

## Paměti EPROM a EEPROM

**Další skupinou z našeho počátečního dělení pamětí ROM jsou paměti EPROM a EEPROM. Dnes se začneme věnovat těm prvním, tedy EPROM. Jak již víme jsou si tyto dvě skupiny velice podobné. V obou případech se jedná o programovatelné paměti ROM, tedy PROM.**

Paměti EPROM.

Další skupinou z našeho počátečního dělení pamětí ROM jsou paměti EPROM a EEPROM. Dnes se začneme věnovat těm prvním, tedy EPROM.

Jak již víme jsou si tyto dvě skupiny velice podobné. V obou případech se jedná o programovatelné paměti ROM, tedy PROM. Dalším společným znakem je obdobný postup při programování, které se zde provádí již elektricky a to zaznamenáním dané binární informace na patřičnou adresu v paměťové matici. Za zmínku snad ještě stojí, že se programování provádí ze stavu 1 bitu dané adresy do stavu 0. Pokud bychom tedy testovali nenaprogramovanou paměť EPROM, nebo EEPROM byl by výsledek vždy stejný. Ať bychom nastavili jakoukoliv adresu, na výstupech by se vždy objevila jen řada logických jedniček.



Programovací algoritmus je totožný. Nejprve je nastavena patřičná adresa, po které jsou nastavena data, která se programovacím impulsem přesné šířky přenesou do paměťové matice. Vše se pak opakuje, adresa – data - programovací impuls - kontrola.



Rozdíl mezi EPROM a EEPROM je však ve způsobu odstranění dat již uložených do paměti. Tento úkol se provádí například při upgradu programu, nebo pokud se při testování objeví chyba. Zatímco EEPROM se maže opět elektricky, stejně jako se programuje, EPROM se maže pomocí UV světla. V pouzdru paměti je zde malý otvor, pod kterým se nachází vlastní čip. Detail čipu je na obrázku. Uprostřed je na tenoučké destičce umístěn vlastní čip, který je, jak je velice dobře vidět, propojen s vývody vlastního pouzdra paměti tenkými vodiči.

Působením světelného záření na čip přes průzor se tak odstraní náboj, který představoval zaznamenaná data a paměť je opět prázdná. Na všech adresách je na každém výstupu log.1.

Příklady pamětí EPROM z praxe se kterými se v seriálu setkáme budou 27C256 a 27C512. Jak napovídá již název jedná se o paměti s velikostí paměťové matice 256 a 512K bitů. Paměti jsou organizované jako 32 a 64Kslov po 8 bitech.

Mezi jejich další vlastnosti patří:

- napájecí napětí 5V DC
- kompatibilita s technologií TTL
- možnost opakovaného programování
- schopnost zapamatování dat po dobu dosahující až 100 let

Příští díl bude věnován problematice programování těchto pamětí v amatérských podmínkách.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 7. ledna 2005

**Problematika programování polovodičových pamětí je obecně velice rozsáhlou kapitolou. V našem seriálu jsme ze zatím zabývali pouze jednoduchým programováním pamětí RAM a PROM. Vždy jsme si ukázali jednoduché zapojení, které splnilo požadovanou funkci a umožnilo tedy naprogramovat daný obvod a tím jsme si tak mohli ověřit jeho činnost.**

Programování EPROM v domácích podmínkách.

Problematika programování polovodičových pamětí je obecně velice rozsáhlou kapitolou. V našem seriálu jsme ze zatím zabývali pouze jednoduchým programováním pamětí RAM a PROM. Vždy jsme si ukázali jednoduché zapojení, které splnilo požadovanou funkci a umožnilo tedy naprogramovat daný obvod a tím jsme si tak mohli ověřit jeho činnost. U skupiny pamětí EPROM a později i EEPROM však takové jednoduché zapojení postačovat nebude.

Veliká variabilnost a také velká kapacita paměťové matice jsou u těchto pamětí vykoupeny o něco složitějším procesem vlastního programování.

Jednotlivé postupy programování mají své specifické úkoly, které je třeba pro zdárný výsledek dodržet.



Asi nejpatrnější odlišností od programování RAM a PROM, které jsme si uvedli je značně větší velikost programovacího napětí. Místo zvýšení napájecího napětí z 5ti voltů na deset, které bylo nutné provést při programování 74188, je pro naprogramování paměti EPROM, která může být zastoupena například obvodem 27C256 potřeba programovací impuls o napěťové úrovni pohybující se okolo 20ti voltů. Napájecí napětí se zde nemění a jeho velikost je pět voltů.

To je jen jedna z podmínek, které je nutné dostat pro úspěšné naprogramování.

Tento problém řeší profesionální programátory. Na druhou stranu je třeba podotknout, že pouhý programátor nestačí.

Pokud pomineme potřebu počítače s volným LPT portem, je zapotřebí i konkrétní program, nejlépe od výrobce programátoru. Cena se tak vyšplhá do ne moc sympatických částek.

Pro amatérskou činnost, kdy si chceme naprogramovat jednoduchou věc, která bude obsažena na pár adresách je však možné použít variantu jinou. Programátor který nebude závislý na PC a tudíž ani na nějakém programu bez kterého nefunguje.

Řešení je to velice jednoduché a hlavně o poznání levnější, než profesionální výrobek a svůj účel splní naprosto stejně.

Příklad takového ručního programátoru je na fotografii. Jak je patrné, je určen pro paměti 27C256 a 512. Dále jsou zde pak spínače pro nastavení adresy a dat, která se mají na tuto adresu zaznamenat. Po stlačení spínače MEM jsou pak tyto data zaznamenána a může se pokračovat další adresou. Tento programátor dokáže nejen data zapisovat, ale i je z paměti číst. Volba módu čtení nebo zápisu se provádí posledním přepínačem. WRITE - zápis, READ - čtení.

Příště se podíváme na tento programátor blíže a vysvětlíme si jak vlastně funguje.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Datum: 14. ledna 2005

**91 - Jak jsme si slíbili na konci minulého dílu, budeme se dnes blíže zabývat vlastním programátorem paměti EPROM. Princip jeho ovládání jsme si již osvětlili. Co se tedy nalézá uvnitř? V prvé řadě to je externí adaptér, který sníží síťové napětí 230V na 15V a ještě ho usměrní.**

Programátor EPROM

Jak jsme si slíbili na konci minulého dílu, budeme se dnes blíže zabývat vlastním programátorem paměti EPROM. Princip jeho ovládání jsme si již osvětlili. Co se tedy nalézá uvnitř?

