

UNIVERZALNI PROGRAMATOR ZA PROGRAMIRANJE BIPOLARNIH PROM-OV IN PLA-JEV

F. ŠTRAVS
M. DRUŽOVEC
M. GERKEŠ
V. ŽUMER

UDK: 681.327.28 - 503.55

VISOKA TEHNIŠKA ŠOLA MARIBOR, JUGOSLAVIJA

Programiranje bipolarnih vezij je sorazmerno zahtevno, ker so predpisani programski impulzi dokaj različnih oblik. Postopki programiranja se ne dajo poenotiti, ker uporablja vsak proizvajalec nekoliko drugačno tehnologijo. Zato mora biti programator sposoben generirati več različnih vrst impulzov, ki se naj v odvisnosti od zahtev ustrezno oblikujejo. V članku je dana izvedba univerzalnega in modularno zgrajenega programatorja s programskim generiranjem impulzov. Za različne vrste vezij je potrebno pognati posamezne programske module, ki se izvajajo z mikro-računalnikom družine 6800. Kot zgled dodajmo programiranje PROM-a 74 S 288 tako, da izvaja funkcijo naslovnega krmilnika IDM 29811.

UNIVERSAL PROGRAMMER FOR PROGRAMMING OF BIPOLAR PROMS AND PLAS: The programming of bipolar circuits is a relatively pretentious task since the required pulses have to be shaped differently in accordance with technology differences of the manufacturers and can not be unified. The programmer must be able to generate pulses in accordance with requirements. The article describes an universal and modularly built programming board for pulse generation. For different circuits the programming modules of the 6800 family microcomputer have to be known. As an example the programming of 74 S 288 PROM, which performs the function of IDM 29811 Next-Address Controller, is described.

UVOD

Prednost, ki jo imajo bipolarne vezja pred vezji v MOS tehnologiji, je predvsem v manjših zakasnilnih časih. Za delo v realnem času lahko v nekaterih primerih, kot so n.pr. hitro regulacije, učinkovito uporabimo bipolarne mikroprocesorje. Za uspešno delovanje sistema morajo tudi pomnilniška in ostala logična vezja zadostiti časovnim zahtevam mikroprocesorja, da ne vnašajo prevelikih zakasnitev oziroma, da ne znižujejo frekvenco ure. Bipolarni PROM-i izpolnjujejo vse zahteve; da lahko sodelujejo v takšnem sistemu, saj je njihov zakasnilni čas pod 50 nsec. Uporabljajo se lahko za spominske elemente, za generiranje zapletenih logičnih signalov, za programirano logiko itd. Tako so lahko n.pr. v PROM-ih vpisani algoritmi oz. podatki, ki jih naj ima mikroprocesor na voljo v najkrajšem času. Posebno mesto zavzemajo programirana logična polja - PLA, ki jih odlikuje poleg hitrih odzivnih časov še velika fleksibilnost. Predstavljajo programirano logiko, ki s programiranjem dobi dokončno obliko, ki je pozneje ni mogoče več spremeniti. Programirana logična polja poleg omenjenih lastnosti še močno zmanjšuje število integriranih vezij. S programiranimi logičnimi polji se dajo realizirati različne logične funkcije, pa tudi enostavnejše aritmetične operacije kot so seštevanje ali množenje.

Programiranje bipolarnih vezij je sorazmerno zahtevno, ker so predpisani programski impulzi dokaj različnih oblik. Zaradi nezanesljive dobave in hitrega razvoja se ni dobro nasloniti le na enega proizvajalca in zato je smiselno, da je programator univerzalen. Ker ima vsak proizvajalec svojo tehnologijo, so tudi postopki programiranja, ki jih podajajo proizvajalci, med sabo različni. Zato mora biti programator sposoben generirati več različnih vrst impulzov, ki se naj v odvisnosti od zahtev ustrezno oblikujejo. Zato je najbolj primeren programator s programskim generiranjem impulzov, ki je fleksibilen, glavni del sprememb pa pade na programsko opremo.

