

KOMPIUTERIŲ ELEMENTAI IR ARCHITEKTŪRA

Kompiuterių elementų dalis

Pranas Kanapeckas

Nagrinējamos temos

Programuojamos loginės struktūros:

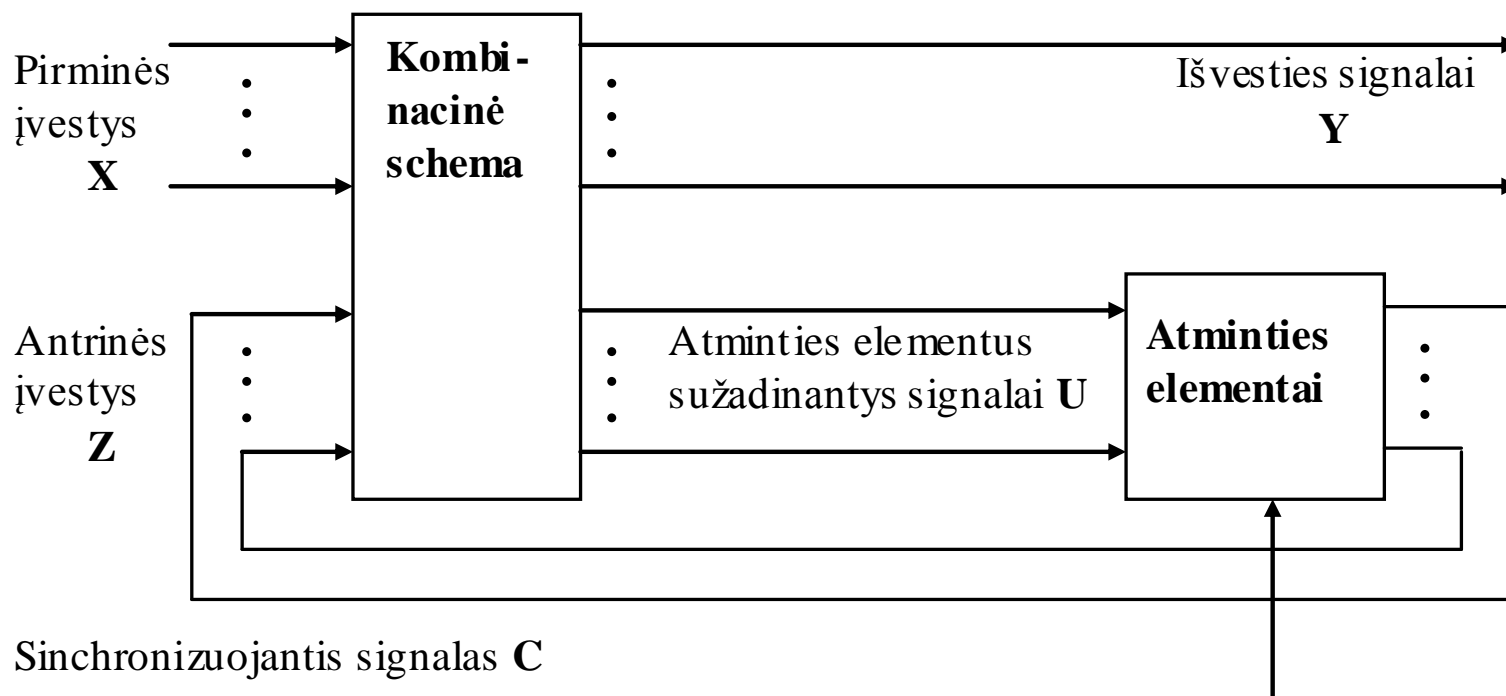
- programuojamos loginės integrinės schemas;
- lanksčios programuojamos ventilinės matricos.

Atmintinės:

- laisvai išrenkamos atmintinės (RAM, SRAM, DRAM);
- pastoviosios atmintinės (ROM, PROM, EPROM, EEPROM).

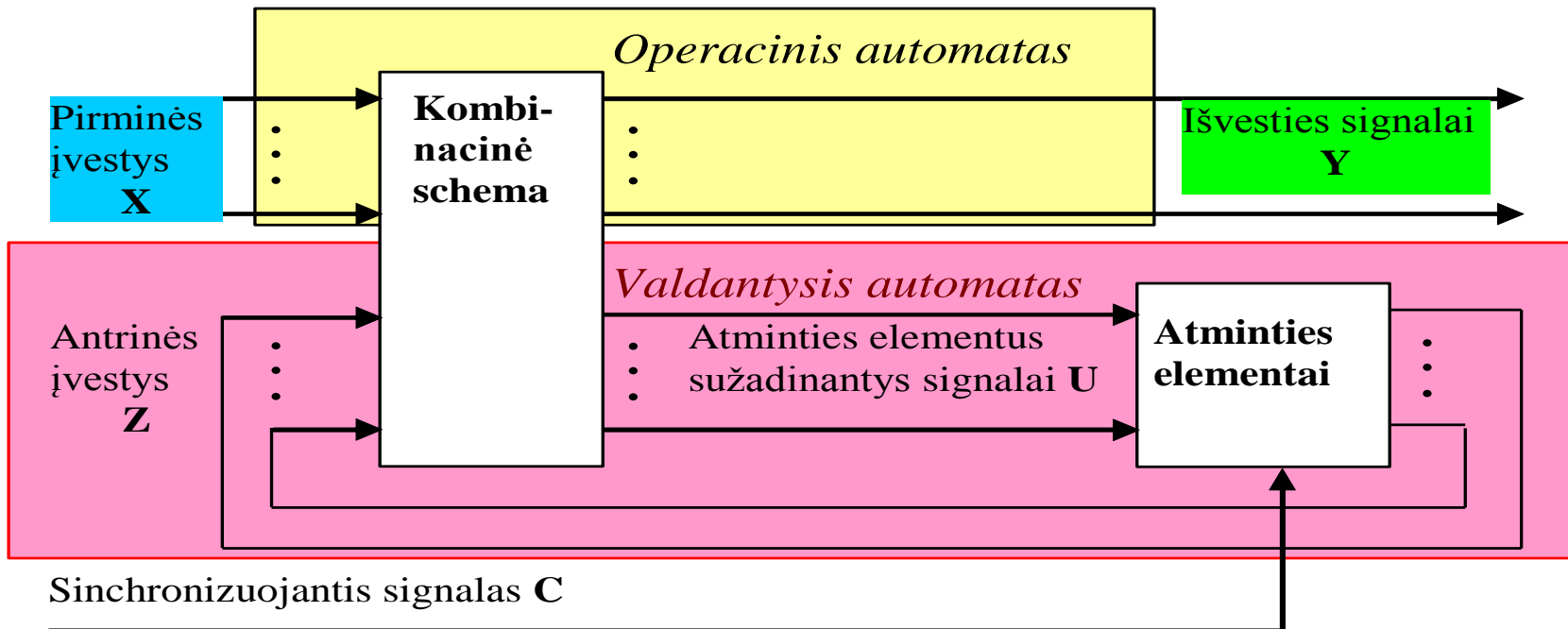
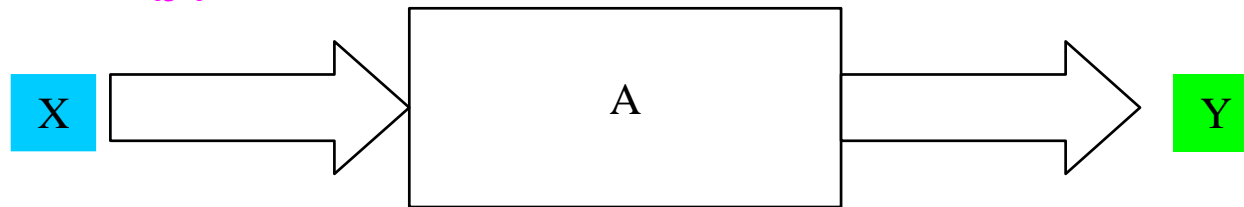
Sinchroninės NS struktūra

- Sinchroninė nuosekloji schema sudaryta iš kombinacinės schemos ir atminties elementų.
- Sinchronizuojantys signalai C paduodami į atminties elementus.



Baigtiniai automatai

- Baigtinis automatas tai *nuosekli loginė schema*
- Kompiuteryje įprasta baigtinį automatą skirstyti į *operacinį* ir *valdantįjį* automatus



Mikroprograma

- Kompiuteryje vykdomą programą sudaro vykdomų komandų seka
- Komandų vykdymas aprašomas algoritmu, kuriame nurodytos elementarios operacijos, vadinamos mikrooperacijomis, jų vykdymo ir valdymo signalų tikrinimo tvarka.
- Aprašytas komandų vykdymo algoritmas vadinamas mikroprograma.

Daugyba tiesioginiame kode

- Raskime skaičių $A=0,1101$ ir $B=0,1010$ sandaugą, daugybą pradedant nuo jauniausių daugiklio skilčių.

$$\begin{array}{r} \times 0,1101 \quad \text{dauginamasis} \\ 0,1010 \quad \text{daugiklis} \\ \hline 0000 \\ 1101 \\ 0000 \\ 1101 \\ \hline 0,10000010 \quad \text{sandauga} \end{array}$$

dalinės sandaugos

Daugyba tiesioginiame kode

- Raskime skaičių $A=0,1101$ ir $B=0,1010$ sandaugą daugybą pradedant nuo aukščiausių daugiklio skilčių.

0,1101	dauginamasis
0,1010	daugiklis
<hr/>	
1101	
0000	} dalinės sandaugos
1101	
0000	
<hr/>	
0,10000010	sandauga

Pavyzdys

- Tegu operandai A ir B yra teigiami: **A = 0.110**, o **B = 0.101**.
- Turime 6 skilčių kombinacinį sumatorių, 6 skilčių postūmio registrą, 6 skilčių lygiagretų registrą ir 5 skilčių postūmio registrą, d triggerį saugoti rezultato ženklą.
- Operandai paimami iš 8 skilčių magistralės DM(8).
- Daugybą atliksime dauginamąjį stumiant nuo žemiausių skilčių.

Daugyba stumiant dauginamąjį (nuo žemiausių skilčių)