Nejlépe bude asi začít zdrojem elektrické energie. V prvé řadě to je externí adaptér, který sníží síťové napětí 230V na 15V a ještě ho usměrní. Z 230V/50Hz vznikne tedy 15V stejnosměrných. Tato velikost napětí je už zcela bezpečná a plně postačující pro napájení vlastního programátoru. Uvnitř programátoru se s napětím provede několik věcí. V první řadě se nejprve sníží na napětí 12V a po té ještě na hodnotu 5V. Důvod těchto úprav je velice jednoduchý. Jelikož je programátor určen pro paměti typu 27C256 a 512, které jsou postaveny na koncepci obvodů TTL, je nutné zajistit jim stabilní napájecí napětí 5ti voltů. Druhá hodnota, tedy 12V, je zde potřeba pro programování. Programovací impuls, který musí mít nejen přesné parametry co se týče doby jeho trvání, také potřebuje určitou úroveň napětí. Tedy 12 voltů.



Další částí jsou vlastní spínače pro nastavení programovaných dat a adres. Konkrétně se jedná o jednoduché páčkové přepínače, kde je prostřední kontakt propojen vždy s danou adresou, nebo datovým vstupem/výstupem. Přepínače určené pro nastavení dat jsou propojeny k těmto kontaktům ještě přes polovodičové spínače. Ty jsou ovládány pomocí přepínače určujícího, zda se z paměti mají data číst, nebo se do ní mají data zapsat, READ/WRITE. V případě, že se mají data z paměti číst, odpojí tyto polovodičové spínače mechanické přepínače, které by jinak zkreslovaly vlastní logické úrovně výstupů.

Indikace toho, zda je na datovém výstupu, taktéž na adresovém vstupu log. jednička, nebo nula zobrazují LED diody, které se vždy nacházejí u každého z přepínačů a jsou napojeny na jejich prostřední kontakty. Posledním prvkem v zapojení, je vlastní zdroj programovacího impulsu. Ten je tvořen klasickým obvodem NE555, zapojeným v řízeném, režimu. Po stlačení tlačítka MEM se vždy zaktivuje obvod NE555 a vyšle do paměti, která má již nastavené datové a adresové vstupy, programovací impuls.

Funkce i zapojení programátoru jsou tedy velice jednoduché, což je dobrý předpoklad pro jeho stavbu.

Příště si ukážeme již vlastní schéma zapojení programátoru a popíšeme si postup jeho stavby.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Datum: 21. ledna 2005

**Dnes se budeme věnovat již vlastní konstrukci napájecí části programátoru. Jak jsme si uvedli v minulém dílu, je pro chod celého zařízení zapotřebí dvou napěťových úrovní. Hodnota 5V pro primární napájení paměti spolu s ostatními součástmi, jako jsou LED diody a například polovodičové spínače a druhá hodnota 12V. Ta je potřeba jen při programování.**

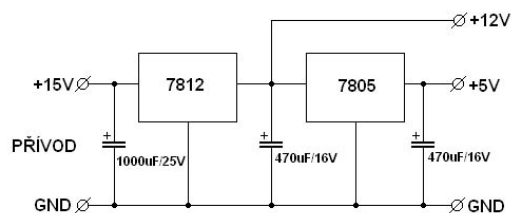
Programátor EPROM – konstrukce napájecí části.

Dnes se budeme věnovat již vlastní konstrukci napájecí části programátoru. Jak jsme si uvedli v minulém dílu, je pro chod celého zařízení zapotřebí dvou napěťových úrovní.

Hodnota 5V pro primární napájení paměti spolu s ostatními součástmi, jako jsou LED diody a například polovodičové spínače a druhá hodnota 12V. Ta je potřeba jen při programování.

Primárním zdrojem nezbytné elektrické energie je externí adaptér, který sníží síťové napětí 230V na 15V a ještě ho usměrní. Z 230V/50Hz tedy vznikne 15V stejnosměrných. S touto hodnotou napětí, která je již zcela bezpečná a plně postačující pro napájení vlastního programátoru pracujeme dál. Následně je potřeba tuto velikost ještě dále snížit na 12V a 5V.

Tím se už dostáváme k vlastnímu zapojení napájecí části a ke konkrétním součástkám. Tento blok obsahuje dva integrované stabilizátory v pouzdru TO220 s dovolenou proudovou zátěží 1A. Konkrétně se jedná o typy s označením 7805 a 7812.



Pokud si prohlédnete schéma zapojení, je zřejmé, že jsou oba stabilizátory zapojeny za sebou, dalo by se říci v sérii. První v pořadí je 7812, který sníží 15V od adaptéru na 12V a dále je pak tato hodnota napětí přivedena do 7805, který ji sníží již na pouhých 5V.

V zapojení jsou pak ještě použity nezbytné elektrolytické kondenzátory. Jejich umístění je vždy před vstupem do stabilizátoru a ihned za jeho výstupem.

Důležitou vlastností celého programátoru je to, že vněm není použit ani jeden plošný spoj. Vše je vzájemně propojeno pomocí vodičů a následně zalito hmotou pro zalévání vinutí transformátorů.

Výjimkou není ani napájecí část. Všechny součástky jsou vhodně rozmístěny a vzájemně propojeny u napájecího konektoru, do kterého za zapojuje přívod od adaptéru. Z tohoto kompletu jsou pak pouze vyvedeny tři vodiče, z nichž jeden slouží jako záporný pól – GND a zbylé dva jsou zdrojem napájecích napětí 12 a 5 voltů.

Příště budeme pokračovat v rozboru schématu a zaměříme se na ovladače adres a datových vstupů.

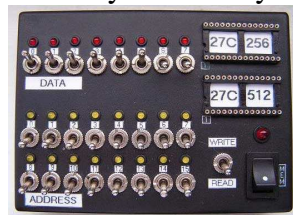
Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala (28. 1. 2005)  
iABC,

**Dnešní díl bude věnován podrobnějšímu popisu té části programátoru, která je zodpovědná za řízení adresových a datových vstupů programované paměti. Toto řízení je prováděno manuálně uživatelem pomocí páčkových přepínačů - každý řídí jeden adresový a datový vstup/výstup.**

Programátor EPROM - zapojení ovladačů datových a adresových vstupů.