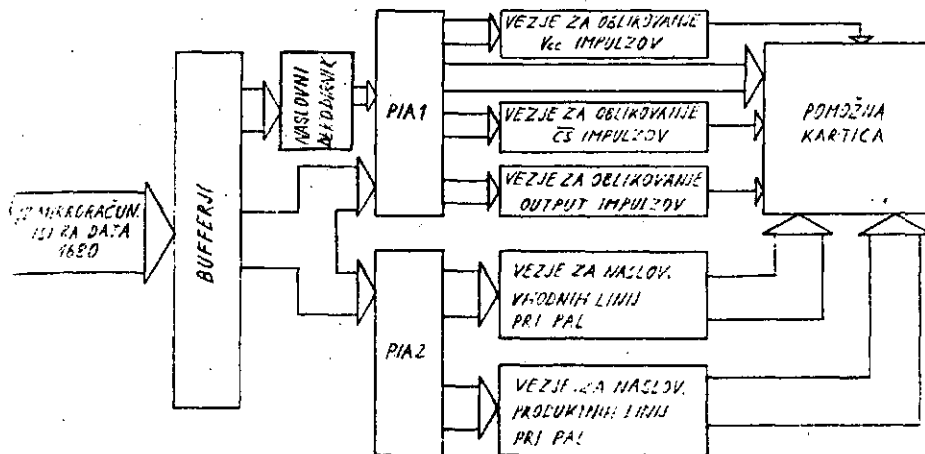
Za vse bipolarne elemente te vrste velja isto načelo programiranja, ki temelji na prežiganju (prekinjanju) specialnih povezav v strukturi integriranega vezja. Pri PROM-u predstavlja pomnilniški del matrika posebnih celic, od katerih vsaka vsebuje po eno povezavo (spoj), ki se lahko prežge ali pa ostane nepoškodovana. Stanje spoja vpliva na logično stanje izhoda. Spoj predstavlja tanka metalna plast, ki je po sredini zožena in ki se pri programiranju zaradi povečanega toka segreje in pregori. Pri PLA-ju pa podobni spoji predstavljajo povezave med logičnimi elementi v polju AND oz. OR vrat.

Pri programiranju se neustrezne povezave prekinajo in ostanejo le tiste, ki jih zahteva željena logična funkcija. Ker se pri prežiganju sprošča toplota, se lahko istočasno programira le en spoj, med posameznimi prežigi pa mora biti zagotovljen čas za hlajenje.

OPIS UNIVERZALNEGA PROGRAMATORJA

Prototip programatorja je v prvi fazi prirejen za programiranje bipolarnih PROM-ov proizvajalcev TEXAS INSTRUMENTS in NATIONAL SEMICONDUCTOR. Med PLA-ji pa se lahko programirajo vezja serije PAL proizvajalca MONOLITHIC MEMORIES in FPLA proizvajalca SIGNETICS. Programator je izveden v modularni izvedbi, tako programska oprema kot vezje. Aparaturni del programatorja predstavlja matična kartica in več manjših pomožnih kartic. Za vsako družino PROM-ov oz. PLA-jev, ki imajo skupne programske lastnosti, je predvidena posebna pomožna kartica z ustreznimi podnožji. S priključitvijo pomožne kartice se avtomatsko vzpostavijo delovni pogoji (napetosti), ki ustrezajo tipu PROM-a oz. PLA. Prav tako je predviden za vsako pomožno kartico svoj podprogram, medtem ko je glavni program skupen. Programator je predviden za priključitev na mikroročunalniški sistem ISKRA DATA 1680 tako, da se matična kartica vstavi v mikroročunalnik, ustrežna pomožna kartica pa se preko konektorja ali podaljška iz ploščatega kabla spoji z matično kartico. Programator se napaja preko zunanje vira 22 V, 1,5 A in 5 V iz mikroročunalnika za napajanje integriranih vezij.

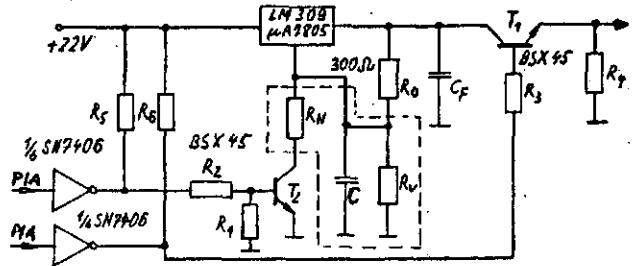
Kot je razvidno iz sheme na sliki 1, sta za krmiljenje programatorja uporabljeni dve PIA-i. Obe sta priključeni preko bufferjev in naslovnega dekodirnika na mikroročunalnik ISKRA DATA 1680. A - stran PIA-e 1 krmili troje vezij, za oblikovanje napajalne napetosti, oblikovanje CS (chip select) impulzov in oblikovanje OUTPUT impulzov, oziroma krmili vezje za naslavljanje produktivnih linij pri PAL. Za PROM-e, ki imajo več kot 8 naslovnih linij, B - stran PIA-e 1 ne zadoštuje več in zato lahko za krmiljenje nadaljnjih 8 naslovnih linij uporabimo B - stran PIA-e 2, pri čemer ustreza linija PBO devetemu bitu. Za naslavljanje vhodnih linij pri PAL je uporabljena PIA 2 v celoti.