A

C

B

0.000110

0.000000

0.101

+0.000110

0,001100

0.000110

0.101

+0.000000

0.011000

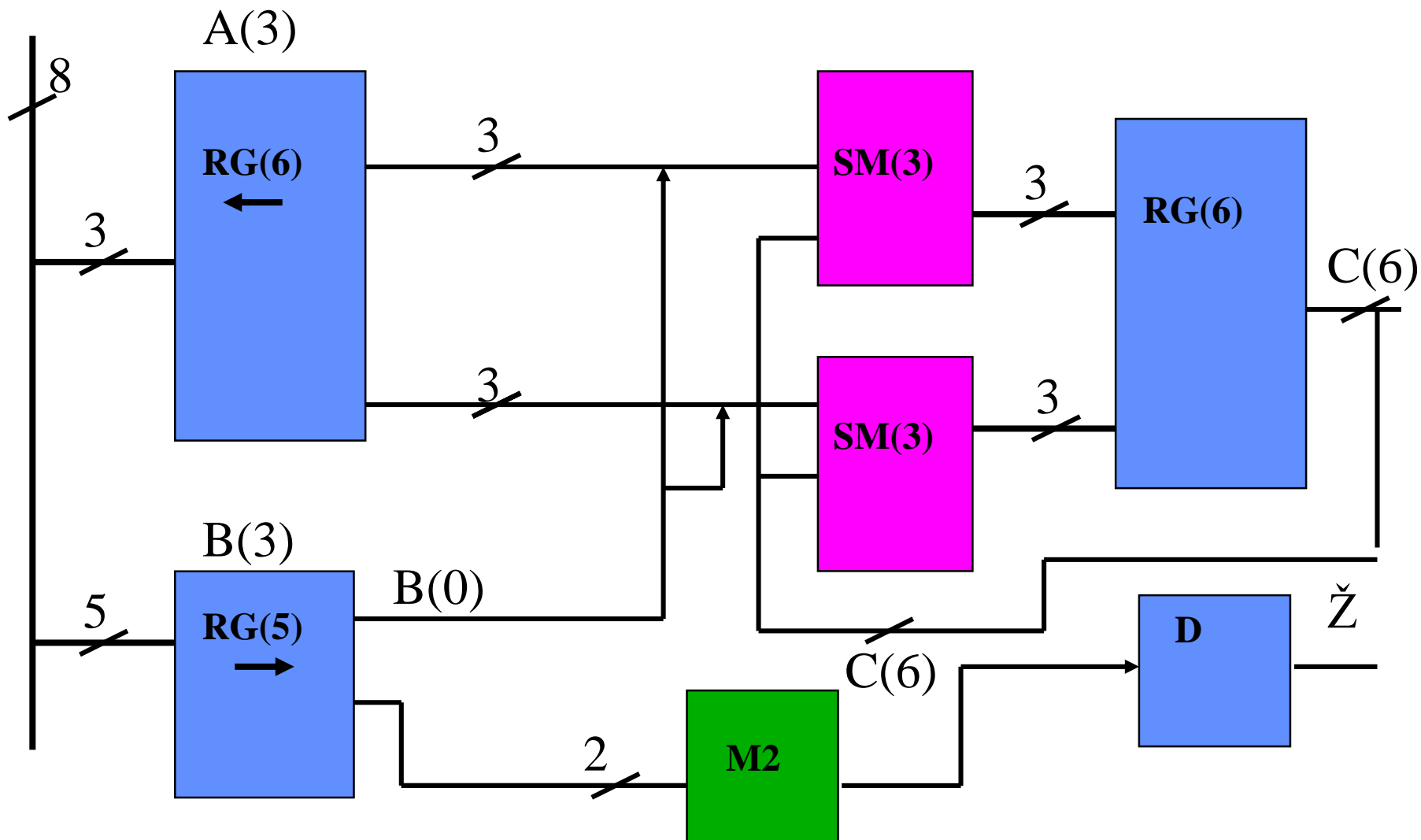
0.000110

0.101

+0.011000

0.011110

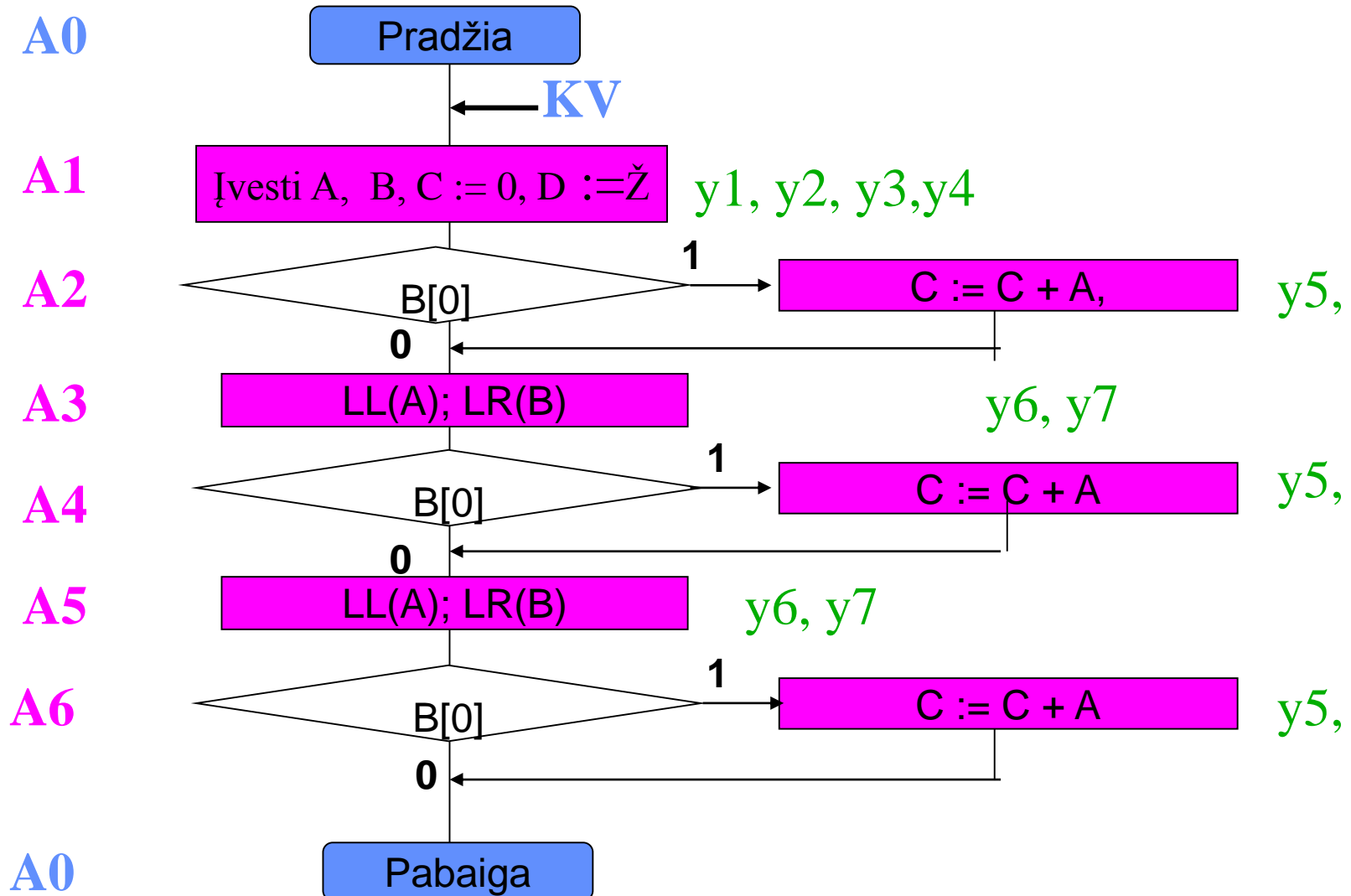
Parenkame operacinio automato struktūrą



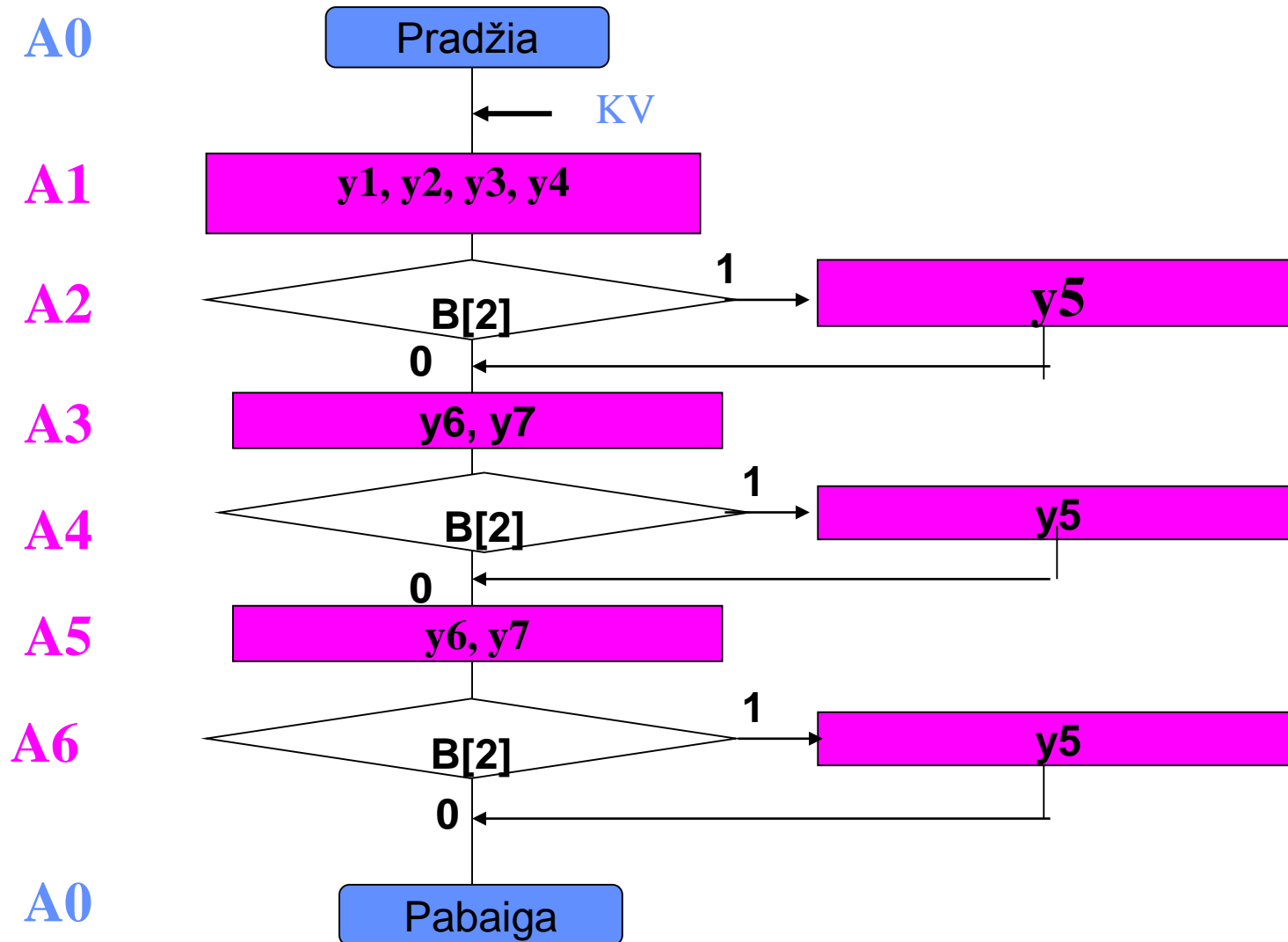
Operacinio automato projektavimas

- Lygiagrečiai su mikroprogramų sudarymu projektuojama operacinio automato struktūra.
- Komandos vykdymo algoritmas skirstomas į mikrooperacijas (veiksnius atliekamus vieno ciklo metu) taip, kad sudaryta struktūra realizuotų algoritmą.
- Signalas **KV** leidžia komandą vykdyti.

Dvejtainės daugybos algoritmas



Dvejtainės daugybos mikroprograma

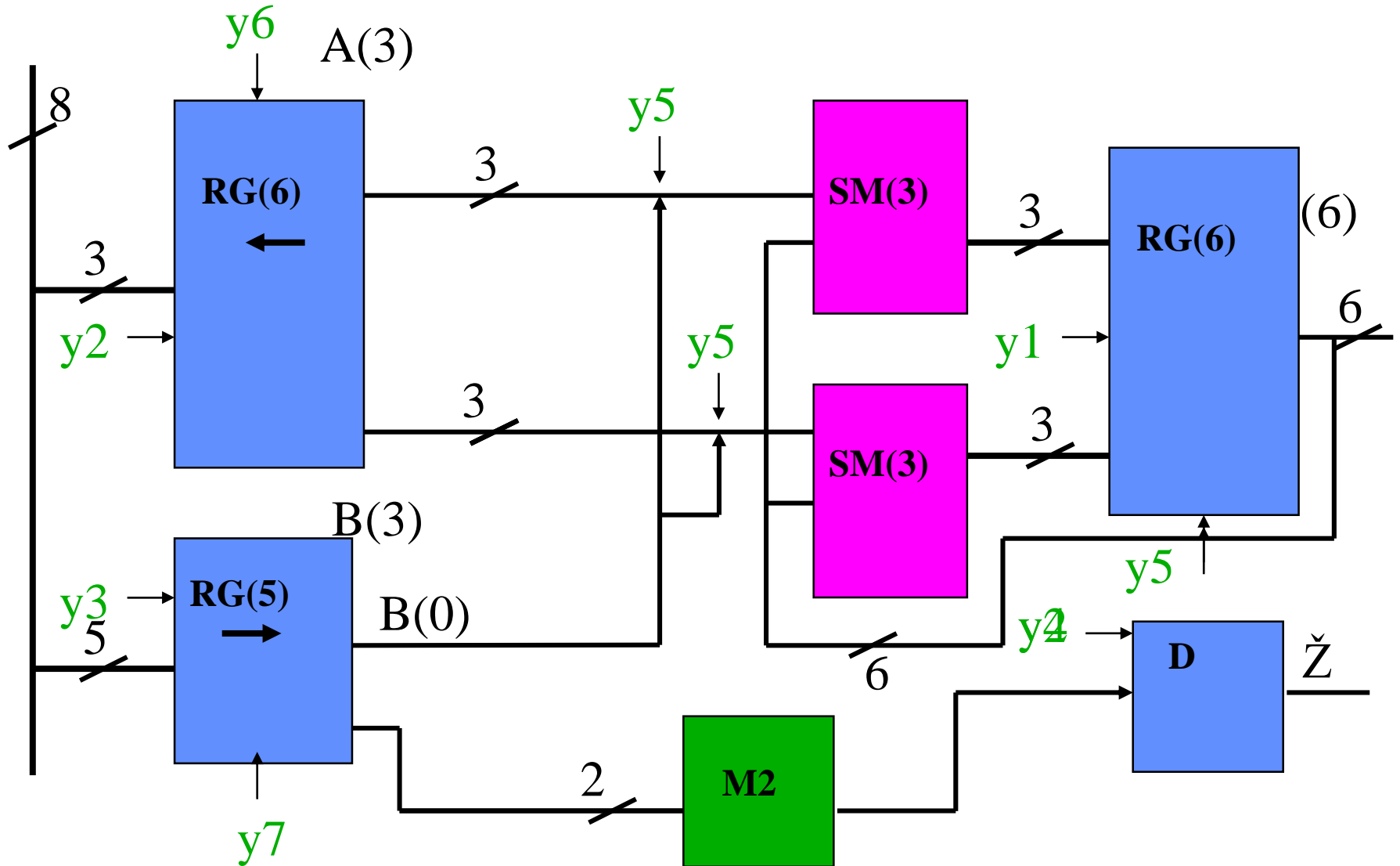


Mikrooperacijos

Kiekvieną veiksmą turi
atitikti viena ar kelios
mikrooperacijos
(MO):