Dnešní díl bude věnován podrobnějšímu popisu té části programátoru, která je zodpovědná za řízení adresových a datových vstupů programované paměti.

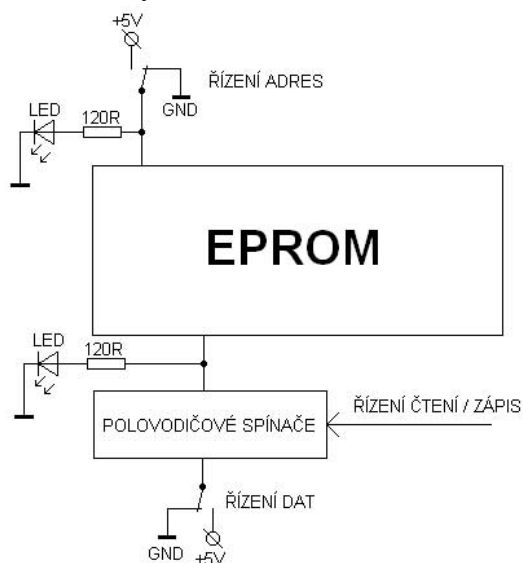


Jak je patrné z fotografie celého kompletu a i z dřívějších dílů seriálu, je toto řízení prováděno manuálně uživatelem pomocí páčkových přepínačů. Každý adresový a datový vstup/výstup je řízen jedním přepínačem, ke kterému je ještě pro lepší indikaci logické hodnoty připojena LED dioda. Když svítí indikuje logickou jedničku, v případě opačném indikuje logickou nulu.

Zatímco u adresových vstupů jsou přepínače napojeny rovnou k vývodům paměti, u vstupů datových je situace o něco složitější.

Mezi paměť a přepínače jsou zde ještě zařazeny polovodičové spínače, které slouží k odpojení přepínačů nachází-li se paměť v čtecím módu. Data z ní čteme a ne zapisujeme.

Tato úprava je nutná z toho důvodu, že pokud by byly přepínače neustále připojeny k výstupům, docházelo by v čtecím módu k ovlivňování výstupních logických hodnot.



Vše je asi nejlépe patrné ze schématického zapojení. Krajní kontakty přepínače jsou připojeny k +5V a GND, zatímco kontakt prostřední je napojen na patřičné vývody paměti.

U přepínačů pro datové vstupy/výstupy, jsou do této cesty zakomponovány již zmiňované polovodičové spínače, které jsou ovládány přepínačem určujícím zda se bude z paměti číst, nebo se do ní bude zapisovat.

Stejně i jako část zdrojová je tento ovládací komplet propojen bez použití plošného spoje. Vše bylo propojeno pomocí tenkých vodičů a zalito hmotou pro zalévání transformátorů.

Příště budeme pokračovat v rozboru schématu a zaměříme se na polovodičové spínače a část pro řízení zápisu a čtení.

Autor: Jindřich Fiala (4. 2. 2005)



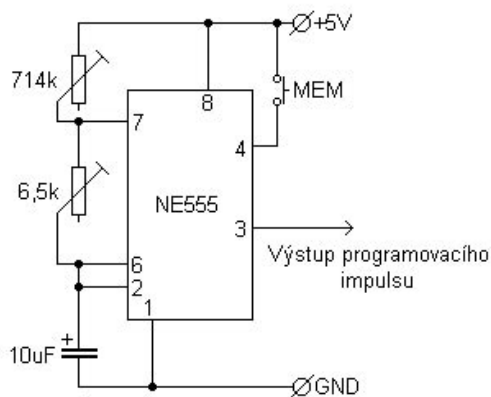
**Další velice důležitou částí celého programátoru, je blok pro řízení zápisu a čtení, spolu s polovodičovými spínači a generátorem programovacího impulsu. Funkci polovodičových spínačů jsme si objasnili již dříve, dnes si tedy jen uvedeme konkrétní obvod.**

Programátor EPROM - řízení zápisu a čtení, zdroj programovacího impulsu.

Další velice důležitou částí celého programátoru, je blok pro řízení zápisu a čtení, spolu s polovodičovými spínači a generátorem programovacího impulsu.

Funkci polovodičových spínačů jsme si objasnili již dříve, dnes si tedy jen uvedeme konkrétní obvod. V katalogu ho nalezneme pod označením 74641, nebo 645. V obou případech se jedná o oboustranný spínač, který dokáže najednou ovládat až osm datových vodičů a to bez ohledu na to, kde bude zdroj signálu. Spínač je totiž obousměrný.

Jeho řízení je obstaráno pomocí části pro určení, zda se bude z paměti číst, nebo se do ní budou data zapisovat. Této části se podrobněji věnovat nebudeme, jelikož si uvedeme zapojení zjednodušeného programátoru pouze pro jeden typ paměti, se kterou pak také budeme dále pokračovat v pokusech. Pokud by to přeci jen někoho zajímalo, je část určující zápis a čtení opatřena několika relátko, která jsou ovládána pomocí jednoho přepínače s označením READ/WRITE.



Další částí je zdroj programovacího impulsu. Tento impuls se musí vyznačovat přesnou dobou trvání a musí být aktivní v logické nule.

Pro jeho generování byl zvolen tradiční obvod NE555, zapojený v řízeném režimu. To znamená, že impuls se na výstupu tohoto obvodu objeví až po stlačení daného tlačítka. V našem případě se jedná o tlačítko označené jako MEM.

Aktivní doba trvání impulsu je maximálně 50ms (milisekund). Pro náš programátor jsme zvolili hodnotu 45ms. To jak bude výsledný impuls přesný, závisí na zvolených součástkách a jejich přesném nastavení. Pro obvod NE555, který má produkovat takovýto impuls je zapotřebí kondenzátor s kapacitou 10 mikrofaraďů a rezistory s hodnotou odporu 714 a 6,5 kiloohmu. Vhodnější je použití potenciometrů, které se na tuto hodnotu přesně nastaví.