Slika 1

VEZJE ZA OBLIKOVANJE IMPULZOV

Pri opazovanju impulzov, ki so predpisani za različna programiranja, lahko ugotovimo, da so vsi skoki napetosti med 0 in 20 V. Ker se pri prežiganju pojavljajo tokovi do 750 mA, mora vezje zadostiti tudi tokovni zahtevi. Na sliki 2 je prikazano uporabljeno vezje, ki omogoča skoke napetosti z možnostjo nastavitve časov vzpona, ki so predpisani. Čas vzpona napetosti je odvisen od vrednosti kondenzatorja C.

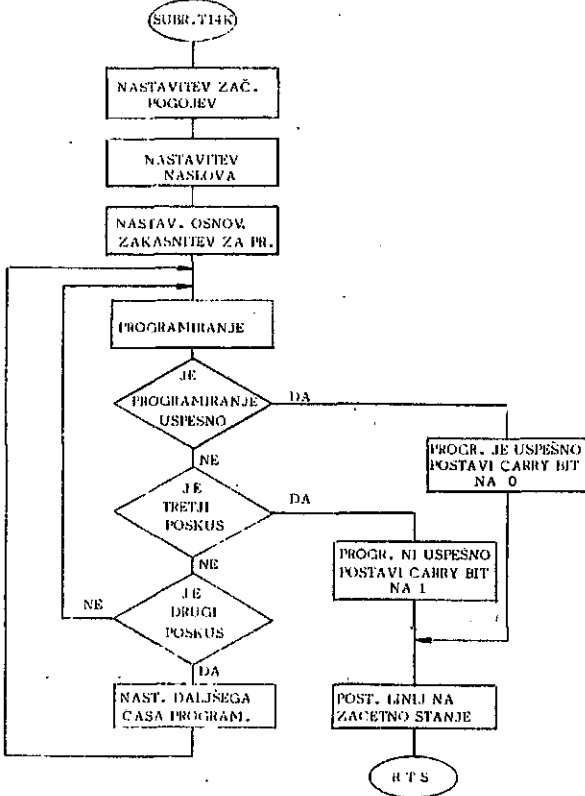


Slika 2

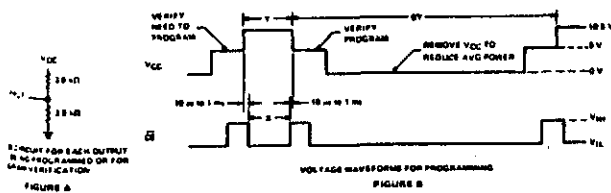
PROGRAMIRANJE PROM-OV (TEXAS INSTRUMENTS)

Programiramo lahko 3 - state PROM-e ali PROM-e z odprtim kolektorjem. Na ustrezni pomožni kartici je predvideno posebno mikrostikalo S_2 , s katerim se nastavi režim za programiranje njih ali drugih. Mikrostikalo S_1 je predvideno za nastavitve izhoda (stolpca), ki naj se programira. Potek programiranja je zamišljen tako, da se najprej sklene prvi kontakt stikala S_1 , s čemer je omogočeno programiranje vseh bitov, ki so dosegljivi na izhodu 0. Nato požene glavni program in mikroročunalnik postopoma naslavlja naslove, kjer se naj vpiše logična 0 in sproti programira. Po vsakem programiranju spoj tudi testira, če je uspešno prežigan. Ko so programirani vsi zahtevani biti v prvem stolpcu, se program ustavi, da lahko sklenemo drugi kontakt na mikrostikalu S_1 . Seveda prvi kontakt sedaj razklenemo in program znova požene. Postopek se ponavlja za vse izhode. Slika 3 prikazuje diagram poteka podprograma. Naslovi spojev, ki se naj prežigejo, so v obliki tabele vpisani v enem izmed RAM-ov mikroročunalnika.