Veiksmai	MO
$C(6) := 0$	y_1
$A(6) := DM$	y_2
$B(5) := DM$	y_3
$\check{Z} := \check{Z}_A \oplus \check{Z}_B$	y_4
$C(6) := C(6) + A(6)$	y_5
$LL(A)$	y_6
$LR(B)$	y_7

Operacinio automato struktūra



Valdančio automato projektavimas

- Sudarome baigtinio automato veikimo lentelę
- Priimkime, kad pradžioje automatas randasi **A0** būsenoje, o pasibaigus komandos vykdymui pereiname į pradinę būseną **A0**
- Būsenas koduosime dvejetainiu kodu
- Naudosime 3 D trigerius
- Projektuojame Muro automata

Automato perėjimų lentelė

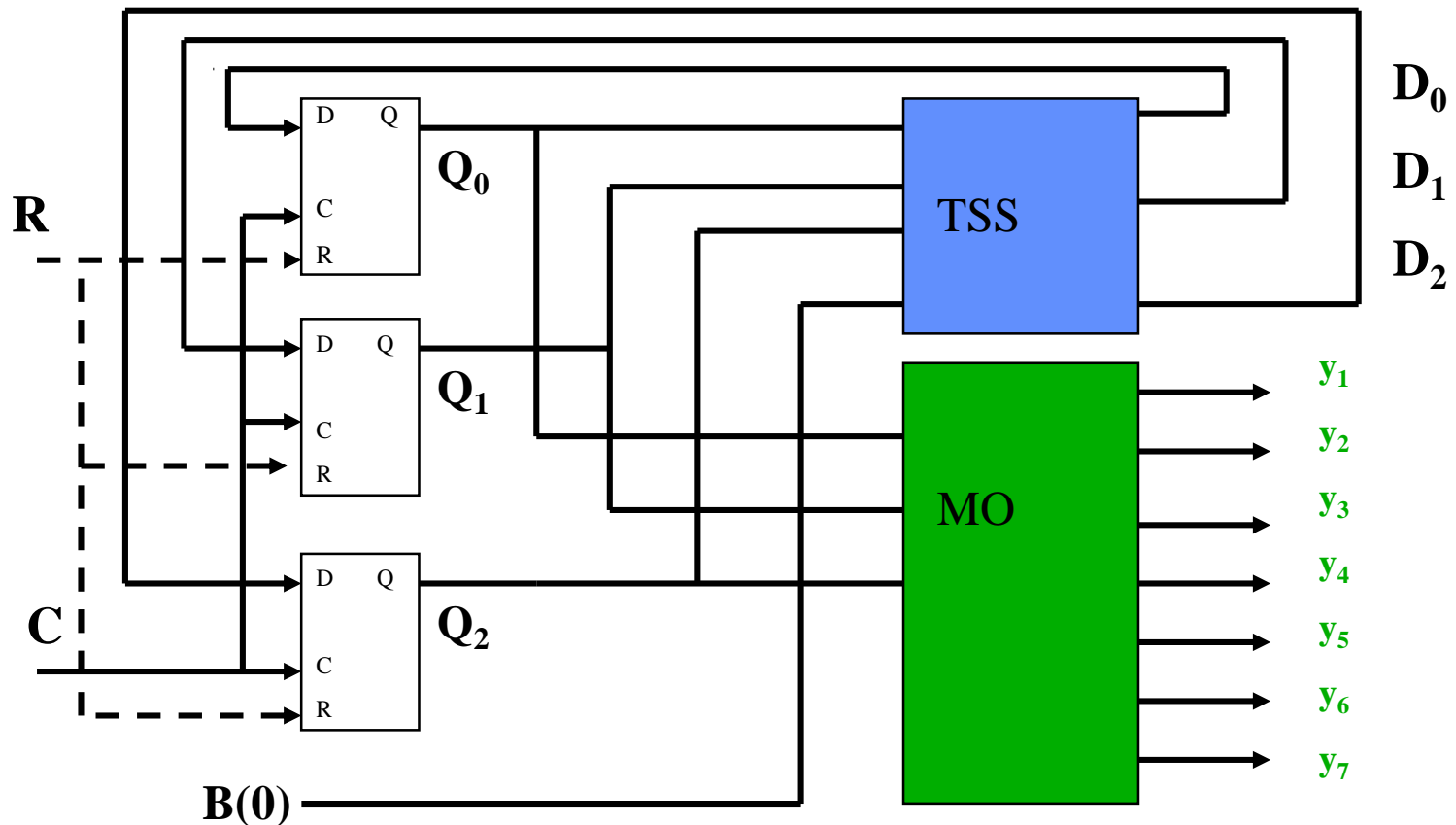
EB	sąlygos	signalai	SB	TSS	
$Q_2 Q_1 Q_0$	B(0) KV	y	$Q_2 Q_1 Q_0$	$D_2 D_1 D_0$	
0 0 0	-- 1		0 0 1	0 0 1	
0 0 1	1	$y_1 y_2 y_3 y_4$	0 1 0	0 1 0	
0 0 1	0	$y_1 y_2 y_3 y_4$	0 1 1	0 1 1	
0 1 0	--	y_5	0 1 1	0 1 1	
0 1 1	1	$y_6 y_7$	1 0 0	1 0 0	
0 1 1	0	$y_6 y_7$	1 0 1	1 0 1	
1 0 0	--	y_5	1 0 1	1 0 1	
1 0 1	1	$y_6 y_7$	1 1 0	1 1 0	
1 0 1	0	$y_6 y_7$	0 0 0	0 0 0	
1 1 0	--	y_5	0 0 0	0 0 0	

Loginės lygtys

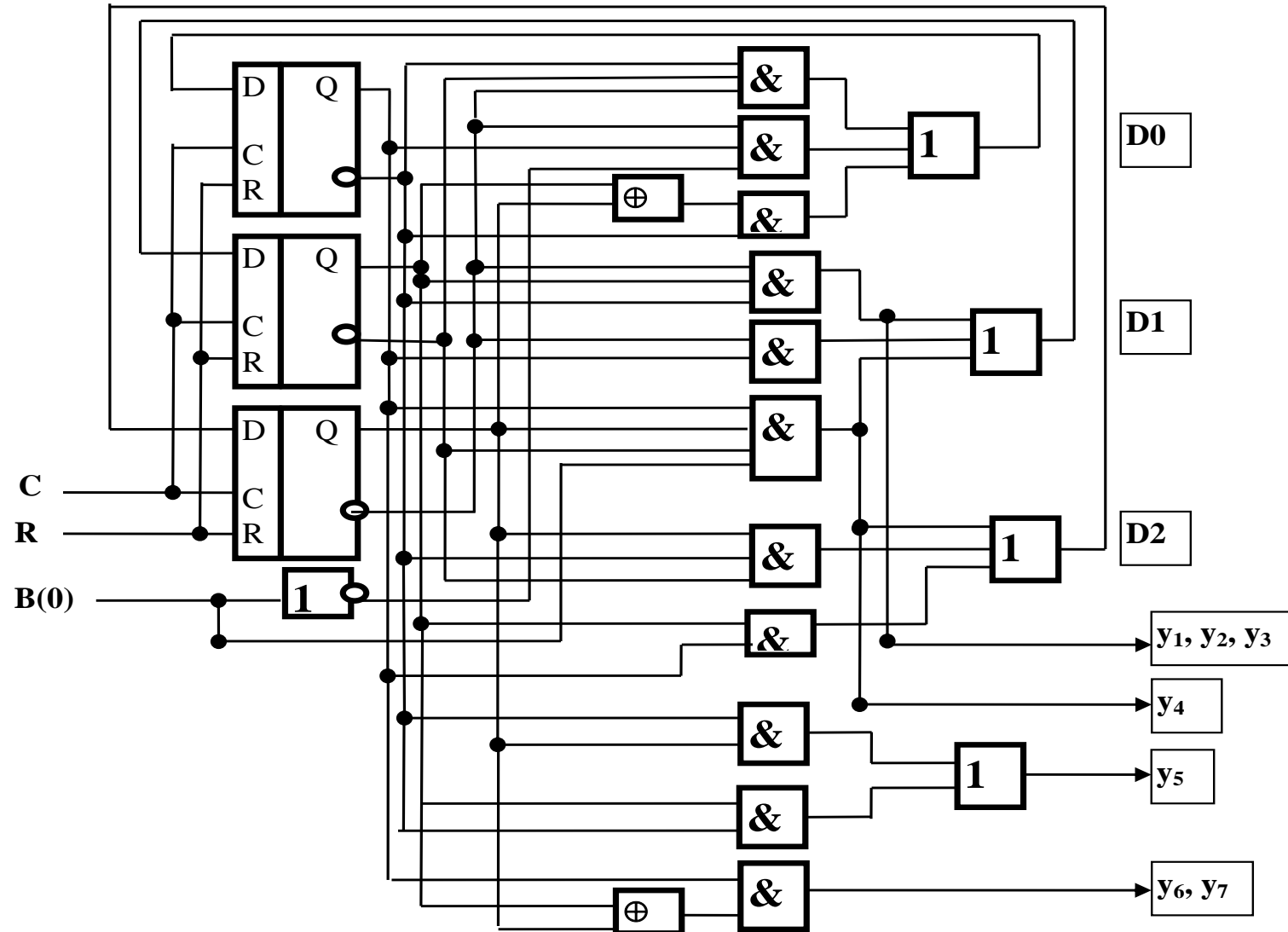
- Pasinaudoję Perėjimų lentelę gauname logines lygtis:
- $D_0 = \overline{Q_2} Q_1 Q_0 KV + Q_2 Q_0 B(0) + Q_0 (Q_2 \oplus Q_1)$
- $D_1 = \overline{Q_2} Q_1 Q_0 + Q_2 Q_1 Q_0 + Q_2 Q_1 Q_0 B(0)$
- $D_2 = \overline{Q_1} Q_0 + Q_2 Q_1 \overline{Q_0} + \overline{Q_2} Q_1 Q_0 B(0)$
- $y_1 = \overline{y_2} = y_3 = \overline{Q_2} \overline{Q_1} \overline{Q_0}$
- $y_4 = \overline{Q_2} \overline{Q_1} Q_0$
- $y_5 = \overline{Q_2} Q_0 + \overline{Q_1} Q_0$
- $y_6 = y_7 = Q_0 (Q_2 \oplus Q_1)$

