

Stručne o pamätoch EPROM a o ich programovaní

V uvode sa treba zmienit vobec o funkcii pamate v pocitaci. Z ucebnic vieme, ze pamat sluzi na uchovavanie informacii, v najhrubsom rozliseni ide bud o programy (instrukcie), alebo o udaje. Pocas prace pocitaca je potrebne menit obsah pamatovych buniek - to je nezbytné k funkciiu pocitaca. Tuto zmenu obsahu umoznuju pamate typu RWM (Read Write Memory), castejsie nazyvane RAM (Random Access Memory - pamat s lubovolnym pristupom). Toto druhe oznacenie napoveda, ze urcite existuju aj pamate, kde pristup k informaciam - k obsahom buniek pamate - nie je lubovolne. Ozaj, su pamate - pouzivaju sa ako vonkajsie jednotky -, ktore pristup k obsahu bunky umoznuju len po vyhľadani vsetkych predoslych buniek (predosle sa rozumia tie, ktore podla organizacie buniek v pamati su pred vyhľadavanou). Tato nutnost, vlastna napríklad perifernym jednotkam pružnych a pevných diskov, nie je veľkou prekážkou vtedy, ak sa počítač k týmto jednotkam obracia malokedy, a rychlost prenosu je vysoka. Najlepsi, najrychlejsi pristup k obsahu maju pamate RWM - maju vsak dve nevyhody; prva, ich obmedzena veľkost (kapacita) v pocitaci, v dnesnej dobe je len symbolicka. Nie je financnym ani konstrukcny problemom zabudovat do pocitaca jednotky az desiatky megabajtov (tu predpokladam, ze citatel ma predstavu o zakladnych jednotkach v informatike). Druha nevyhoda, ze az na (so zvlastnymi prostriedkami dosiahnute) vynimky stracaju pamate RWM svoj obsah pri odpojeni napajacieho napatia, a samozrejme pri vypnutí pocitaca.

Predstavme si teraz počítač po zapnutí. Ma pamat RWM, ale prazdnu (alebo naplnenu nahodnymi nezmyselnymi udajmi), a potrebujeme do nej zaviesť predovšetkým program ovladajuci a riadiaci jednotlivé casti pocitaca (obecne sa tomu hovorí operacny system), dalej nejaky systemovy program - vacsinou programovaci jazyk, no a nakoniec zavedieme alebo cez klavesnicu napiseme nas vlastny (alebo ziskany) program. Pocas prevadzky pocitaca musi byt operacny system neustale v pamati. Aj najbeznejši nami pouzivany programovaci jazyk by tam mohol byt. Bolo by velmi pohodlne, keby sme sa o zavedenie tychto dvoch programoch po zapnutí vobec nemuseli starat. Kedze vsak nemame (len poniektorí z nas) rychlu a inteligentnu vonkajšiu pamat (pružny alebo pevný disk), po nuka sa nam (cez vyrobcoov nami pouzivanych tzv. domacich pocitacov) tretí druh pamate pre počítač, a to pamat taktiez polovodicova ako RWM (je jasne pri tom, ze diskove pamate su magneticke), ale s pevnym obsahom buniek. To znamena, ze aj bez napajacieho napatia si udrzi (na hociaku dobu) obsah vsetkych svojich buniek. Tento obsah v jednoduchych pripadoch sa ani neda zmenit - hovorime o pamätoch ROM (Read Only Memory - pamat, jej obsah sa da len citat, a po citani zostane bez zmeny). Je logicke, ze do takej pamate sa uklada taky obsah, ktory sluzi na vymedzeny ucel, a ten obsah sa nemusí nikdy menit.

Typickou aplikaciou pamati typu ROM je umiestnenie operacneho systemu aj programovacieho jazyka BASIC (ktory aj ked sa prakticky u kazdeho typu pocitaca od druheho mierne lisi, je vsak spolocnym "materinskim" jazykom tejto kategorie) domacich pocitacov. (Vo svete sa im hovorí Home Computer, Hobby Computer; u nas osobny mikropočítač). Prakticky to pre uzivatela znamena, ze po zapnutí jeho počítač hned "zije", a je schopny rozumiet jazyku BASIC. Samozrejme miesto BASIC-u moze byt zabu-

dovaný aj iný jazyk.