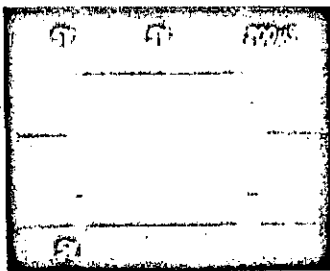
Tabela se s pomočjo programiranja po stolpcih preslika v PROM. Na sliki 4 je časovni diagram predpisanih programskih impulzov, na sliki 5 pa je fotografiran oscilogram impulzov dobljenih iz programatorja.



Slika 3



Slika 4



Slika 5

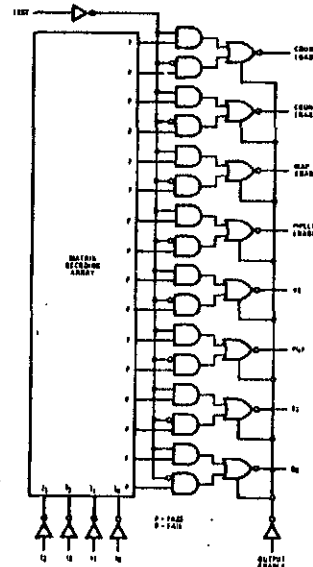
Na sliki 6 je prikazan program za vpisovanje podatkov v PROM 74 S 288. PROM ima 32 8-bitnih besed. Tabela, ki jo vanj preslikamo, je podana na naslovih od 300 - 31F. Na naslovu STL je vpisana številka stolpca (1 - 8), ki se iz tabele preslika v PROM in mora ustrezati številki sklenjenega kontakta na stikalu S₁. Za primer kako lahko PROM uporabimo n.pr. za nadomestitev logičnih vezij, smo vanj vpisali pravilnostno tabelo naslovnega krmilnika IDM 29 811 (glej sliko 7 in tabelo 1). Z ustrezno razmestitvijo stolpcev iz tabele 1 smo celo dosegli, da je razporeditev priključkov na PROM-u enaka ustreznim priključkom na vezju IDM 29 811.

```

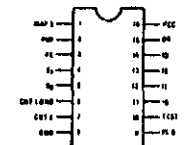
*PROGRAMIRANJE PROM-A (TEXAS
*
0330 TAB EQU $330
0331 TAI EQU $331
0340 STL EQU $340
0200 TI4K EQU $200
0400 ORG $400
0403 CE 0300 LDX #1300
0406 FF 0330 STX TAB
0409 A6 00 ZAC LDX TAB
040B F6 0340 LDAA 0,X
040E 44 LDAB STL
040F 5A PON LSRA
0410 26 FC DECB
0412 24 06 BCC NAP
0414 F6 0331 LDAB TAI
0417 8D 0200 JSR TI4K
041A 7C 0331 NAP INC TAI
041D B6 0331 LDAA TAI
0420 81 20 CMPA #120
0422 26 E2 BNE ZAC
0424 3F SWI
0425 80 FCB $80
0000 END
    
```

Slika 6

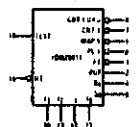
Logic Diagram



Connection Diagram



Logic Symbol



Slika 7

Truth Table

Mnemonic	Function	Pi No.	Inputs													Outputs						
													Next ADDR Source			File		Counter		MAPE	PLE	
			i ₉	i ₂	i ₁	i ₀	TEST	S ₁	S ₀	FE	PUP	LOAD	EN	MAPE	PLE							
JZ	JUMP ZERO		L	L	L	L	L	L	L	L	H	H	H	H	H	H	L	L	H	L	L	
CJS	COND JSB PL		L	L	L	L	H	L	L	L	L	H	H	H	H	L	L	H	H	H	L	L
JMAP	JUMP MAP		L	L	H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H
CJP	COND JUMP PL		L	L	H	H	L	L	L	L	L	H	H	H	H	L	L	H	H	H	L	H
PUSH	PUSH/COND LD CNTR		L	H	L	L	L	L	L	L	L	L	L	H	H	L	L	H	H	H	L	H
JSAP	COND JSB R/PL		L	H	L	H	L	L	L	L	L	H	H	H	H	L	L	H	H	H	L	H
CJV	COND JUMP VECTOR		L	H	L	H	L	L	L	L	L	H	H	H	H	L	L	H	H	H	L	H
JRP	COND JUMP R/PL		L	H	H	H	L	L	L	L	L	H	H	H	H	L	L	H	H	H	L	H
RFCT	REPEAT LOOP, CTR # 0		H	L	L	L	L	L	L	L	H	L	H	L	H	L	H	L	H	L	H	L
RPCT	REPEAT PL, CTR # 0		H	L	L	H	L	L	L	L	H	H	H	H	H	L	H	L	H	L	H	L
CRTM	COND RTM		H	L	H	L	L	L	L	L	L	L	L	H	H	L	L	H	H	H	L	H
CJPP	COND JUMP PL & POP		H	L	H	H	L	L	L	L	L	H	L	H	L	H	L	H	H	H	L	H
LDCT	LD CNTR & CONTINUE		H	H	L	L	L	L	L	L	L	H	H	H	L	H	H	L	H	H	L	H
LOOP	TEST END LOOP		H	H	L	H	L	L	L	L	L	L	L	L	H	H	H	H	H	H	L	H
CONT	CONTINUE		H	H	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	L	H
JP	JUMP PL		H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	L	H
			H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	L	H

