhraních RS232, RS485 resp. po proudových smyčkách.

Řízení místní slouží k volbě signálů pro místní kontrolní pracoviště, tvořené jedním až čtyřmi monitory. Volba požadovaného signálu se pak provádí infračerveným (IR) dálkovým ovládáním, obdobně jako při volbě TV kanálu.

Řízení centrální je prováděno z hlavního kontrolního pracoviště, je nadřazené řízení místnímu. K ovládání slouží operační ovladač nebo vizualizační systém.

Řízení z nadřazeného systému slouží k nastavení na základě podnětů, přicházejících ze signalizačního systému nezávisle na obsluze (nadřazeno místnímu i centrálnímu řízení).

Kamery jsou napájeny ze zdrojů 24 V stř., což umožňuje využít synchronizaci a zajistit tak bezchybný střih. Povel pro pohyblivé kamery jsou vedeny po sériovém rozhraní přímo z řídicí jednotky VCU.

Moderní CCD kamery s digitálním zpracováním obrazu dovolují nasazení v nejtěžších světelných podmínkách a to jak ve vnitřních prostorech s protisvětlem (okna), tak ve venkovních prostorech, zejména pak v pohyblivých kamerách s objektivy zoom.

Bezpečnostní kryty kamer, chránící nejen vlastní kameru s objektivem, ale i přívodní kabeláž, jsou zhotoveny z nerezové oceli a pevnostního skla v průzoru.

Dlouhodobý záznam obrazu pomocí videorekordéru typu Time-Lapse a krátkodobý záznam alarmové situace na běžný VHS videorekordér je díky pokroku v technologii komprese obrazu a jeho počítačovému zpracování v současnosti nahrazován digitálními videorekordéry, používajícími jako primární záznamové medium pevný disk a pro archivaci kazety DV nebo DAT.

Digitální videorekordér spojuje funkce multiplexního provozu, digitálního záznamu, videozáznamu poplachových událostí a videosenzoru s možností snadného a rychlého vyhledávání libovolných videozáznamů. Digitální videorekordér se dodává ve verzích s 9 nebo 16 vstupy s možností záznamu na kazetu DAT. Jeho obsluha je možná prostřednictvím tlačítek na předním panelu nebo pomocí myši. Největší výhodou tohoto zařízení je možnost tzv. triplexního režimu. Triplexní režim umožňuje současný záznam, prohlížení záznamu a zálohování na DAT kazetu.

Při použití dálkového řízení vyhledávání lze řídit videorekordér i ze vzdáleného pracoviště, některé typy digitálních rekordérů umožňují přímé připojení do sítí LAN nebo po doplnění o vícekanálový kamerový server umožňují připojení vzdálených pracovišť přes TCP/IP protokol.

Digitální videorekordér je více než pouhý digitální videorekordér, je to první inteligentní systém průmyslové televize pro záznam, přehrávání a analýzu obrazových dat, koncipovaný pro 21. století.

Digitální videosenzory pohybu, doplňující střežící systém, pracují na principu postupného porovnávání okamžitých výstupů jednotlivých kamer s referenčními obrazy uloženými v paměti videosenzoru. Pro každou kameru lze nastavit individuální strategii střežení tak, aby bylo možno dosáhnout optimálních výsledků i při vyhodnocování různých druhů scén. Používaný princip detekce využívá řadu nástrojů pro potlačení nežádoucích rušivých vlivů, vyskytujících se při střežení venkovních prostorů.

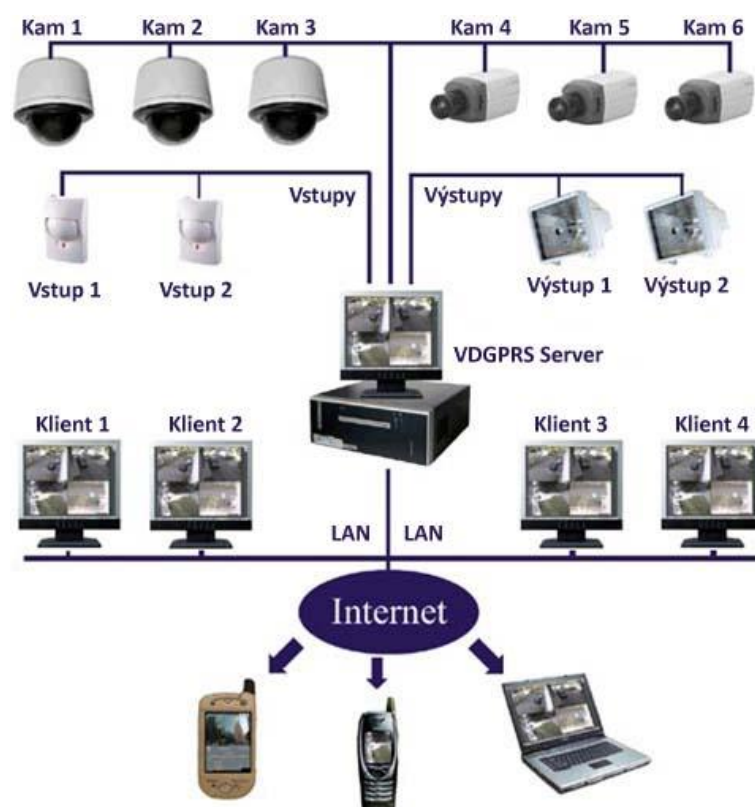


Schéma kamerového systému



Venkovní kamery

5.8 Programování PIC - IR přenos

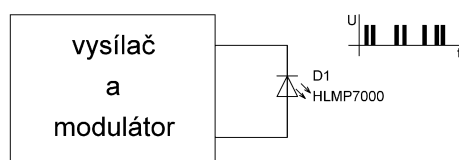
5.8.1 IR přenos

Namíříme, stiskneme tlačítko na dálkovém ovladači a televize se zapne. Jak je to možné? Dochází k IR komunikaci.