A teraz som tam, kde som mohol byť aj na začiatku, ale bez úvodu: aj keď obsah pamäte ROM je pevný, predsa nie vždy v pôvodnej forme vyhovuje. Alebo, urobím si program, ktorý je nemenný, odladený, často používaný, teda ho chcem mať často a rýchlo prístupný.

V týchto prípadoch má pamäť EPROM svoje opodstatnenie. Výhodu voči RAM má zrejmu; voči diskom má tu, že je oveľa rýchlejšia a prístupnejšia, nehovoriac o tom, že u nás ziaľ diskové jednotky nie sú rozšírené. (Inac v dohľadnej dobe mienim napísať článok o jednotkách pružných a pevných diskov).

Teraz konkrétne k pamätom ROM. Ich rozdelenie môže byť roznohodnotené. Snad najbezpečnejšie rozdelenie by mohlo byť také:

- ROM - Read Only Memory - jej obsah je vytvorený trebárs pri výrobe, výrobnou technológiou.
- PROM - Programmable ROM - jej obsah môže užívateľ sám určiť - pomocou vhodného zariadenia, ktorým určuje stav každého bitu obsahu (stavy môžu byť logická 1 a logická 0. O týchto veciach, a ako sa vytvára, respektívne funguje bunka pamäte, rádím precíznejšie - v prípade potreby - štát z odbornej literatúry). Tomuto postupu sa hovorí naprogramovanie, "napalenie" obsahu. Tento obsah sa nebude dať už nikdy zmeniť (okrem výnimočných prípadov, keď by nový obsah vhodne "prekryl" pôvodný).
- EPROM - Erasable PROM - ako predosle, ale v prípade potreby sa obsah dá vymazať a pamäť znovu novým obsahom naprogramovať. Vymazanie sa robí cez okienko na puzdre EPROM ultrafialovým svetlom. Počet cyklov napaleni / vymazanie býva pár desiatok.
- EEPROM - Electrically Erasable PROM - na rozdiel od predosleho sa vymazanie robí pomocou napätia. (Označuje sa ešte ako E²ROM, EEROM, EAROM.)

Este poznámku k výrazu "napalenie". Vhodným postupom pripojenie tzv. programovacieho napätia vytvára v štruktúre bunky pamäte (prakticky je to MOS-FET tranzistor) taký stav, aby sa výstup (vývod bunky) nachádzal po naprogramovaní (napalení) v požadovanom stave. Vlastne sa upravuje stav tohoto tranzistora z dvoch možných úrovní na jednu - pri druhej úrovni netreba nič s bunkou robiť. Vymazanie zabezpečí, aby sa všetky tranzistory všetkých buniek dostali do rovnakeho stavu - a znova môžu byť stavy požadovaných buniek zmenené - napätím - na opačné.

Z horeuvedeného vyplýva, že sa budeme zaoberať druhom EPROM, ktorý je najrozšírenejší, najlacnejší (z hľadiska aplikácie) a najpraktickejší.

Pri vývoji pamäti EPROM sa zrodil neprehľadný počet typov a druhov. Cosko- ro sa niektoré rozšírili a stali sa neoficiálnym štandardom, a to z hľadiska rozloženia vývodov, rozmerov, aj z hľadiska programovacieho postupu. V ďalšom budu popisované bežné typy EPROM, vyskytujúce sa vo svete.