Stejně i jako pro předešlé bloky zapojení, není ani zde použit jakýkoliv plošný spoj. Při zapojování je však potřeba velká pečlivost u nastavování hodnot potenciometrů.

Příště budeme pokračovat ve schématu dále a popíšeme si, jaký je rozdíl v zapojení pamětí 27C256 a 512 a co z toho vyplývá pro náš programátor.

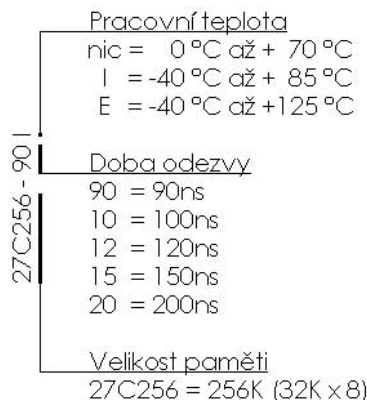
Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Datum: 11. února 2005

**Dnes se budeme věnovat popisu odlišností zapojení pamětí 27C256 a 27C512, pro které je určen náš programátor. Prvním rozdílem, který je patrný již z označení, je velikost vlastní paměťové matice. Typ 27C256 obsahuje paměť o velikosti 256 K bitů, kdežto typ 512 plný dvojnásobek.**

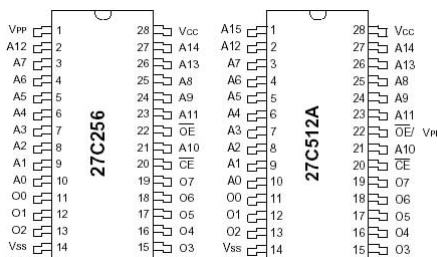
Programátor EPROM - odlišnosti pamětí 27C256 a 512.

Dnes se budeme věnovat popisu odlišností zapojení pamětí 27C256 a 27C512, pro které je určen náš programátor.



Prvním rozdílem, který je patrný již z označení, je velikost vlastní paměťové matice. Typ 27C256 obsahuje paměť o velikosti 256 K bitů, kdežto typ 512 plný dvojnásobek. Pro naše pokusy však bude plně stačit 27C256, který se vyznačuje také jednodušším ovládáním. Neobsahuje totiž žádné kontakty s dvojí funkcí.

Rozložení jednotlivých vývodů je nejlépe patrné z obrázku. Oba typy se vyznačují stejným rozložením datových vstupů/výstupů a stejnou polohou napájecích kontaktů (Vcc = +5V a Vss = GND).



Adresové vstupy se liší jen v jednom případě a to u paměti 27C512, kde je třeba díky větší paměťové matici také větší počet adresových vstupů. Přesně o jeden. Ten se nachází na místě kontaktu označeném jako Vpp u 27C256. Větší paměťová matice u 27C512 tak způsobila sloučení některých nastavovacích vstupů do jednoho vývodu. Konkrétně se jedná o negovaný vstup OE se vstupem Vpp, který musel uvolnit místo adresovému vstupu A15.

### 27C256

Operation Mode	$\overline{CE}$	OE	VPP	A9	O0 - O7
Read	VIL	VIL	Vcc	X	DOUT
Program	VIL	VIH	VH	X	DIN
Program Verify	VIH	VIL	VH	X	DOUT
Program Inhibit	VIH	VIH	VH	X	High Z
Standby	VIH	X	Vcc	X	High Z
Output Disable	VIL	VIH	Vcc	X	High Z
Identity	VIL	VIL	Vcc	VH	Identity Code

### 27C512

Operation Mode	$\overline{CE}$	$\overline{OE}/VPP$	A9	O0 - O7
Read	VIL	VIL	X	DOUT
Program	VIL	VH	X	DIN
Program Verify	VIL	VIL	X	DOUT
Program Inhibit	VIH	VH	X	High Z
Standby	VIH	X	X	High Z
Output Disable	VIL	VIH	X	High Z
Identity	VIL	VIL	VH	Identity Code

Logické úrovně jednotlivých vstupů pro módy čtení - Read, zápisu - Program, nebo stavu s nízkým odběrem proudu - Standby, jsou patrné z tabulek.

Posledním obrázkem, který si dnes uvedeme, je klíč pro dešifrování dalších důležitých údajů o pamětech. Z hodnot uvedených výrobcem na pouzdru, se dá kromě velikosti paměťové matice vyčíst i to, jaký je rozsah pracovních teplot, ve kterých je zaručena bezproblémová funkce paměti, nebo jaká je doba odezvy daného obvodu. Tento údaj udává, za jak dlouho se po nastavení dané adresy objeví data na této adrese uložená na výstupech. Řádově se jedná o desítky nanosekund.

Příště budeme v rozboru těchto pamětí pokračovat a více se již zaměříme na 27C256.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

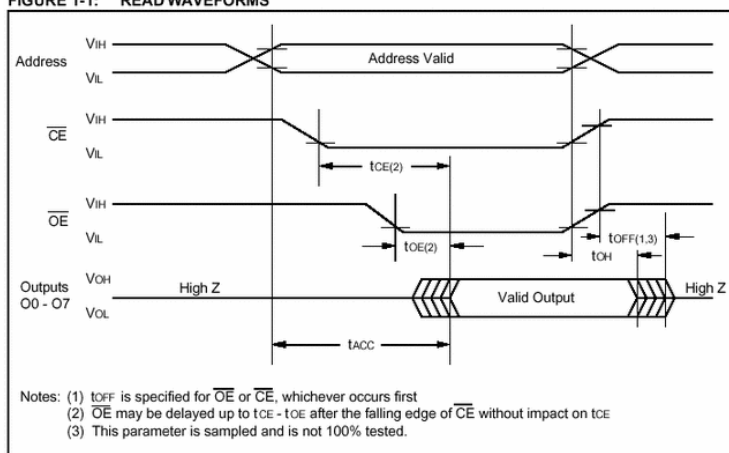
Datum: 18. února 2005

**Programátor je již z větší části hotov, proto se dnes budeme blíže věnovat vlastní paměti, kterou budeme programovat. V minulém dílu jsme si naznačili již první rozdíly mezi typy 27C256 a 27C512. Také jsme si uvedli, že budeme pro naše pokusy používat typ 27C256, který je vhodnější díky jednoduššímu ovládání režimů zápisu a čtení.**

Podrobný rozbor parametrů paměti 27C256.