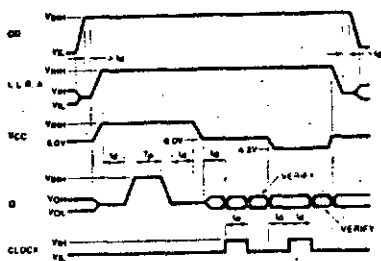
L = Low, H = High

Tabela 1

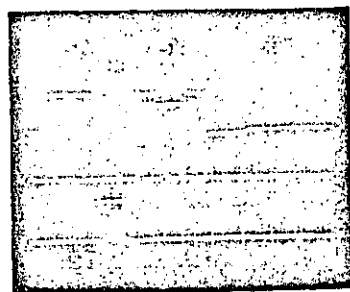
PROGRAMIRANJE PAL-OV (MONOLITHIC MEMORIES)

S tem programatorjem je možno programirati vseh 15 tipov PAL-ov MONOLITHIC MEMORIES. Pomožna kartica vsebuje dve podnožji z 20 priključki, stikalo S₁ in potrebne elemente za predhodno nastavitve pravih napetosti. Ker poteka v prvi fazi programiranje spojev na produktih linijah od 0 do 31, mora biti PAL vstavljen v podnožje 1 na pomožni kartici. Odvisno od tipa PAL je potrebno nastaviti položaj stikala S₁. V primeru programiranja takšnega PAL-a, ki daje v aktivnem stanju visok nivo (1) na izhodu, je potrebno postaviti S₁ v položaj ACTIVE HIGH. Obratno velja za PAL-e z nizkim aktivnim nivojem položaj ACTIVE LOW. Programiranje je zamišljeno tako, da so v enem izmed RAM-ov vpisane vse koordinate spojev (številka vhodne linije, številka produkne linije), ki naj se programirajo. Ker so koordinate v RAM-u podane binarno, je v podprogramu narejen programski dekodirnik, ki postavi vse naslovne linije na pravilne nivoje in ugotovi izhod, s pomočjo katerega se bo spoj prežigal. Ko požene glavni program, ta prečita koordinate za prvi spoj in skoči v podprogram, ki opravi programiranje in testira prekinitev spoja. Nato glavni program prečita nove koordinate in postopek se ponovi. Postopek se ponavlja za vse spoje na produktih linijah od 0 do 31, ki so predvideni za prežiganje. Po prežigu zadnjega spoja se program ustavi, da lahko PAL vstavimo v podnožje 2 in s tem preide programiranje v drugo fazo.

V drugi fazi je možno programirati vse spoje na produktih linijah od 32 do 63. Znova požene glavni program, ki postopoma čita nove koordinate spojev in s pomočjo podprograma opravi programiranje. Ko se prežgejo vsi potrebni spoji, se program ustavi in programiranje je končano. Slika 8 prikazuje časovni diagram predpisanih impulzov, na sliki 9 pa je posnet oscilogram impulzov, dobjenih iz programatorja.