Neobídeme sa však bez histórie. Prvé rozšírené typy (okolo r. 1972) EPROM-ov boli I2704 a I2708 firmy INTEL. Sú to pamäte s organizáciou 512 x 8 bitov a 1024 x 8 bitov, teda 4 Kbitov a 8 Kbitov. Vývodov majú 24, napájacie napätie je +5V, +12V, -5V. Majú pomerne zložitý a náročný postup programovania. I2708 sa ešte (najmä jeho ekvivalent M588708 u nás) používa. Tento typ začali vtedy vyrábať aj iné firmy. Dokonca firma Texas Instruments skorej vyšla na trh s TMS2716 (2 K x 8 bit), ako INTEL.

Zato od I2716 začína era aj dnešných rozšírených EPROM-ov. Vyzaduje totiž len napájanie +5V; programovací postup je jednoduchý. Tu upozorňujem, že ani vyvodová kompatibilita nie je medzi TMS2716 a I2716! Preto treba vedieť, že s TMS2716 (Texas) sú shodné: MOTOROLA TMS2716, NATIONAL SEMICONDUCTOR NMC2716. S I2716 sú shodné výrobky firmami: AMD, FAIRCHILD, NEC, OKI, NAT.SEMICONDUCTOR 2716; taktiež FUJITSU MB8516, MOTOROLA MCM2716 a nakoniec aj Texas TMS2516.

Ďalšie členy typovej rady Intel vychádzajú z I2716. I2732 má 4k x 8, I2764 má 8Kx8 ... atď. bitov. Dokonca dodatočne došlo k rozšíreniu "dole", I2758 má shodné vlastnosti (okrem kapacity) s I2716. Súčasne Texas Instruments dala na trh TMS 2532, TMS 2564; tie sa líšia od odpovedajúcich typov Intel rozložením vyvodov.

V súčasnej dobe sú bežne k dostaniu typy až 27512 od radu výrobcov. V Spectru je, ako známe, ROM 27128, čo sa dá nahradiť EPROM-om 27128, t.j. 16K x 8 bit = 16 KB.

Vyrazne zlepšenie priniesli EPROM vyrobené technológiou CMOS. Označujú sa zvyčajne ako 27C16 a podobne. Majú oveľa nižšiu spotrebu ako "klasické" EPROM.

Teraz pred tabuľkou najrozšírenejších typov EPROM, ich odlišností a vyvodov niekoľko slov o bežnom postupe použitia (nie programovania!) EPROM.

EPROM je typická súčiastka, ktorá sa dáva do objímky - mysliac na prípadnú neskorsiu úpravu obsahu. Táto úprava sa robí v programátore EPROM. (Na nom býva špeciálna "rychloupinacia" objímka.) Objímka v počítaní slúži aj na to, že pri vyvoji a ladení programu, ktorý chceme uložiť (hotový) do EPROM, nahradíme EPROM pamäť SRAM (statický RAM), ktoré majú shodné kapacity a vyvody (aj na to, že majú vyvod na riadenie zápisu do pamäte - WRITE). Ak je program odladený, môže sa EPROM týmto obsahom naprogramovať a používať. Tieto zariadenia sa obecné volajú ako simulátory EPROM.

Ďalšia vlastnosť, ktorá je potrebná (snad okrem typov CMOS) najmä pri väčšom počte EPROM v zariadení: možnosť režimu tzv. STANDBY - pohotovosť. Vtedy má obvod oveľa nižšiu spotrebu, ako pri aktívnom režime (pri čítaní z neho).


Pred popisom postupu programovania EPROM sa môžeme pozrieť na vyvody EPROM, a na tabuľku rôznych typov a druhov. Je pozorovateľná značná snaha na zachovanie kompatibility vyvodov; je narúšaná vlastne len pri prechode z 24 vyvodového typu (do 2732) na 28 vyvodový (od 2764) a pri privádzaní programovacieho napätia - táto posledná inkompatibilita platí vlastne len pri programovaní. Suma summarum, keď niekde navrhujeme EPROM 2732, tak radšej urobíme miesto pre 28 vyvodové puzdro, a potom nebude problém neskorsie ho nahradiť väčším typom.