Valdančio automato struktūra

- Valdantį automata sudarys 3 D trigeriai ir kombinacinė schema



Valdančio automato principinė schema

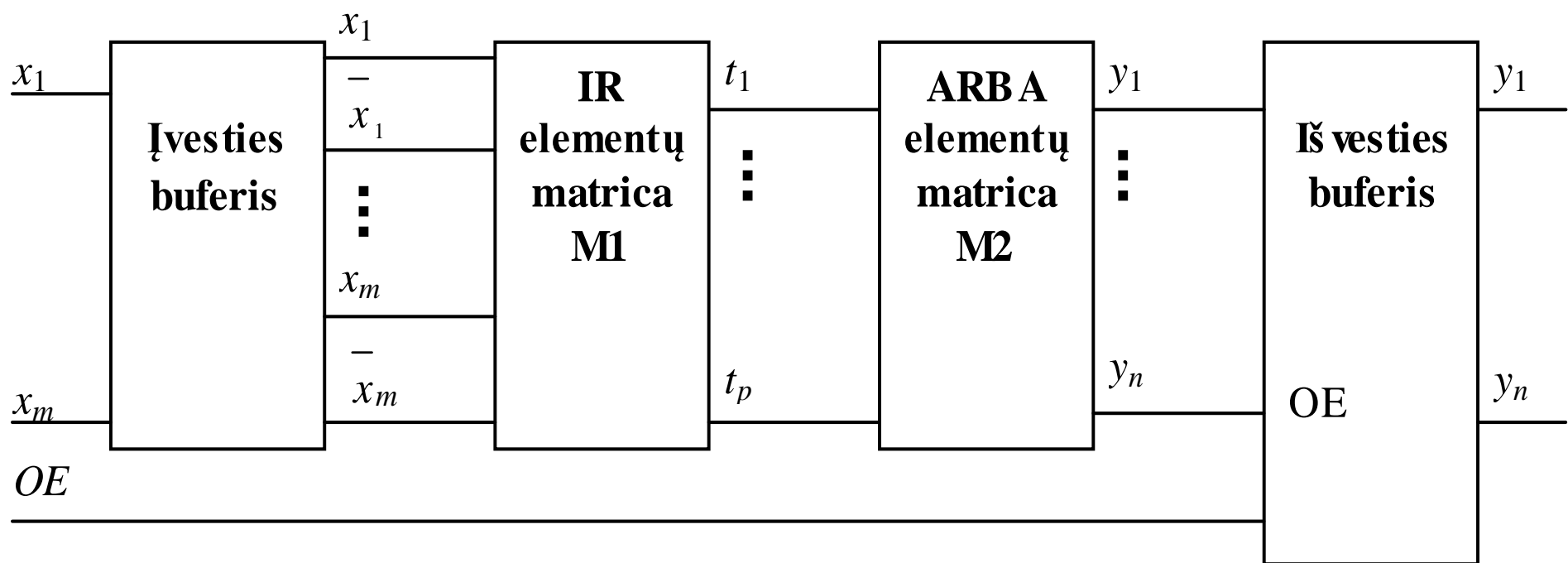


Programuojamos loginės schemos

- Skaitmeninė aparatūra pagrindinai sudaryta iš standartinių didelės integracijos schemų (DIS) ar superdidelės integracijos schemų (SDIS).
- Nestandartinės sistemų dalys suprojektuotos panaudojant MIS arba VIS sudėtingos, brangios, mažas jų patikimumas bei veikimo sparta.
- Projektuoti specializuotas DIS/SDIS yra per daug brangu.
- Sukurtos programuojamos loginės matricos PLM (PLA angl.), programuojama matricinė logika PML (PAL angl.), vadinamos programuojamomis loginėmis schemomis PLS (PLD angl.), ir baziniai matriciniai kristalai BMK, kartais vadinami ventilių matrica VM (GA angl.).

Programuojamos loginės matricos PLM

- PLM (1970) – vienos iš perspektyviausių DIS su reguliariomis struktūromis. Jos sudarytos iš programuojamų matricų, IR ir ARBA loginių elementų, o taip pat įeina įvesties ir išvesties buferių



Programuojamos loginės matricos PLM

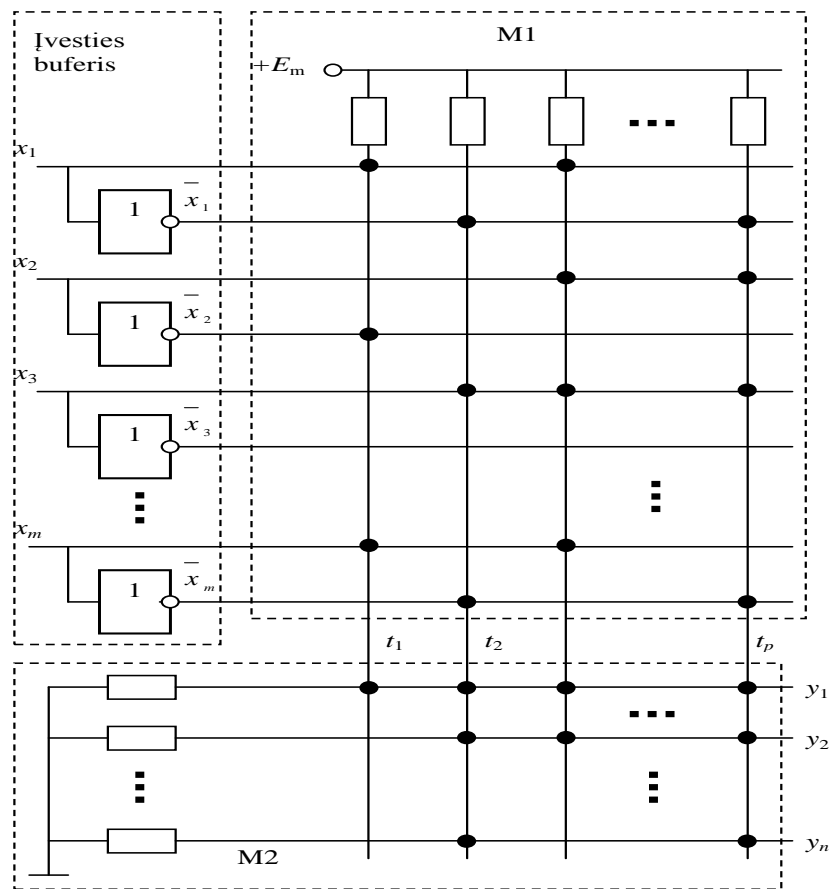
- Projektuojant PLM išvesties signalai aprašomi Bulio funkcijų sistema:
- Funkcijų sistemą patogiau pavaizduoti lentele

$$\begin{cases} y_1 = f_1(x_1, x_2, \dots, x_m), \\ y_2 = f_2(x_1, x_2, \dots, x_m), \\ \text{-----} \\ y_n = f_n(x_1, x_2, \dots, x_m). \end{cases}$$

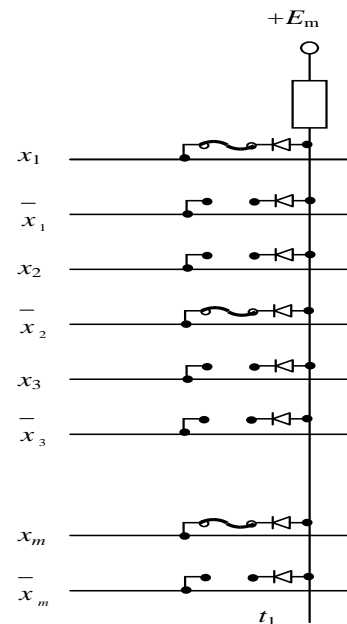
Argumentai $x_1 \ x_2 \ x_3 \dots x_m$	Termai (kubai)	Funkcijos $y_1 \ y_2 \ y_3 \dots y_n$
0 1 x ... 1	t₁	1 0 1 ... 0
1 x 1 ... x	t₂	1 1 0 ... 0
1 0 0 ... 1	t₃	0 1 0 ... 1
.....
0 1 1 ... 0	t_p	0 1 0 ... 1

PLM struktūra

- PLM sudaro:
- parafazinis buferis B;
- elementų IR aibė (M1);
- elementu ARBA aibė (M2).
- Ryšiai sudaryti iš tirpių takelių jungčių ir diodų.

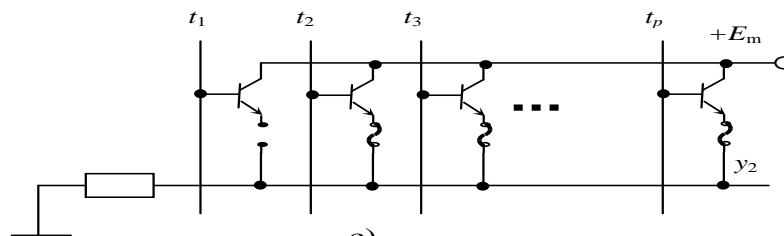


a)

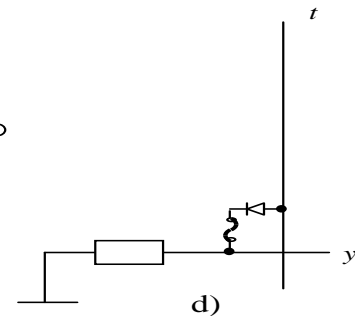


b)

Tirpi jungtis
Ištirpinta tirpi jungtis



c)



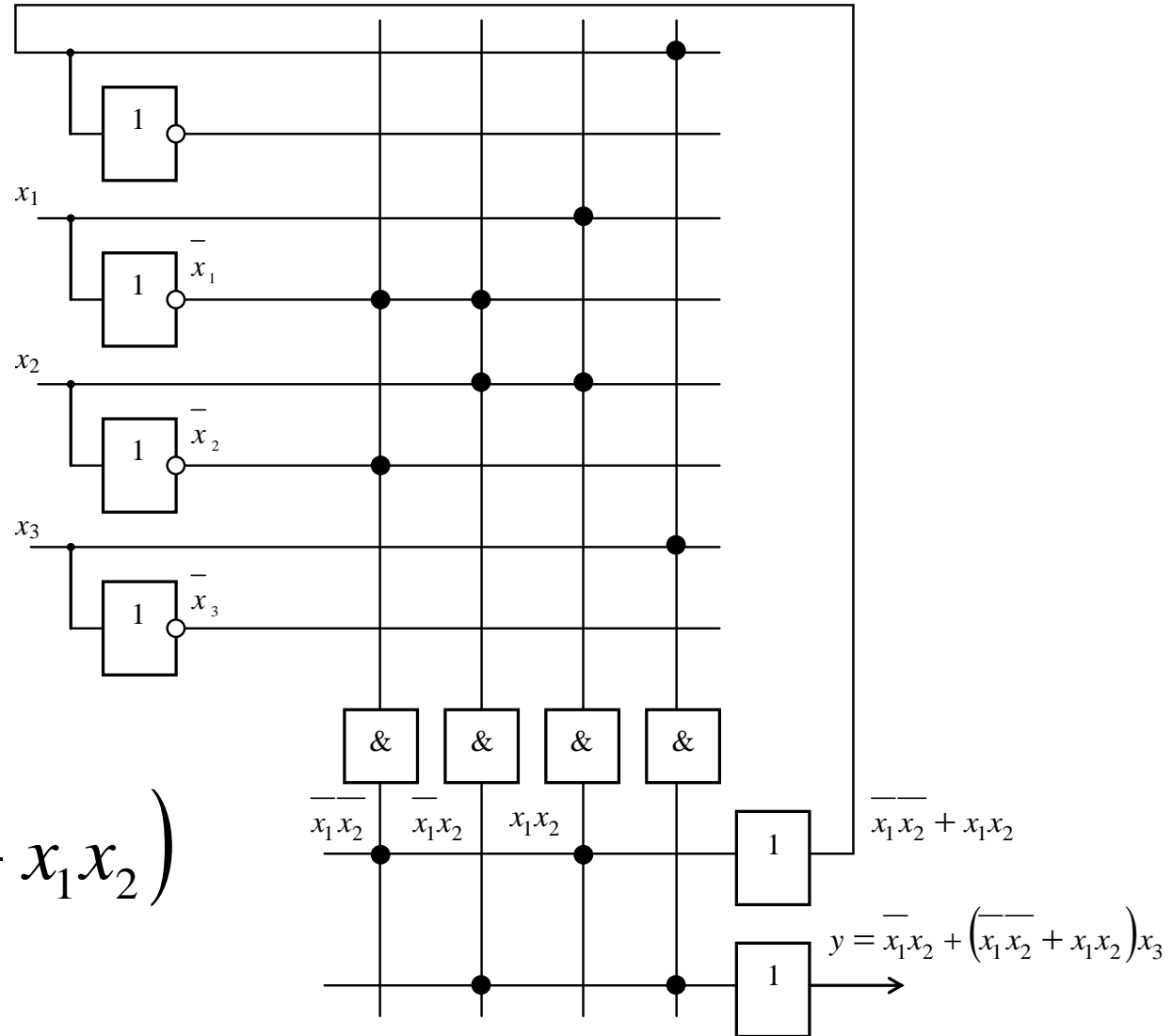
d)

PLM programavimas

- Projektuojant PLM stengiamasi minimizuoti duotos funkcijų sistemos termų kiekį.
- PLM programuoja vartotojas ištirpindamas tirpias jungtis.
- Dažniausiai pateikiama informacija apie tirpias jungtis programavimo įrenginiui reikiama forma.
- Dažnai naudojami tokie simboliai: H – tiesioginė kintamojo reikšmė (M1); L – inversinė kintamojo reikšmė (M1); „-“ – kintamasis nenaudojamas (M1); A – duota IR funkcija naudojama (M2); „●“ – duota IR funkcija nenaudojama (M2).
- Logines funkcijas patogiau pateikti disjunkcine normaline forma.