Slika 8



Slika 9

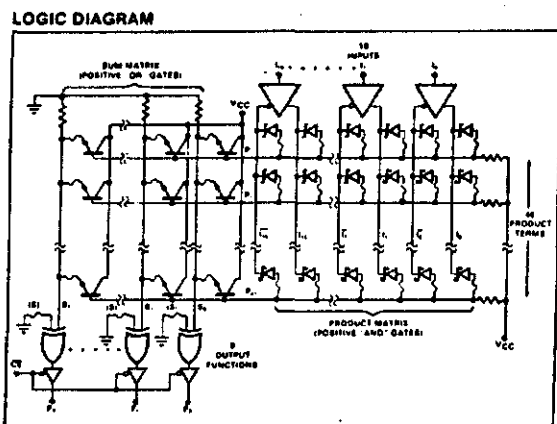
PROGRAMIRANJE PPLA (SIGNETICS)

Programiranje poteka v treh fazah in sicer v naslednjem zaporedju:

1. Programiranje izhodne polaritete (aktiven nivo izhoda)
2. Programiranje produktih termov
3. Programiranje OR termov.

Za vsako fazo postopka je predvideno svoje podnožje na pomožni kartici, ker vsaka faza zahteva svojo obliko

programskih impulzov. Za vsako fazo je predviden svoj podprogram. Na sliki 10 je prikazana notranja povezava FPLA.



Slika 10

Postopek programiranja izhodne polaritete se prične tako, da se na stikalu S_1 sklene eden izmed kontaktov, ki omogoči dostop programskim impulzom na ustrezen izhod. Nato požene program, ki prežge spoj S na ustreznem izhodu. Postopek se ponovi na vseh tistih izhodi, kjer funkcija to zahteva.

Pri programiranju produktnih termov postopamo tako, da najprej sklenemo prvi kontakt stikala S_2 , s čemer je omogočeno prežiganje vseh spojev vezañih na vhod I_0 . Nato požene glavni program, ki prečita ustrezen naslov produktnega terma z informacijo o programiranju spojev. Informacija o programiranju spojev pove, kateri od spojev I_1 ali \bar{I}_1 se naj prežge oziroma pove, da se naj prežgeta oba spoja, če funkcija zahteva, da vhod I_1 nima vpliva na ta produktni term. Glavni program zãtem s pomočjo podprograma opravi programiranje enega ali obeh spojev. Nato glavni program prečita naslednji naslov produktnega terma in spet skoči v podprogram, ki opravi ustrezno programiranje. Postopek se ponovi še za vse ostale produktne terme. Ko so prežgani vsi ustrezni spoji v stolpcih I_1 in \bar{I}_1 , se program ustavi, da lahko sklenemo drugi kontakt stikala S_2 . S tem je omogočeno prežiganje spojev v stolpcih I_1 in \bar{I}_1 , ki določajo vpliv vhoda I_1 na posamezni produktni term. Nato znova požene glavni program in postopek se ponovi. Postopek je enak za vseh 16 vhodnih spremenljivk.

Pri programiranju OR termov postopamo tako, da najprej sklenemo prvi kontakt stikala S_1 , s čemer je omogočeno programiranje vseh spojev na OR termu za izhod F_0 . Nato požene glavni program, ki postopoma nastavlja naslove ustreznih termov in s pomočjo podprograma prežiga spoje. Po končanem programiranju prvega OR terma se program ustavi, da lahko sklenemo drugi kontakt na stikalu S_1 , s čemer je omogočeno programiranje drugega OR terma. Program znova požene in postopek se ponovi. Postopek je enak za vseh 8 OR termov.

ZAKLJUČEK

Programator je narejen tako, da se lahko priredi za programiranje še ostalih PROM-ov oz. PLA-jev, ki v tej fazi niso zajeti. V principu je mogoče programiranje katerekoli bipolarnega elementa, ki ima možnost programiranja. Matična kartica ostane v vseh primerih enaka, pomožne kartice pa se dodatno naredijo. Prav tako je potrebno za vsako novo kartico narediti poseben podprogram, medtem ko je glavni program lahko za večino primerov skupen.

LITERATURA

1. Bipolar LSI Databook, Monolithic Memories, 1979
2. PAL - Programable Array Logic Handbook, Monolithic Memories, 1978
3. TTL Data Book for Design Engineers, TEXAS INSTRUMENTS, 1977
4. Supplement to the TTL Data Book, TEXAS INSTRUMENTS
5. Memory Databook, National Semiconductor, 1977
6. IDM 2900 Family Microprocessor Databook, National Semiconductor, 1978
7. Memory products, MOTOROLA 1979
8. Bipolar and MOS microprocessors, SIGNETICS, 1978
9. Logic TTL, SIGNETICS, 1978
10. Bipolar microprocessor family 2900, AMD
11. M 6820 Data Sheet, MOTOROLA
12. Linear integrated circuits, FAIRCHILD, 1976
13. Katalog ISKRA DATA 1680