Typ EPROM	Vpp	Typ EPROM	Vpp
2716	25 V	2764A	12,5 V (Typ CMOS)
2732	25 V	27128	21 V
2732A	21 V	27128A	12,5 V (Typ CMOS)
2764	21 V	27256	12,5 V (Ucc = 6V)

ROZLOZENIE VYVODOV PAMÄTI EPROM


512	256	128	64	32	2716				2716	2732	2764	128	256	512	
A15	Vpp	Vpp	Vpp			1	28					+5V	+5V	+5V	+5V
A12	A12	A12	A12			2	27					PGM	PGM	A14	A14
A7	A7	A7	A7	A7	A7	3	1	24	26	+5V	+5V	NC	A13	A13	A13
A6	A6	A6	A6	A6	A6	4	2	23	25	A8	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	A5	5	3	22	24	A9	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	A4	6	4	21	23	Vpp	A11	A11	A11	A11	A11
A3	A3	A3	A3	A3	A3	7	5	20	22	OE	OE/Vpp	OE	OE	OE	OE/Vpp
A2	A2	A2	A2	A2	A2	8	6	19	21	A10	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	A1	9	7	18	20	CE	CE	CE	CE	CE	CE
A0	A0	A0	A0	A0	A0	10	8	17	19	Q7	Q7	Q7	Q7	Q7	Q7
Q0	Q0	Q0	Q0	Q0	Q0	11	9	16	18	Q6	Q6	Q6	Q6	Q6	Q6
Q1	Q1	Q1	Q1	Q1	Q1	12	10	15	17	Q5	Q5	Q5	Q5	Q5	Q5
Q2	Q2	Q2	Q2	Q2	Q2	13	11	14	16	Q4	Q4	Q4	Q4	Q4	Q4
⊥	⊥	⊥	⊥	⊥	⊥	14	12	13	15	Q3	Q3	Q3	Q3	Q3	Q3

Obecný postup pri programovaní jednotlivých typov:

I2716 : Pripojíme programovacie napätie (Vpp), OE dame na H. Po aktivizácii adresných liniek zapálime obsah do bunky pripojením CE impulsom  50 ms dlhým.

I2732 : Vyvod Vpp a OE je spoločný. OE je potrebné len pri čítaní. Tzn. pri palení dame na vyvod 20 Vpp. Po aktivizácii adresných a datových liniek nasleduje impulz na CE  50 ms.


I2732 A : Je to typ CMOS. Vpp je 21 V.

I2764 : Ma už 28 vyvodov. Pripojíme Vpp, OE na H, aktivizujeme adresné a datové linky. CE dame na L. Impulsom  na PGM 50 ms napálime obsah bunky.

I2764 A : CMOS, Vpp je 12,5 V

I27128 : shodné s 2764

I27128 A: shodné s 2764 A

I27256 : Programovací postup sa zásadne líši od predoslych. Napájacie napätie sa zvýši z 5 V na 6 V. Pripojíme Vpp. Aktivizujeme adresné a datové linky. Vyvod CE sa cyklicky ovláda impulsom  1 ms; po každom kroku sa kontroluje zapísaný byte. Po prvom úspešnom prečítaní este opakujeme palenie trikrát dlhšie ako do prvého úspešného čítania. Tento tzv. rýchly (alebo interaktívny) programovací algoritmus sa používa aj u typov 2764 a 27128, s tým, že "doprogramovacia" doba je 4-násobná. Úspora času je u 27128 zhruba 13 minút (z 15 na 2 minúty). Každopadne sa pre konečné verzie (u 2764 a 27128) doporučuje klasický postup.

Tolko teda zbečne o programovaní EPROM. Este dve poznámky:

1. Najviac sa medzi nami používajú EPROM na nahradenie pôvodného obsahu ROM v Spectre vylepšeným obsahom. Doporučuje sa výmena za 27256, potom prepínaním jednej adresnej linky možno voliť jednu a druhú polovicu pamäti (trebárs kombináciu ORIGINAL / VYLEPSENÝ ROM).
2. Pomocou Spectra sa dá programovať (pomocou vhodného prípravku = HW+SW, zvaný programátor) trebárs aj 27256, dokonca aj 27512.