Programátor je již z větší části hotov, proto se dnes budeme blíže věnovat vlastní paměti, kterou budeme programovat. V minulém dílu jsme si naznačili již první rozdíly mezi typy 27C256 a 27C512. Také jsme si uvedli, že budeme pro naše pokusy používat typ 27C256, který je vhodnější díky jednoduššímu ovládání režimů zápisu a čtení.

FIGURE 1-1: READ WAVEFORMS



Logické úrovně jednotlivých vstupů pro módy čtení - Read, zápisu - Program, nebo stavu s nízkým odběrem proudu - Standby, si doplníme o názorné grafy, které budou v časových úsecích vždy znázorňovat logické úrovně jednotlivých nastavovacích vstupů.

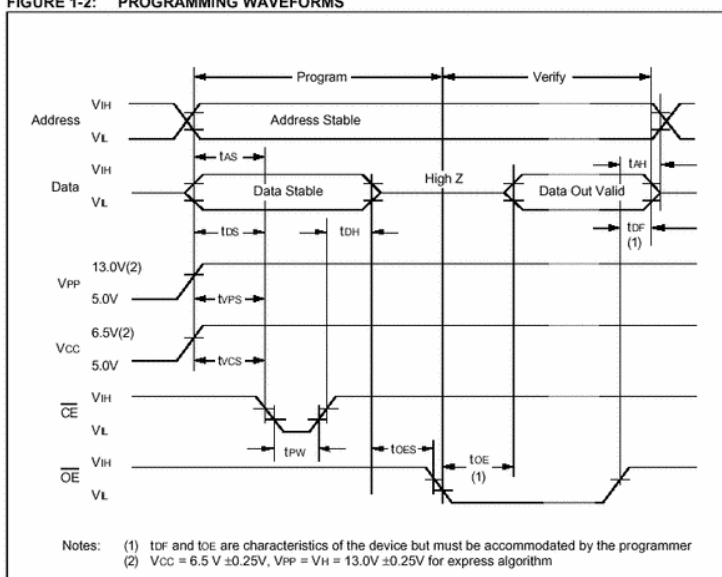
Jednotlivá vyobrazení obsahují tyto položky: Adress - adresové vstupy, Outputs - vstupy/výstupy informací, to podle toho zda se jedná o zápis, nebo čtení a dále pak nastavovací stupy CE a OE. Graf pak vždy zobrazuje, zda se daný kontakt nachází ve stavu VIH, nebo VIL

(logická jednička, nebo logická nula).

U grafu pro zápis pak ještě nalezneme vstup VPP, který se při programování musí nalézat na napěťové úrovni 13 V a při čtení na 5 V (logická jednička).

Nejlépe bude začít režimem čtení (READ WAVEFORMS). Budeme předpokládat, že v paměti jsou již nějaká data zaznamenána.

FIGURE 1-2: PROGRAMMING WAVEFORMS



Prvním krokem je nastavení patřičné adresy. Tu zvolíme kombinací logických úrovní na adresových vstupech. Adresa je tedy nastavená a zbývá pouze přivést i určené logické úrovně k negovaným nastavovacím vstupům OE a CE. Oba se musí nalézat v logické nule.

Po tomto úkonu se na datových výstupech (outputs) objeví data zaznamenaná na naší zvolené adrese.

Druhým módem, který budeme při testovacích zapojeních potřebovat, je režim zápisu (PROGRAMMING WAVEFORMS).

Zde je situace o něco komplikovanější, ale není to zas tak strašné. Prvním krokem je opět nastavení adresy, kam ovšem tentokrát chceme data zapsat.

Druhým krokem je třeba nastavit data, která chceme zaznamenat. To se provede přivedeným patřičných logických úrovní k datovým vstupům. Ty slouží při čtení i jako datové výstupy.

Nyní přijde na řadu vstup VPP, zde je nutné zvýšit napětí z 5 V na 13 V. Předposledním krokem je nastavení negovaného vstupu OE do logické jedničky.

Negovaný vstup CE, který zbyl, slouží pro přivedení programovacího impulsu. Ten jak již víme, musí být aktivní v logické nule a trvat přesně 50ns (45).

Po jeho přivedení se data nastavená na vstupech zaznamenají na určenou adresu.

Příště budeme pokračovat v konstrukci programátoru.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

**Dnes se budeme věnovat posledním úpravám programátoru, před jeho prvním užitím pro vlastní programování pamětí. V průběhu několika dílů, jsme si postupně rozebrali jednotlivé části celého zapojení, které zbývá jen umístit do vhodné plastové krabičky, vzájemně je propojit a zalít hmotou pro vinutí transformátorů.**

Dokončení konstrukce jednoduchého programátoru EPROM

Dnes se budeme věnovat posledním úpravám programátoru, před jeho prvním užitím pro vlastní programování pamětí. V průběhu několika dílů, jsme si postupně rozebrali jednotlivé části celého zapojení, které zbývá jen umístit do vhodné plastové krabičky, vzájemně je propojit a zalít hmotou pro vinutí transformátorů.

Důležitým prvkem v celé konstrukci je také vizuální provedení celého výrobku, převážně pak označení jednotlivých přepínačů pro data a adresy.

Pro prototyp byla zvolena plochá, plastová krabička, do které se bez problémů podařilo umístit a také vzájemně propojit všechny bloky zapojení. Před vlastním zalitím je však dobré ještě vše odzkoušet.

Značení jednotlivých částí lze provést několika způsoby. Nejjednodušší je si navrženou předlohu vytisknout laserovou tiskárnou na samolepící fólii. Tu pak rozdělit na potřebné části s jednotlivými čísly adres, nebo jinými popisky a jednoduše ji nalepit na krabičku.



Při označování je třeba držet se stanoveného řádu. Nejvhodnější je označit datové přepínače od 0 do 7 a adresové přepínače jako 0 až 15, respektive 14, protože v našem případě budeme používat paměť 27C256, která má adresových vstupů jen čtrnáct.

Dále je vhodné označit někde vedle patice pro paměť, kde se nachází první vývod. Pro snadnější ovládání programátoru i jinou osobou, než konstruktérem, je pak vhodné označení přepínače, kterým se řídí režim zápisu a čtení.