PLM su grįžtamais ryšiais

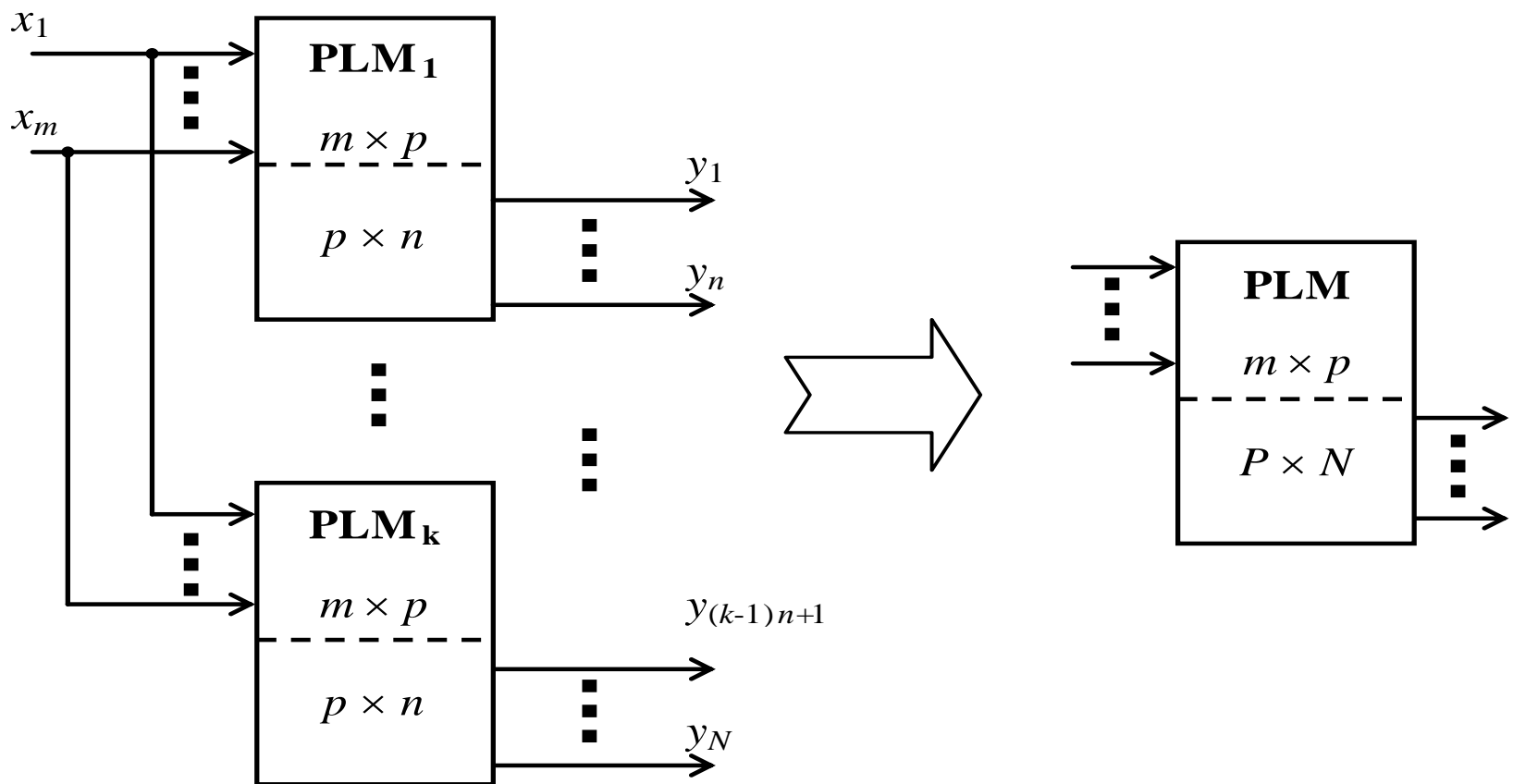
- Turint PLM su grįžtamais ryšiais patogiau realizuoti funkciją, pateiktą su skliaustais.



$$y = \overline{x_1}x_2 + (\overline{\overline{x_1}x_2} + x_1x_2)x_3$$

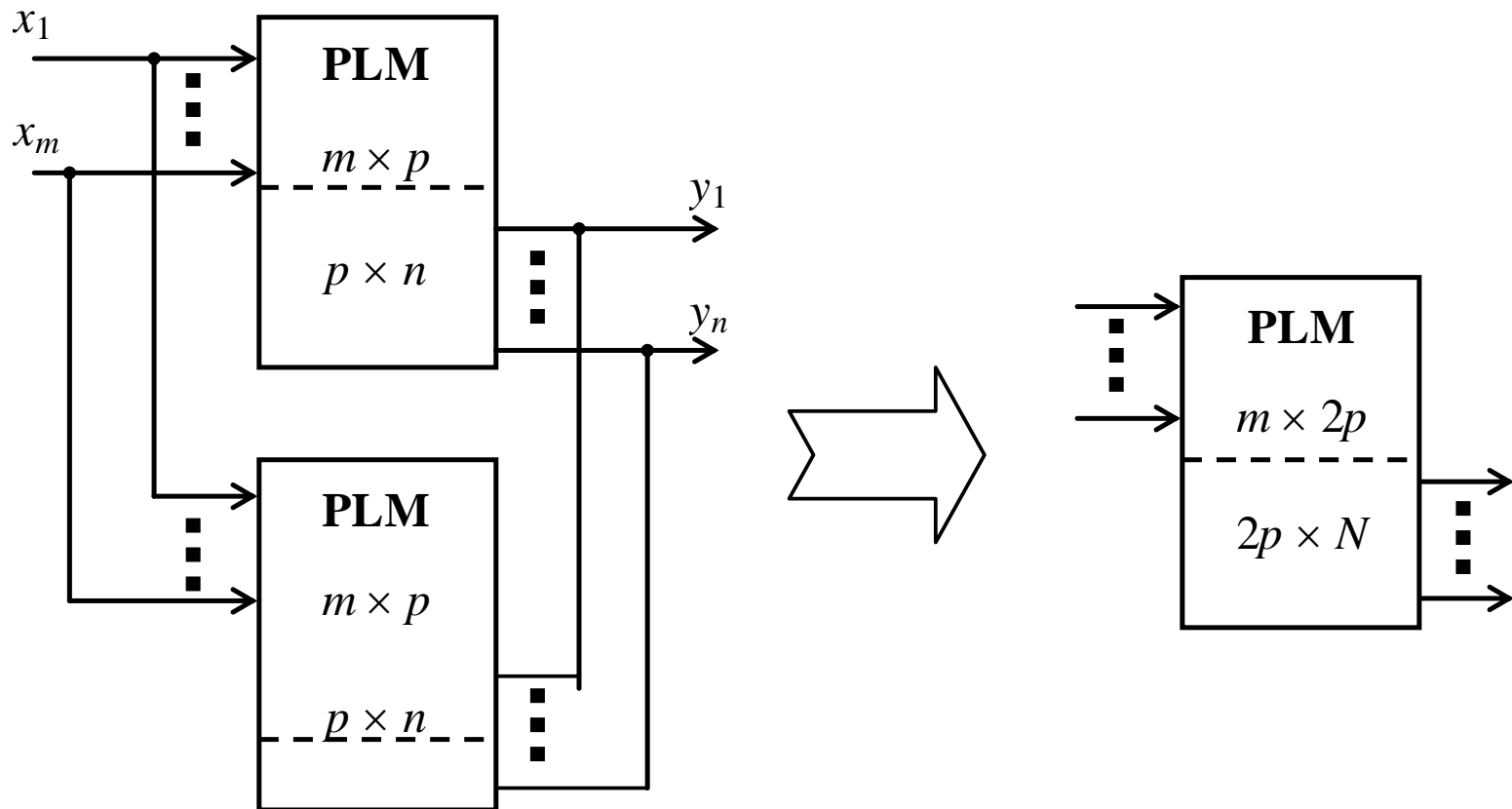
PLM išvesčių kiekio išplėtimas

- Kai formuojamų išvesties signalų kiekis N sistemoje viršija PLM išvesčių kiekį n ($N > n$), panaudojama $k \geq N/n$ PLM schemų, kurių įvestys sujungtos lygiagrečiai.



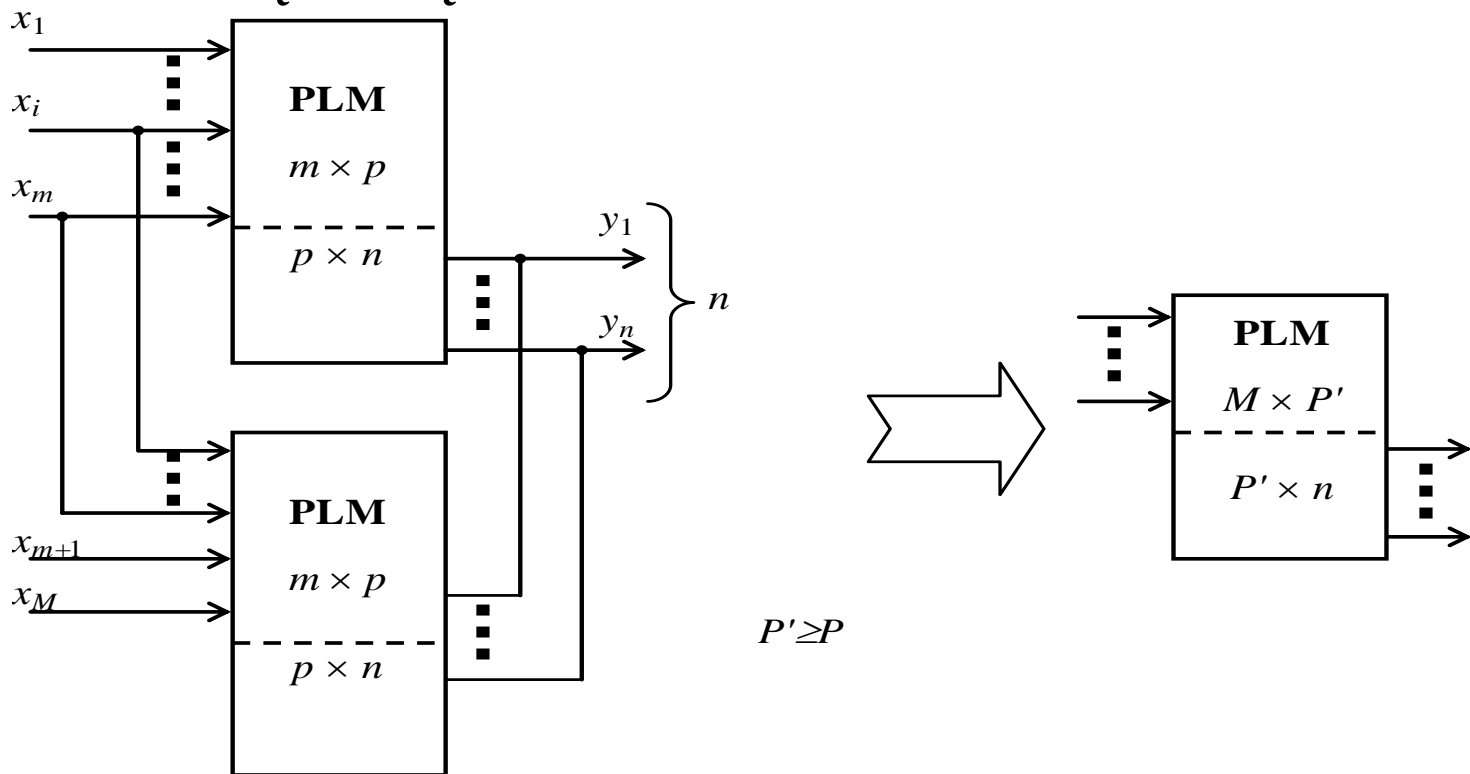
PLM termų kiekio išplėtimas

- PLM įvestys sujungiamos lygiagrečiai, o atitinkamos išvestys sujungiamos naudojant ARBA elementą arba sujungiamos tarpusavyje (tribūvės išvestys), arba sujungtos naudojant „montažinį ARBA“ (atviro kolektoriaus išvestys).