TABULKA NAJROZSIRENEJSICH EPROM S TECHNICKYMI UDAJMI
(neobsahuje prístupovú dobu, udávanú v nanosekundach)

VYROBCA	Typ	organizacia pamate	Vpp	progr. metoda	poznámka
AMD	AM9716	2K x 8	25 V	N	
	AM2716	2K x 8	25 V	N	
	AM2732	4K x 8	25 V	N	
	AM2732A	4K x 8	21 V	N	
	AM2764	8K x 8	21 V	N;I	
	AM2764A	8K x 8	12,5 V	I	
	AM27128	16K x 8	21 V	N;I	
	AM27128A	16K x 8	12,5 V	I	
	AM27256	32K x 8	12,5 V	I	
	AM27512	64K x 8	12,5 V	I	
Fujitsu	MBM2716	2K x 8	25 V	N	
	MBM8516	2K x 8	25 V	N	
	MBM2732A	4K x 8	21 V	N	
	MBM27C32A	4K x 8	21 V	N	CMOS
	MBM2764	8K x 8	21 V	N;I	
	MBM27C64	8K x 8	21 V	N;I	CMOS
	MBM27128	16K x 8	21 V	N;I	
	MBM27256	32K x 8	12,5 V	I	
	MBM27C256	32K x 8	12,5 V	N;I	CMOS
	MBM27C512	64K x 8	12,5 V	I	CMOS
Hitachi	HN462716	2K x 8	25 V	N	
	HN462532	4K x 8	25 V	N	
	HN462732	4K x 8	25 V	N	
	HN462732A	4K x 8	21 V	N	
	HN482764	8K x 8	21 V	N;I	
	HN27C64	8K x 8	21 V	N;I	CMOS
	HN482764P	8K x 8	21 V	N;I	OTP
	HN4827128	16K x 8	21 V	N;I	
	HN27128P	16K x 8	21 V	N;I	OTP
	HN27256	32K x 8	12,5 V	I	
	HN27512	64K x 8	12,5 V	I	
Intel	2716	2K x 8	25 V	N	
	2732A	4K x 8	21 V	N	

	P2732A	4K x 8	21 V	N	OTP
	2764	8K x 8	21 V	N;I	
	P2764	8K x 8	21 V	I	OTP
	2764A	8K x 8	12,5 V	I	
	27C64	8K x 8	12,5 V	I	CMOS
	P2764A	8K x 8	12,5 V	I	OTP
	27128	16K x 8	21 V	N;I	
	27128A	16K x 8	12,5 V	I	
	P27218A	16K x 8	12,5 V	I	OTP
	27256	32K x 8	12,5 V	I	
	27C256	32K x 8	12,5 V	I	CMOS
	87C256	32K x 8	12,5 V	I	CMOS
	27512	64K x 8	12,5 V	I	
	27513	4 x 16K x 8	12,5 V	I	Paged
Mitsubishi	M5L2716	2K x 8	25 V	N	
	M5L2732	4K x 8	25 V	N	
	M5L2764	8K x 8	21 V	N	
	M5L27128	16K x 8	21 V	N;I	
	M5L27256	32K x 8	12,5 V	I	
Mostek	MK2716	2K x 8	25 V	N	
Motorola	MCM2716	2K x 8	25 V	N	
	MCM27L16	2K x 8	25 V	N	LP
	MCM2532	4K x 8	25 V	N	
	MCM25L32	4K x 8	25 V	N	LP
	MCM68764	8K x 8	25 V	M	
	MCM68766	8K x 8	25 V	M	
	MCM68769	8K x 8	25 V	M	
National Semiconductor	NMC2716	2K x 8	25 V	N	
	NMC27C16	2K x 8	25 V	N	CMOS
	NMC27C16H	2K x 8	25 V	F2	CMOS
	NMC27C16B	2K x 8	12,5 V	I	CMOS
	NMC27C32	4K x 8	25 V	N	CMOS
	NMC27C32H	4K x 8	25 V	F2	CMOS
	NMC27C32B	4K x 8	12,5 V	I;F2	CMOS
	NMC27C64	8K x 8	12,5 V	I;F2	CMOS
	NMC27CP128	16K x 8	12,5 V	I;F2	CMOS
	NMC27C256	32K x 8	12,5 V	I;F2	CMOS
	NMC27C512	64K x 8	12,5 V	I;F2	CMOS
NEC	μPD2716	2K x 8	25 V	N	
	μPD2732	4K x 8	25 V	N	
	μPD2732C	4K x 8	25 V	N	OTP
	μPD2732A	4K x 8	21 V	N	
	μPD2764	8K x 8	21 V	N;I	
	μPD27C64	8K x 8	21 V	N;I	CMOS
	μPD2764C	8K x 8	21 V	N;I	OTP
	μPD27C64C	8K x 8	21 V	N;I	CMOS
	μPD27128	16K x 8	21 V	N;I	OTP
	μPD27128C	16K x 8	21 V	N;I	OTP
	μPD27256	32K x 8	21 V	I	
	μPD27C256	32K x 8	21 V	I	CMOS
Rockwell	R87C32	4K x 8	21 V	N	CMOS
	R87C64	8K x 8	21 V	N	CMOS
	R27C64P	8K x 8	21 V	N	CMOS
SEEQ	2764	8K x 8	21 V	N;I	