Zde je možné například použít nápisy WRITE / READ (zápis / čtení) a pro popisek tlačítka, kterým se spouští programovací impuls například zkratku MEM.

Příště se již budeme věnovat programování paměti EPROM pomocí našeho programátoru.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala (4. 3. 2005)

**Prvním programem, který si zkusíme společně naprogramovat pomocí našeho jednoduchého programátoru pamětí EPROM, bude jen jakýsi blikač. Ze začátku nám půjde hlavně o to, abychom si osvojili základní postupy při programování. Ty nám pak budou sloužit dále, již při složitějších konstrukcích, kde bude potřeba vyvinout i složitější program.**

Vlastní programování EPROM – první program.

Prvním programem, který si zkusíme společně naprogramovat pomocí našeho jednoduchého programátoru pamětí EPROM, bude jen jakýsi blikač. Ze začátku nám půjde hlavně o to, abychom si osvojili základní postupy při programování. Ty nám pak budou sloužit dále, již při složitějších konstrukcích, kde bude potřeba vyvinout i složitější program.

Blikač bude složen z osmi LED diod, které budou ovládány výstupy paměti EPROM. Mezi paměť a vlastní ledky jsou ještě pro odlehčení výstupů paměti vřazeny NPN tranzistory BC546B.



ADRESA		DATA							
A0	A1 – A14	D0	D1	D2	D3	D4	D5	D6	D7
1	1	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1

Princip svitu diod je velice jednoduchý. Dá se rozdělit na dvě fáze. V první budou svítit diody ob jednu a ve druhé fázi, se jejich svit jen vymění. Ty co svítily zhasnou a naopak ty co byly zhaslé se rozsvítí.

Efekt to jistě není nijak úchvatný, ale jde nám v první řadě o osvojení si práce s vlastním programátorem.

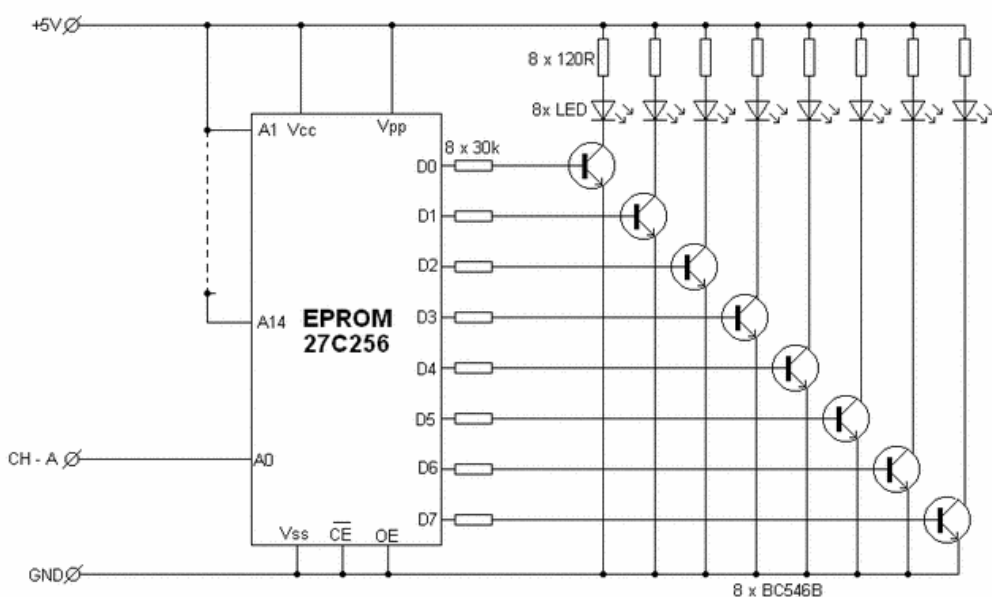
Co tedy dnes budeme potřebovat: programátor EPROM, paměť 27C256, tranzistory BC546B, rezistory s hodnotou odporu 120Ohmů, 30kOhmů a tentokrát i náš univerzální zdroj napětí a frekvence.

Princip činnosti programátoru a vlastní postup při programování, si již vysvětlovat nebudeme, dostatečně jsme se mu věnovali v několika předchozích dílech. Zabrousíme tedy rovnou k vlastnímu programu, který je třeba zaznamenat do paměti.

Vše je v binární podobě zaneseno do tabulky. Zde se nalézají logické úrovně jednotlivých datových a adresových vstupů.

Datová část nese informaci o tom, jaká z diod bude svítit a jaká bude v danou chvíli zhaslá (1 - svítí, 0 - nesvítí).

Adresovou částí budeme ovládat, jaká kombinace jedniček a nul se objeví na datových výstupech.



Jelikož pro řízení paměti - adresových vstupů, použijeme náš univerzální zdroj napětí a frekvence, máme k dispozici dvě logické úrovně 0 a 1. Ty se budou objevovat na výstupu zdroje frekvence. Dokážeme tedy řídit dvě adresy v paměti. Jedna adresa bude mít hodnotu :

0111111111111111 a druhá 1111111111111111. Na tyto adresy musíme zaznamenat patřičné binární informace k nim přiřazené. Jak je ostatně patné z tabulky.

Pokud je paměť naprogramována, zbývá pouze paměť zapojit do pokusného schématu, které je na obrázku, přivést napájecí napětí a zdroj impulsů pro řízení paměti.

Pokud by kupříkladu byl pro řízení paměti použit binární, nebo BCD čítač, dalo by se pochopitelně ovládat adresy více. S užitím jednoho binárního čítače až 16.

Příště budeme v programování pokračovat na dalším zapojení řízeném pomocí jednoduchého programu.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 11. března 2005

**Od jednoduchého blikátka dnes postoupíme k již zajímavějšímu efektu, který je velice oblíben mezi začínajícími elektroniky. Předvedeme si zapojení napodobující efekt běžícího světla, které přebíhá přes střed ze strany na stranu.**

Vlastní programování EPROM – jednoduché běžící světlo.

Od jednoduchého blikátka dnes postoupíme k již zajímavějšímu efektu, který je velice oblíben mezi začínajícími elektroniky. Předvedeme si zapojení napodobující efekt běžícího světla, které přebíhá přes střed ze strany na stranu.