IR komunikace je optickou komunikací, která využívá neviditelné infračervené (tepelné) záření. Zdrojem záření bývá obvykle infra LED (LED vysílající v pásmu infračerveného spektra). Jas této LED diody lze snadno modulovat. Modulace jasu je nezbytná proto, aby nedocházelo k falešnému řízení přijímače jinými tepelnými zdroji umístěnými v blízkosti přijímače (žárovka, kamna, atd.). Šetří se tím také baterie dálkového ovladače, protože LED dioda nemusí svítit stále.

Zmáčknutím tlačítka se rozeběhne program mikrokontroléru v dálkovém ovladači, který podle kódu tlačítka vyšle na výstupní port signál odpovídající v daném standardu.

Kódový rámec se obvykle skládá z povelu a čísla výrobce (nebo typu zařízení). Tato identifikace umožňuje mít vedle sebe dva přístroje a např. zapnout ten první aniž by se nechtěně zároveň sepnul ten druhý.



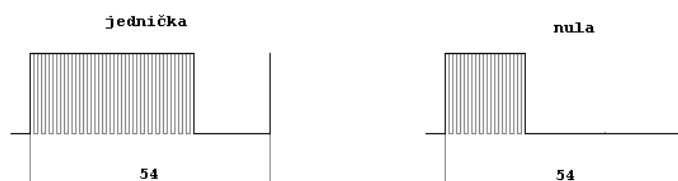
Obr. 1: Blokové schéma IR vysílače

Samotné kódování je standardizováno. Nejrozšířenější (převážně v Evropě) je standard RC5 vyvinutý firmou Phillips. Kódy vyvinuté japonskými firmami jsou např. kód NEC, kód Mitsubishi nebo kód SIRCS vyvinutý firmou Sony.

5.8.2 Zjednodušená realizace mikrokontrolérem PIC16F84A

Na vysílací straně stiskneme tlačítko a tím vyšleme kód. Na přijímací straně se platný kód zapíše na PORTB a zůstane tam do příchodu nového kódu, který ho přepíše.

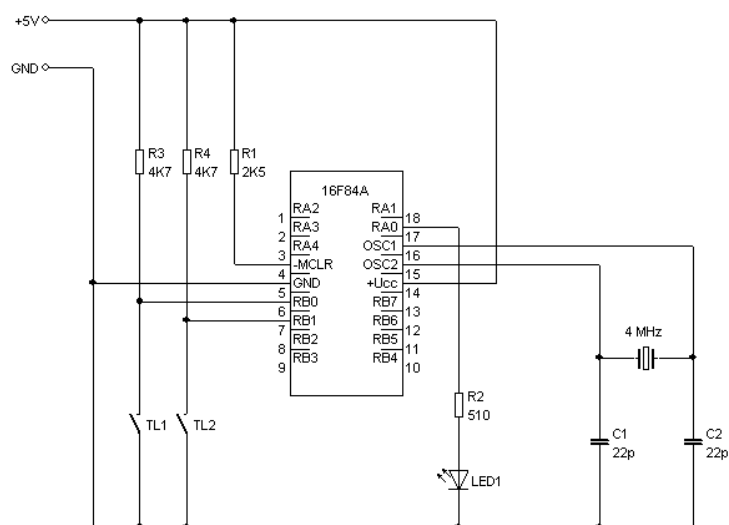
Popis protokolu: Pro přenos použijeme 16 bitový kód, 8 bitů je hlavička (pro kontrolu) a dalších 8 je datových. Hlavička má hodnotu 0Fh. Na straně přijímače je vždy přijatý kód negaci vysílaného!!!



Obr. 2: Popis protokolu

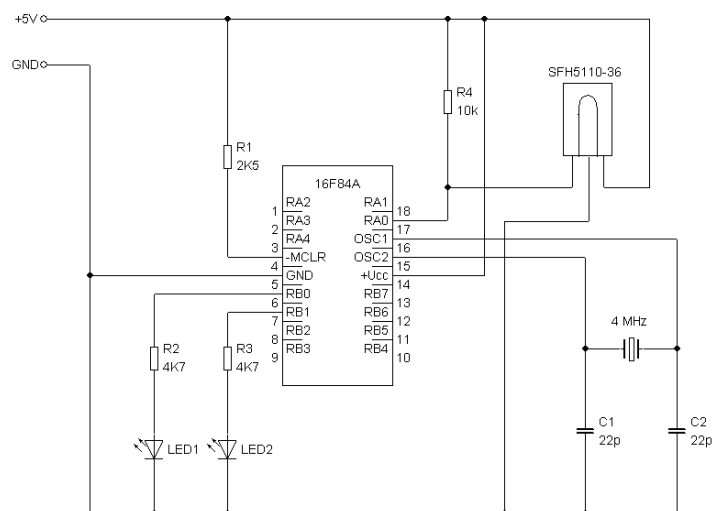
Každý bit je složen z 54 pulsu IR diody. Prvních 18 je vždy jednička, druhých 18 je datových (jednička nebo nula) a posledních 18 je vždy nula (1 a 3 slouží pro kontrolu na straně přijímače). Schéma bitu složeného z pulsu je na obrázku č. 2.

Schéma zapojení vysílače: (pro vysílání na větší vzdálenosti bude třeba připojit na výstup do IR diody tranzistor)



Obr. 3: Zapojení IR vysílače

Schéma zapojení přijímače:



Obr. 4: Zapojení IR přijímače