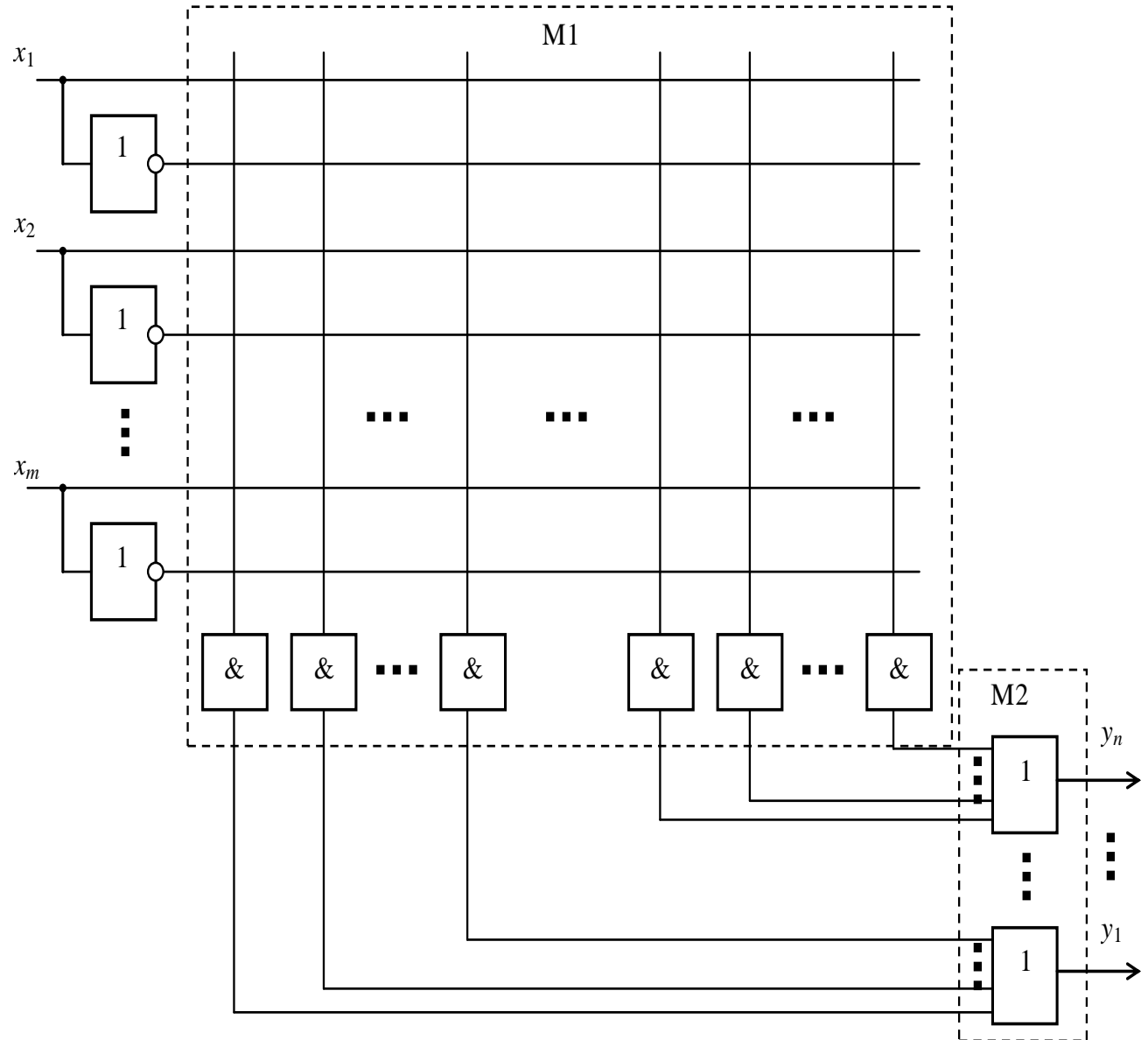
PLM įvesčių kiekio išplėtimas

- Kai įvesčių kiekis didesnis nei PLM ($M > n$), vykdoma funkcijų dekompozicija termų aibę išskaidant į poaibius, turinčius ne daugiau, kaip m vienodų kintamųjų, kuriems panaudoti atskiri PLM, o išvestis sujungtos taip pat, kaip ir išplečiant termų kiekį.



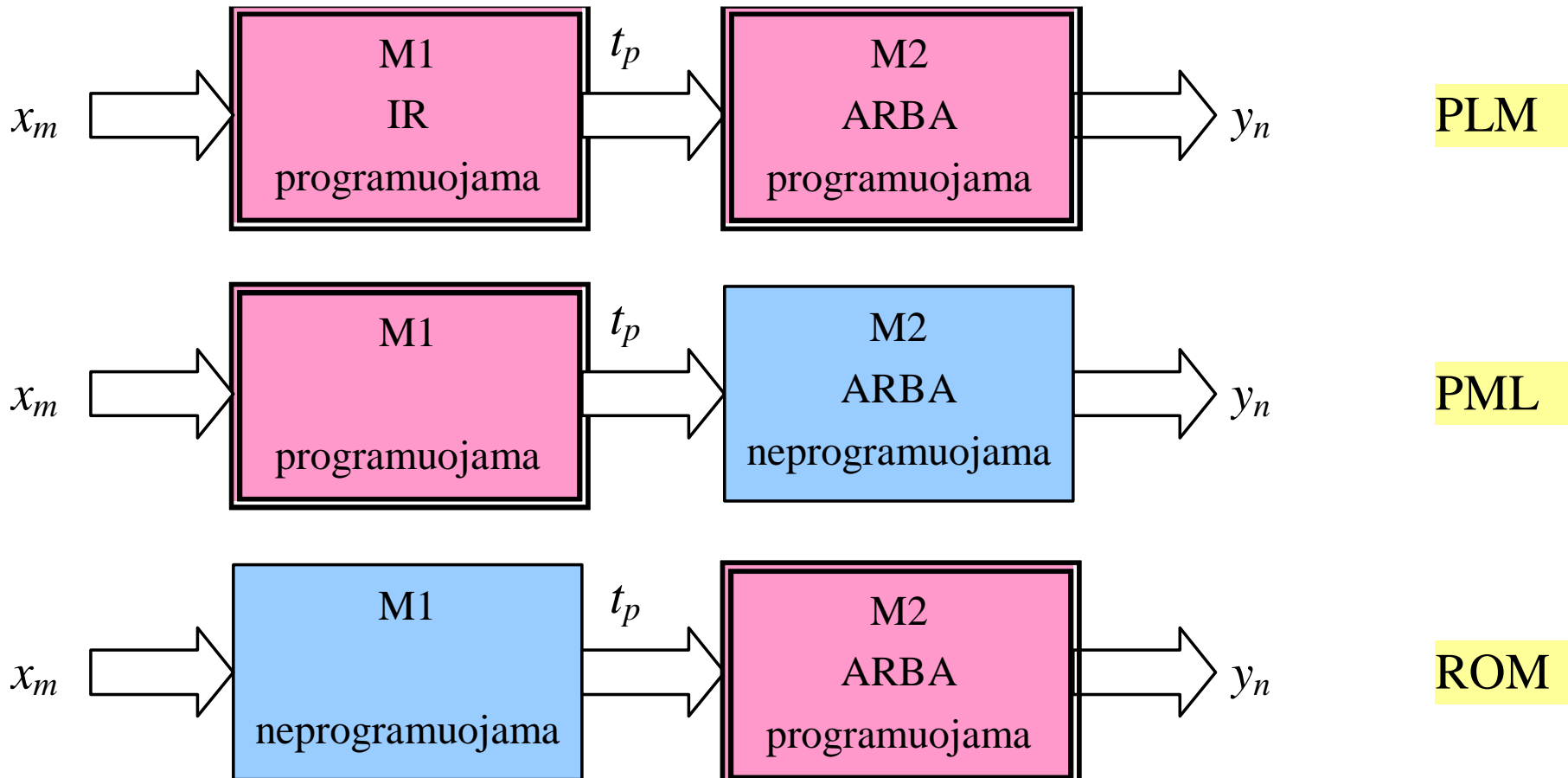
Programuojama matricinė logika PML

- PML elementų IR išvestys (termai) yra tvirtai paskirstytos tarp ARBA elementų.
- Kartais uždavinys supaprastėja neprogramuojant matricos ARBA.



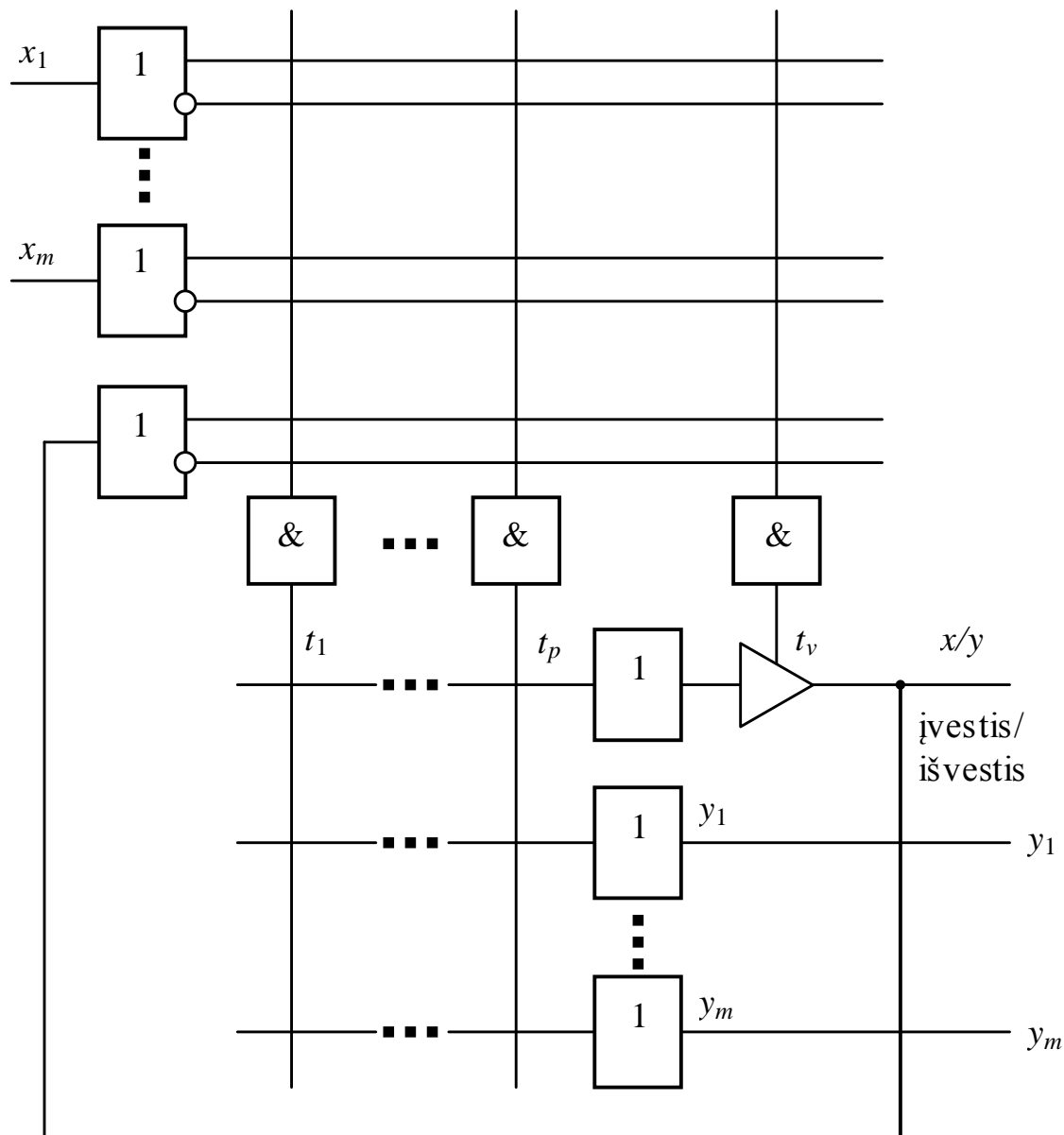
PLS skirstymas

- Pagal matricių programavimą PLS skirstomos į: PLM; PML; pastovias atmintines (ROM).