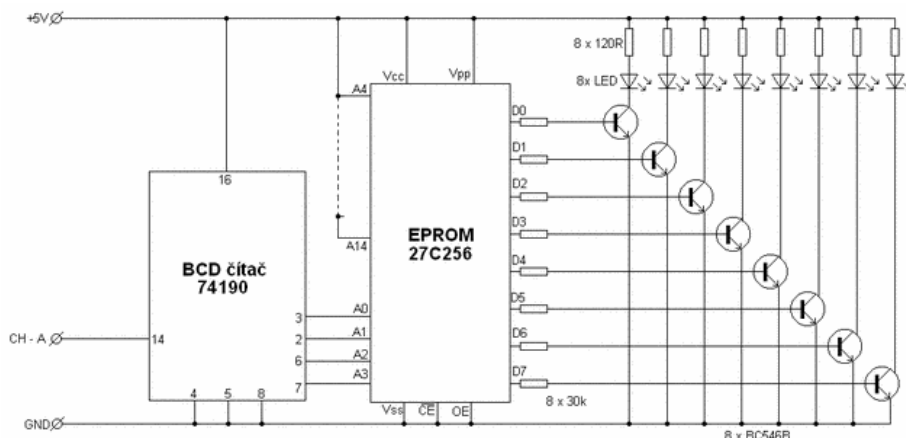
Stejně jako i minule bude k výstupům paměti připojeno osm LED diod, pro zobrazení daného efektu.

Co tedy dnes budeme potřebovat: programátor EPROM, prázdnou paměť 27C256, BCD čítač 74190, tranzistory BC546B, rezistory s hodnotou odporu 120Ohmů, 30kOhmů a univerzální zdroj napětí a frekvence.

Postup při programování jsme si již objasnili, proto se budeme zabývat pouze binárním programem a vlastním zapojením.

	ADRESA					DATA								
	A0	A1	A2	A3	A4 – A14	D0	D1	D2	D3	D4	D5	D6	D7	
0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
1	1	0	0	0	1	0	1	0	0	0	0	0	1	0
2	0	1	0	0	1	0	0	1	0	0	1	0	0	0
3	1	1	0	0	1	0	0	0	1	1	0	0	0	0
4	0	0	1	0	1	0	0	1	0	0	1	0	0	0
5	1	0	1	0	1	0	1	0	0	0	0	1	0	0
6	0	1	1	0	1	1	0	0	0	0	0	0	0	1
7	1	1	1	0	1	0	0	0	0	0	0	0	0	0
8	0	0	0	1	1	0	0	0	0	0	0	0	0	0
9	1	0	0	1	1	0	0	0	0	0	0	0	0	0

Informace, kterou je třeba zaznamenat do paměti je opět v tabulce. Tentokrát je poněkud obsáhlejší. Pro řízení paměti je použit BCD čítač, který jak víme dokáže na svých čtyřech výstupech zobrazit deset rozličných binárních informací. Můžeme tedy ovládat deset samostatných adres paměti, do kterých lze vždy zaznamenat po osmibitové binární informaci.



Při pohledu na datovou část tabulky, je na první pohled patrný motiv kříže sestavený z jedniček. Ty jak víme představují svítící LED diodu. Pokud budeme postupovat po jednotlivých řádcích, zjistíme, že se poloha jedniček postupně mění. Od krajních poloh se postupně přibližují s každým dalším řádkem ke středu a po dosažení středu se zase navrací zpět. Výsledkem je efekt světla, které přebíhá z obou stran přes střed a zase zpět.

Poslední dva řádky v datové části tabulky jsou vyplněny pouze nulami. Po proběhnutí jednoho cyklu efektu vždy tak nastane na krátkou chvíli malá odmlka.

Rychlost jakou se bude světlo pohybovat a také jakou dobu bude trvat pauza ovlivňuje velikost frekvence přivedené ke vstupu BCD čítače. Čím bude větší, tím se také bude světlo pohybovat rychleji.

Příště budeme v programování pokračovat na dalším zapojení, kde bude pro řízení paměti použit binární čítač.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

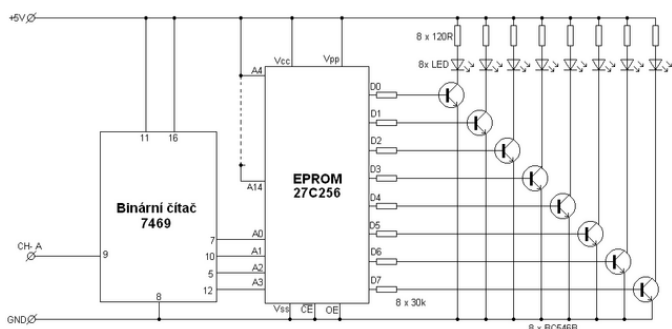
Datum: 18. března 2005

V předposledním dílu našeho seriálu použijeme pro řízení paměťových adres testované paměti EPROM místo čítače BCD, již čítač binární. Tato záměna nám umožní ovládat plných šestnáct adresových míst a uložit do paměti složitější program pro řízení LED diod připojených k datovým výstupům paměti.

Vlastní programování EPROM – řízení pomocí binárního čítače.

V předposledním dílu našeho seriálu použijeme pro řízení paměťových adres testované paměti EPROM místo čítače BCD, již čítač binární. Tato záměna nám umožní ovládat plných šestnáct adresových míst a uložit do paměti složitější program pro řízení LED diod připojených k datovým výstupům paměti.

Výsledkem bude opět efekt běžícího světla, které bude tentokrát přebíhat z jedné strany na druhou.



Pro dnešní pokusné zapojení budeme potřebovat tyto komponenty: programátor EPROM, prázdnou paměť 27C256, binární čítač 7469, tranzistory BC546B, rezistory s hodnotou odporu 120Ohmů, 30kOhmů a univerzální zdroj napětí a frekvence.

Informace v binární podobě, kterou je opět potřeba zaznamenat do paměti je uvedena v tabulce. Oproti minulému dílu je opět o něco obsáhlejší.