	5133	8K x 8	21 V	N, I	
	27128	16K x 8	21 V	N; I	
	5143	16K x 8	21 V	N; I	
	27C256	32K x 8	12,5 V	I	CMOS
SGS/ATES	M2716	2K x 8	25 V	N	
	M2732A	4K x 8	21 V	N	
	M2764	8K x 8	21 V	N; I	
Texas Instruments	TMS2516	2K x 8	25 V	N; F1	
	TMS2532	4K x 8	25 V	N; F1	
	TMS25L32	4K x 8	25 V	N	LP
	TMS2732	4K x 8	25 V	N	
	TMS2732A	4K x 8	21 V	N	
	TMS2564	8K x 8	25 V	N; F1	
	TMS2764	8K x 8	21 V	N; I	
	TMS27128	16K x 8	21 V	N; I	
Thomson CSF	ET2716	2K x 8	25 V	N	
	ETC2716	2K x 8	25 V	N	CMOS
	ETC2732	4K x 8	25 V	N	CMOS
	ET2764	8K x 8	21 V	N	
Toshiba	TMM323	2K x 8	25 V	N	
	TMM2732	4K x 8	25 V	N	
	TMM2764	8K x 8	21 V	N; I	
	TMM2764D1	8K x 8	21 V	N; I	
	TMM27128	16K x 8	21 V	N; I	
	TMM27256	32K x 8	21 V	I	
	TC57256	32K x 8	21 V	I	

Vysvetlivky k poznámkam:

- N normalna programovacia metoda (50 ms cyklus)
- I interaktivne programovanie
- F1 zrychlene programovanie (20 ms cyklus)
- F2 zrychlene programovanie (10 ms cyklus)
- M programovacia metoda MOTOROLA
- LP Low-Power verzie (s malou spotrebou)
- OTP jednorazovo programovatelna pamat
- CMOS pamat CMOS (nizka spotreba)

Ak by bol zaujem o konkretny univerzalny programator, mohol by sa uverejnit. Osobne si vsak myslim, ze z dovodu jednak nedostupnosti suciastok, jednak malej vyuzitosti nema zmysel mat po republike trebars stovky programatorov k Spectru. Stacili by podla mna 3-4 spolahlive miesta pre sluzbu (i za rozumnu uhradu) nam vsetkym Spectristom. (Pripustam, ze aj nespectristom.)

Do prace s pocitacom a s EPROM praje vela chuti, energie a najma uspechov

10.10.1987 Komarno

Csenger Sándor