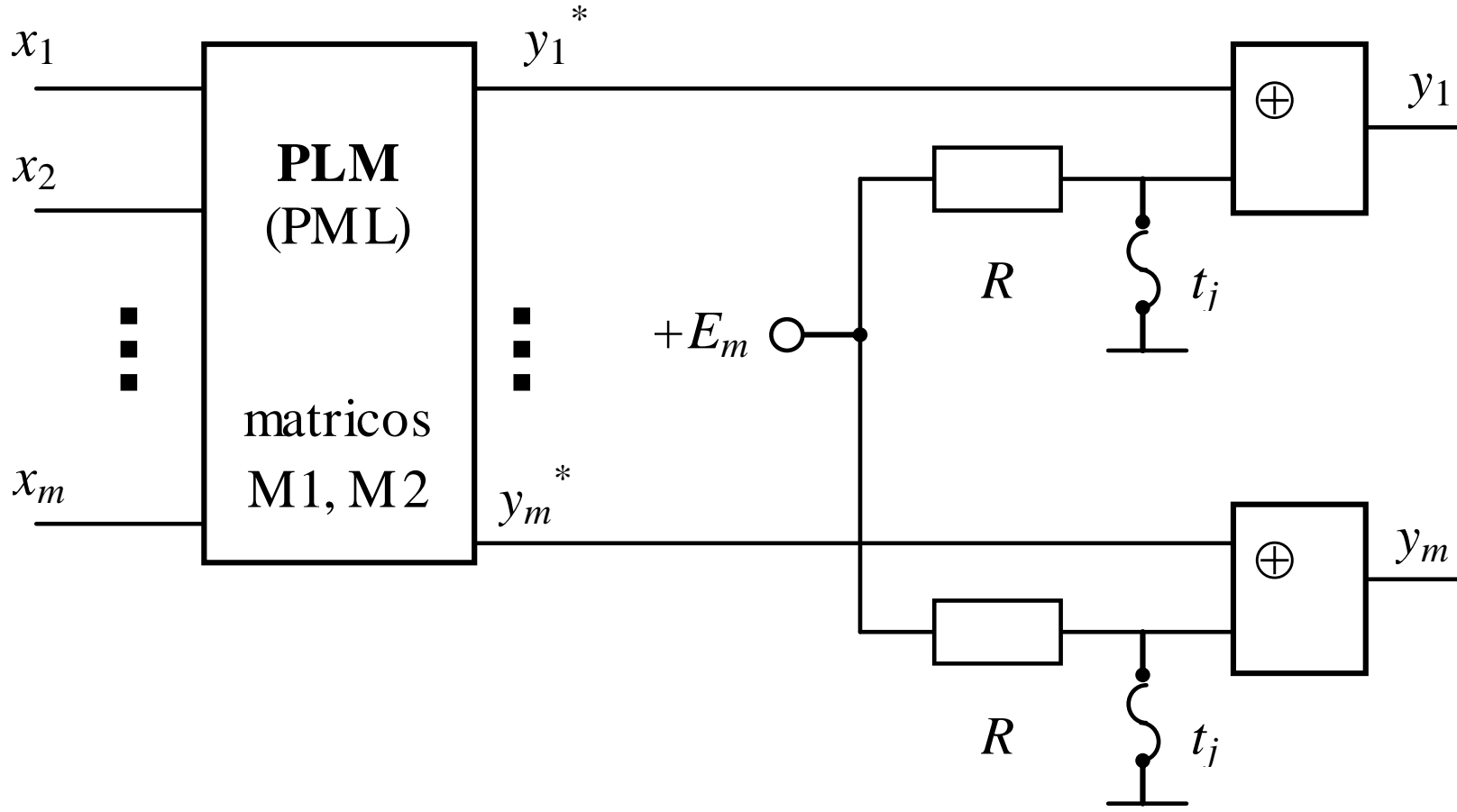
PLS struktūros

- Plečiamos PLS funkcinės galimybės įvedant įvairius papildymus jų struktūrose.
- Schemose su dvikrypčiais išvadais dirbančiais skirtingais režimais (išvesties, įvesties, grįžtamojo ryšio išvesties, programuojamos išvesties).



PLS su programuojama išvestimi

- Išvestyje galima turėti tiesioginę ar inversinę funkcijos reikšmę.
- Galima supaprastinti schemos realizavimą.



Programuojamos išvesties panaudojimas

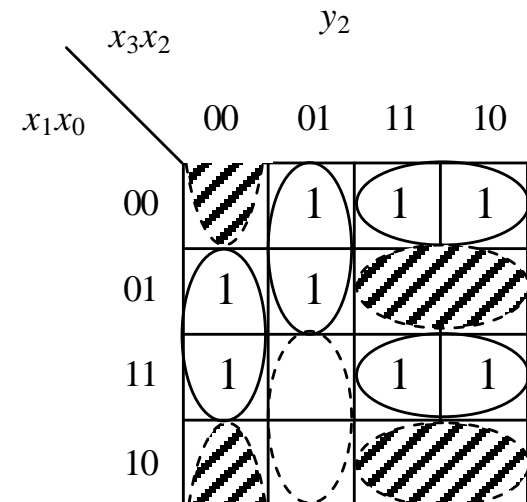
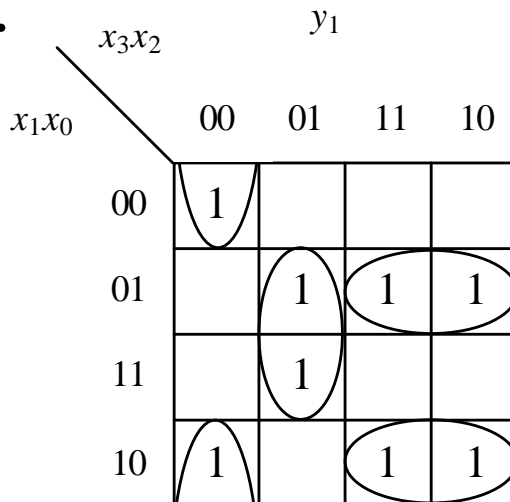
- Realizuojame dvi funkcijas (reikia 8 termų):
- Išnaudojus invertuotas išvesties galimybę, pakaks 5 termų, nes y_2 invertuotos reikšmės 3 termai sutampa su y_1 termiais.

$$y_1 = \overline{x_0}\overline{x_2}\overline{x_3} + x_0x_2\overline{x_3} + x_0\overline{x_1}x_3 + \overline{x_0}x_1x_3$$

$$y_2 = \overline{x_0}\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + x_0\overline{x_1}x_3 + x_0x_1x_3$$

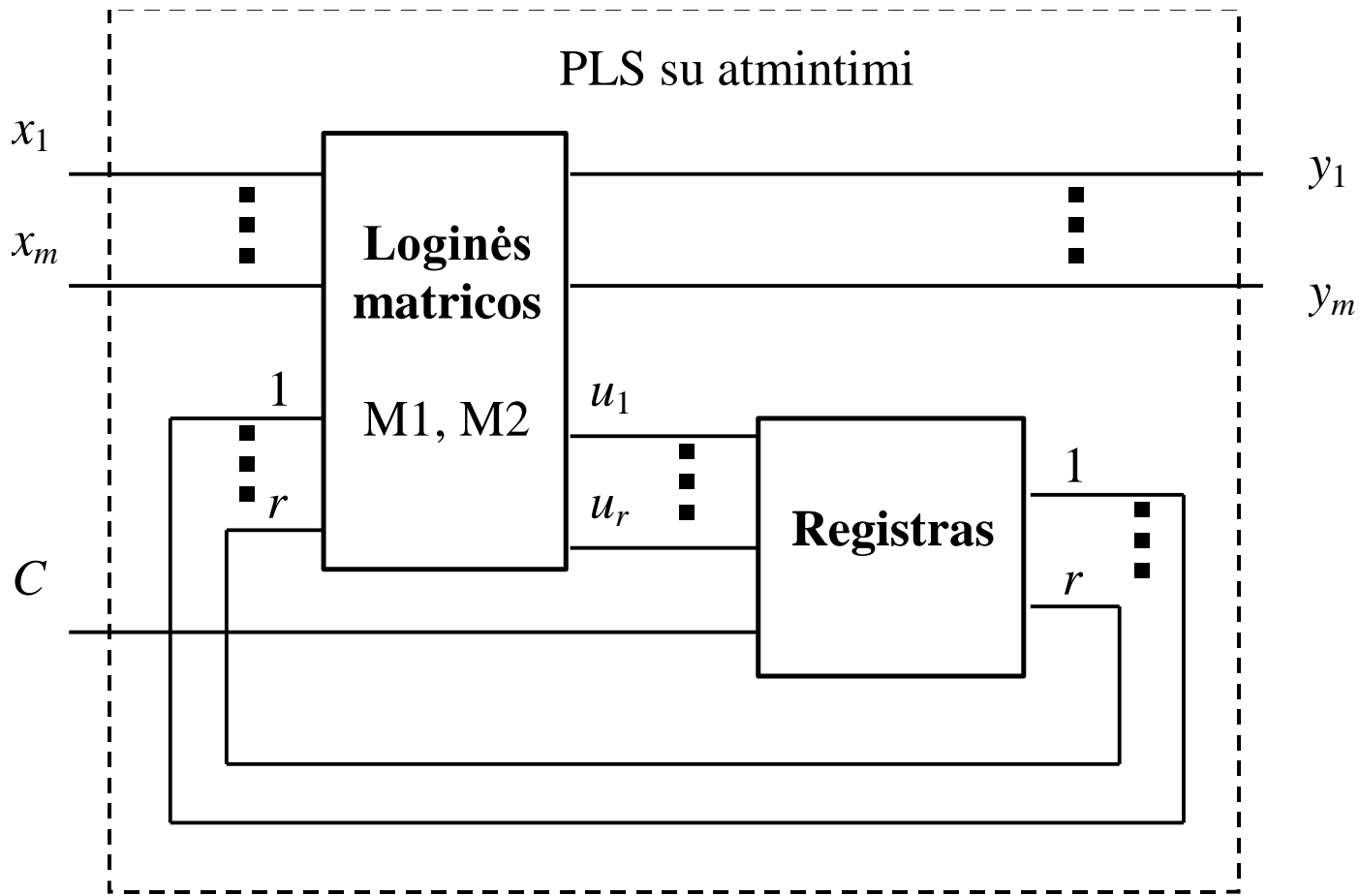
$$y_1 = \overline{x_0}\overline{x_2}\overline{x_3} + x_0x_2\overline{x_3} + x_0\overline{x_1}x_3 + x_3x_1\overline{x_0}$$

$$y_2 = \overline{x_0}\overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3} + x_0\overline{x_1}x_3 + \overline{x_0}x_1x_3$$