Datová část tabulky v sobě ukrývá vlastní světelný efekt. Jak bylo již řečeno, bude světlo přebíhat z jedné strany na druhou. To je však jen část efektu. Po přeběhnutí dojde k dvěma krátkým rozsvícením celé řady LED diod, po kterém bude následovat zabliknutí vždy jedné diody. Po skončení tohoto posledního kroku opět dojde přeběhnutí světla a vše se bude opakovat znovu.

	ADRESA					DATA								
	A0	A1	A2	A3	A4 - A14	D0	D1	D2	D3	D4	D5	D6	D7	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Rychlost jakou se bude běžící světlo pohybovat a také jakou dobu budou trvat další momenty efektu, ovlivňuje velikost frekvence přivedené ke vstupu binárního čítače. Čím bude větší, tím se také bude efekt rychle měnit.

V příštím a také zároveň posledním dílu tohoto seriálu, si ukážeme jak použít paměť EPROM co by dekodér BCD kódu pro řízení LED displeje.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu:

[cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: (25. 3. 2005)

iABC,

Poslední díl našeho seriálu věnujeme konstrukci dekodéru kódu BCD, pro řízení sedmissegmentového LED displeje. Tuto úlohu zastane naše oblíbená paměť EPROM, kterou jsme až do teď používali spíše pro generování jednoduchých světelných efektů.

Vlastní programování EPROM – dekodér BCD pro sedmissegmentovku.

Poslední díl našeho seriálu věnujeme konstrukci dekodéru kódu BCD, pro řízení sedmissegmentového

	ADRESA					DATA								
	A0	A1	A2	A3	A4 - A14	D0 - a	D1 - b	D2 - c	D3 - d	D4 - e	D5 - f	D6 - g	D7	
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	0	1	1	1	0	0	0	0	0
2	0	1	0	0	1	1	1	0	1	1	0	1	1	1
3	1	1	0	0	1	1	1	1	1	0	0	1	1	0
4	0	0	1	0	1	0	1	1	0	0	1	1	1	1
5	1	0	1	0	1	1	0	1	1	0	1	1	1	0
6	0	1	1	0	1	1	0	1	1	1	1	1	1	1
7	1	1	1	0	1	1	1	1	1	0	0	0	0	0
8	0	0	0	1	1	1	1	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	0	1	1	1	0

LED displeje. Tuto úlohu zastane naše oblíbená paměť EPROM, kterou jsme až do teď používali spíše pro generování jednoduchých světelných efektů.

Princip takového dekodéru pro LED displej není potřeba zcela jistě dlouho vysvětlovat. Již jsme se sním

setkali v předchozích dílech seriálu věnovaných právě dekodérům. Tedy jen pro zopakování. Princip obvodu s takovým označením, spočívá v dekodování BCD informace přivedené k jeho čtyřem datovým vstupům a následném rozsvícení příslušných segmentů na LED displeji.

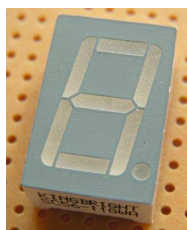
V praxi existuje mnoho variant takových obvodů, které jsou buď samostatné dekodéry, nebo v sobě obsahují i BCD čítač.

V našem dnešním zapojení bude hrát paměť EPROM roli pouze jednoduchého dekodéru, ke kterému použijeme externí čítač. Pro tuto úlohu použijeme obvod 74190, představující BCD čítač.

Zapojení je jako vždy velice jednoduché. Opět budeme potřebovat programátor EPROM, prázdnou paměť 27C256, BCD čítač 74190, tranzistory BC546B, rezistory s hodnotou odporu 120Ohmů, 30kOhmů, univerzální zdroj a také LED displej se společnou anodou (SA).

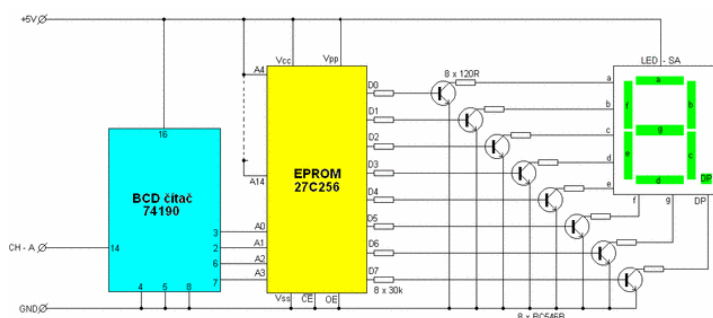
Vlastnímu zapojení se blíže věnovat nebudeme. Pokud budeme při zapojování precizní, musí vše fungovat na první pokus.

Zaměříme se tedy opět na vlastní binární informaci, kterou je třeba zaznamenat do paměti.



Všechna potřebná data obsahuje tabulka. Levá část je vyhrazena pro adresy, pravá pro konkrétní data příslušné adresy.

Jedničky v datové části představují svítící segmenty na LED displeji. První řádek zobrazuje rozsvícenou nulu, druhý jedničku, třetí dvojku, až do devítky, která je na posledním řádku.



Poslední datový výstup paměti – D7 je určen pro řízení desetinné tečky – DP. Zde se pravidelně střídá 1 a 0. Výsledkem tedy bude její blikání při postupném zobrazování cifer od 0 do 9.

Testování naprogramované paměti spočívá pouze v jejím zapojení do obvodu a v přivedení napájecího napětí a hodinových impulsů k BCD čítači.

Pokud bylo programování úspěšně, bez chyby, budou se na displeji zobrazovat dekodované hodnoty čísel, které generuje BCD čítač.

Rychlost s jakou se budou střídát čísla na LED displeji lze opět ovlivnit velikostí vstupní frekvence do čítače.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala (1. 4. 2005)

iABC,

Zdroj:

<http://abc.blesk.cz/clanek/serialy/5992/cislicova-technika-89.html>

<http://abc.blesk.cz/clanek/serialy/6016/cislicova-technika-91.html>

<http://abc.blesk.cz/clanek/serialy/6075/cislicova-technika-94.html>

<http://abc.blesk.cz/clanek/serialy/6084/cislicova-technika-95.html>

<http://abc.blesk.cz/clanek/serialy/6099/cislicova-technika-96.html>

<http://abc.blesk.cz/clanek/serialy/6124/cislicova-technika-98.html>

<http://abc.blesk.cz/clanek/serialy/6147/cislicova-technika-99.html>