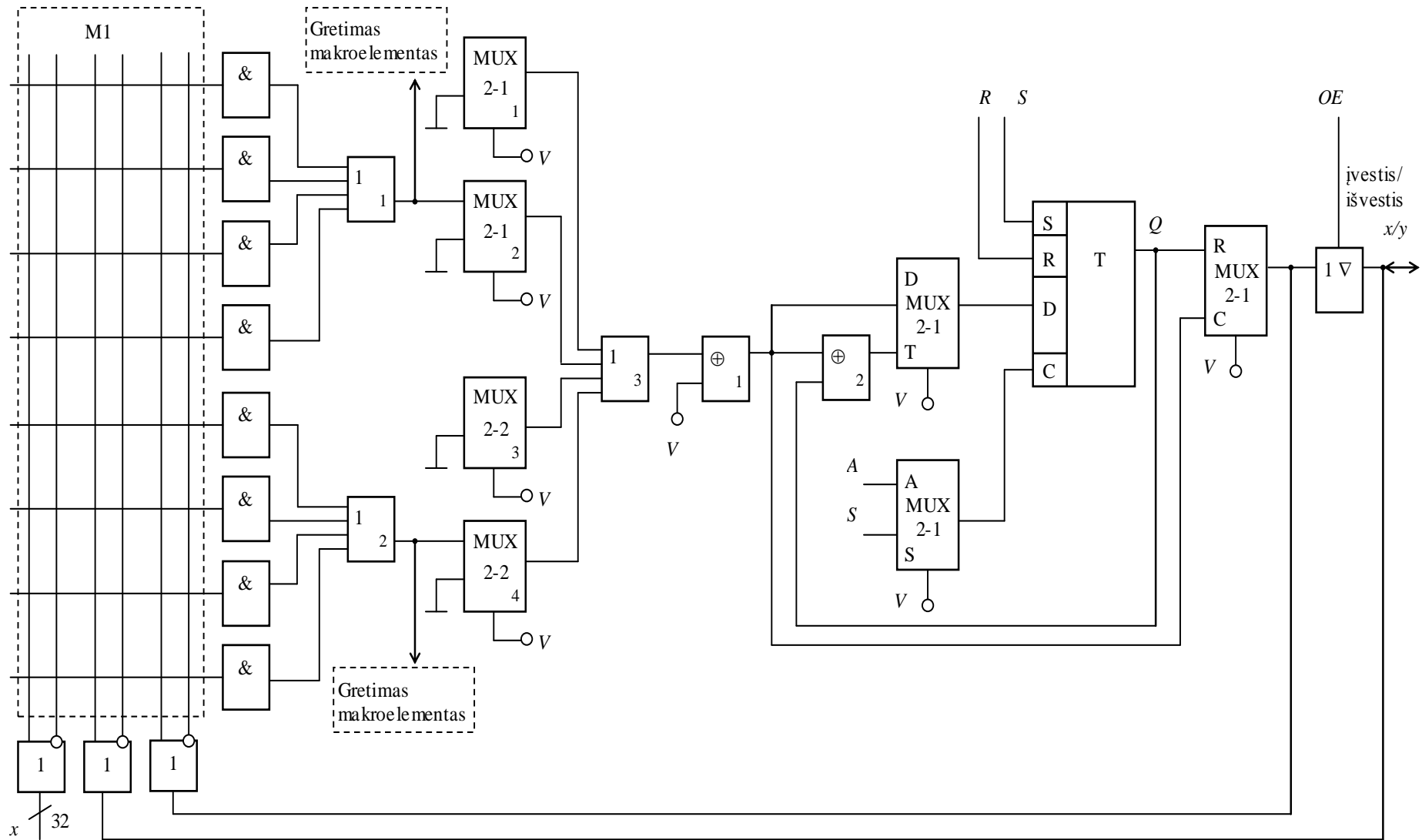
PLS su atminties elementais

- PLS turi parametą r – atminties elementu kiekį.
- Galima realizuoti automatus (iki 2^r vidinių būsenų).
- Būsenos keičiamos tik atėjus sinchronizacijos signalui.



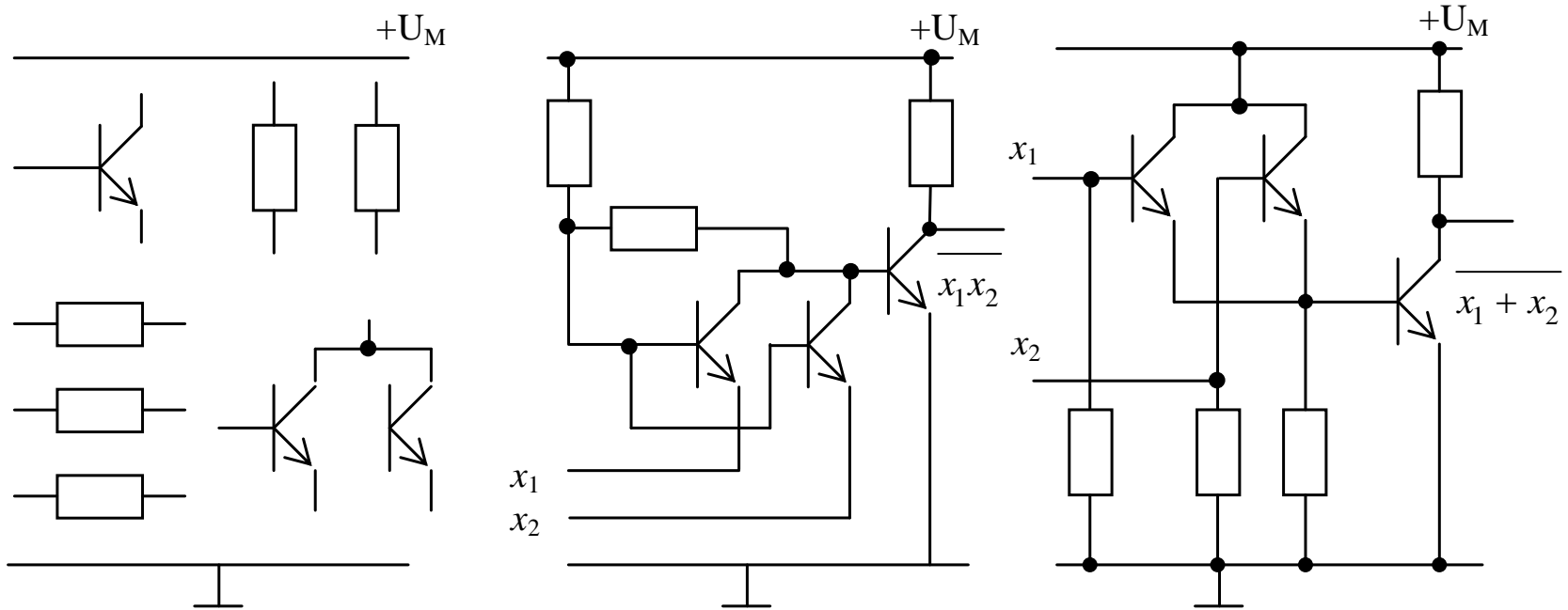
PLS su makroelementais

- Makroelementai leidžia keisti PLS funkcines galimybes.



Baziniai matriciniai kristalai

- BMK (GA) (1975) pagrindas – reguliariai išdėstytos bazinės ląstelės (BL) ir yra zona sujungimų (kanalų) sudarymui.
- BL – tai nesukomutuoti elementai.
- Sujungimus kristale realizuoja vartotojas naudojant plonasluoksne technologiją.

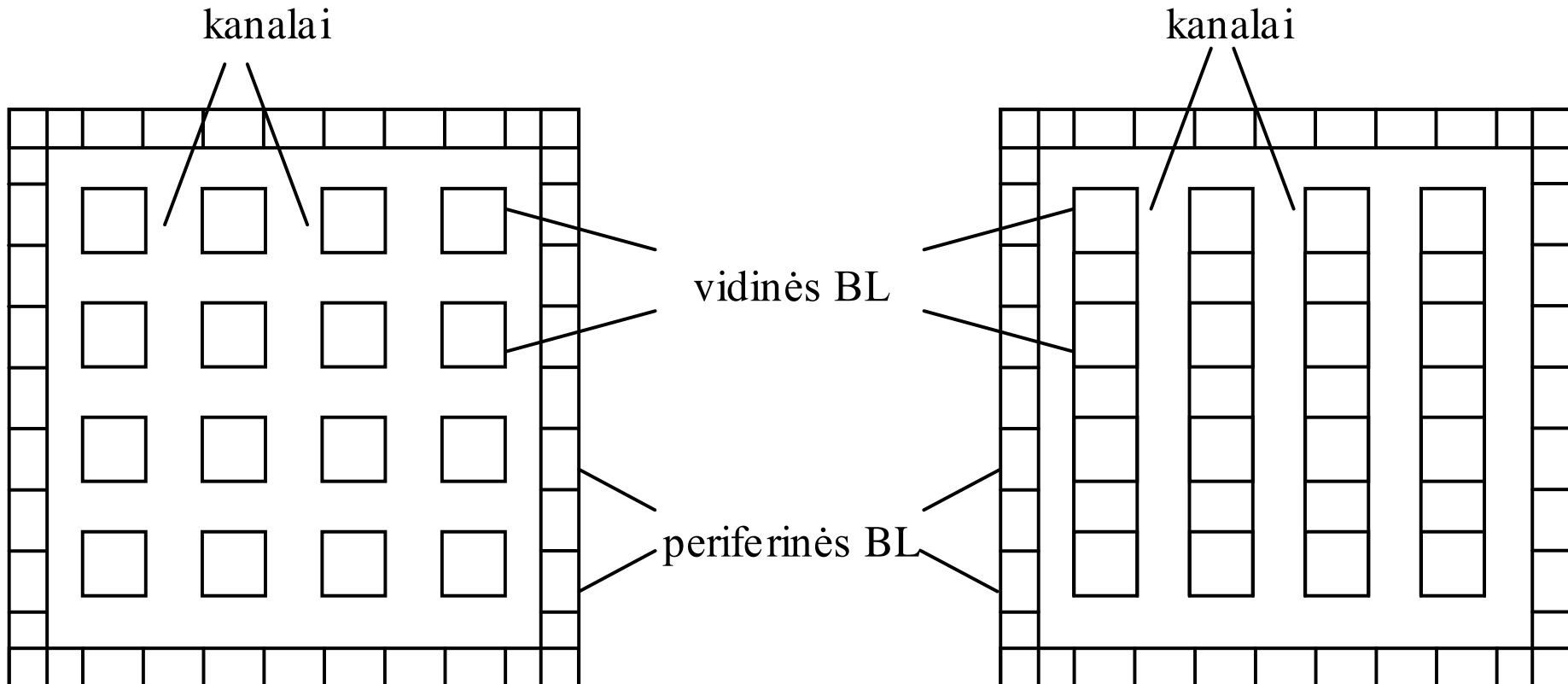


BL sudarymo būdai

- Iš nesujungtų elementų BL sudarytas vienas loginis elementas, o sudėtingos funkcijos realizavimui panaudojamos kelios BL.
- Iš nesujungtų elementų BL gali būti sudarytas bet koks funkcinis mazgas, o elementų kiekį ląstelėje apsprendžia sudėtingiausia realizuojama funkcija.
- Funkcinė ląstelė (FL) tai užbaigta schema, realizuota sujungiant vienos ar kelių BL elementus.
- FL rinkinys sudaro funkcinų ląstelių biblioteką, kuri naudojama projektuojant BMK. BMK projektuojama funkciname-loginame lygyje (bibliotekoje esamais elementais), o schemotechninio lygio projektavimo problemos sprendžiamos sudarant biblioteką.

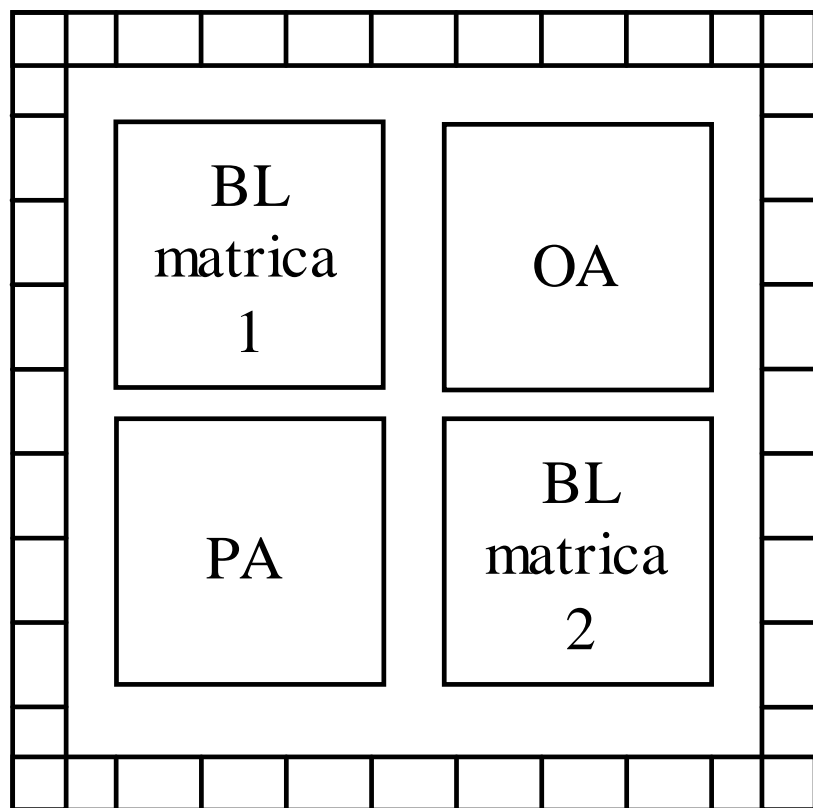
Kanalinė BMK struktūra

- Kanalai gali būti vertikalūs ir horizontalūs arba tik vertikalūs.
- Kanalinės BMK pasižymi didelėmis ryšių sudarymo galimybėmis, tačiau neefektyviai išnaudojamas kristalo paviršius.



Blokinė BMK struktūra

- Blokiniai BMK turi ne tik loginių schemų ląsteles, bet ir atminties bei kitokius specializuotus blokus.
- Svarbi BMK charakteristika – jungiamųjų sluoksnių kiekis (turi nuo 2 iki 6).

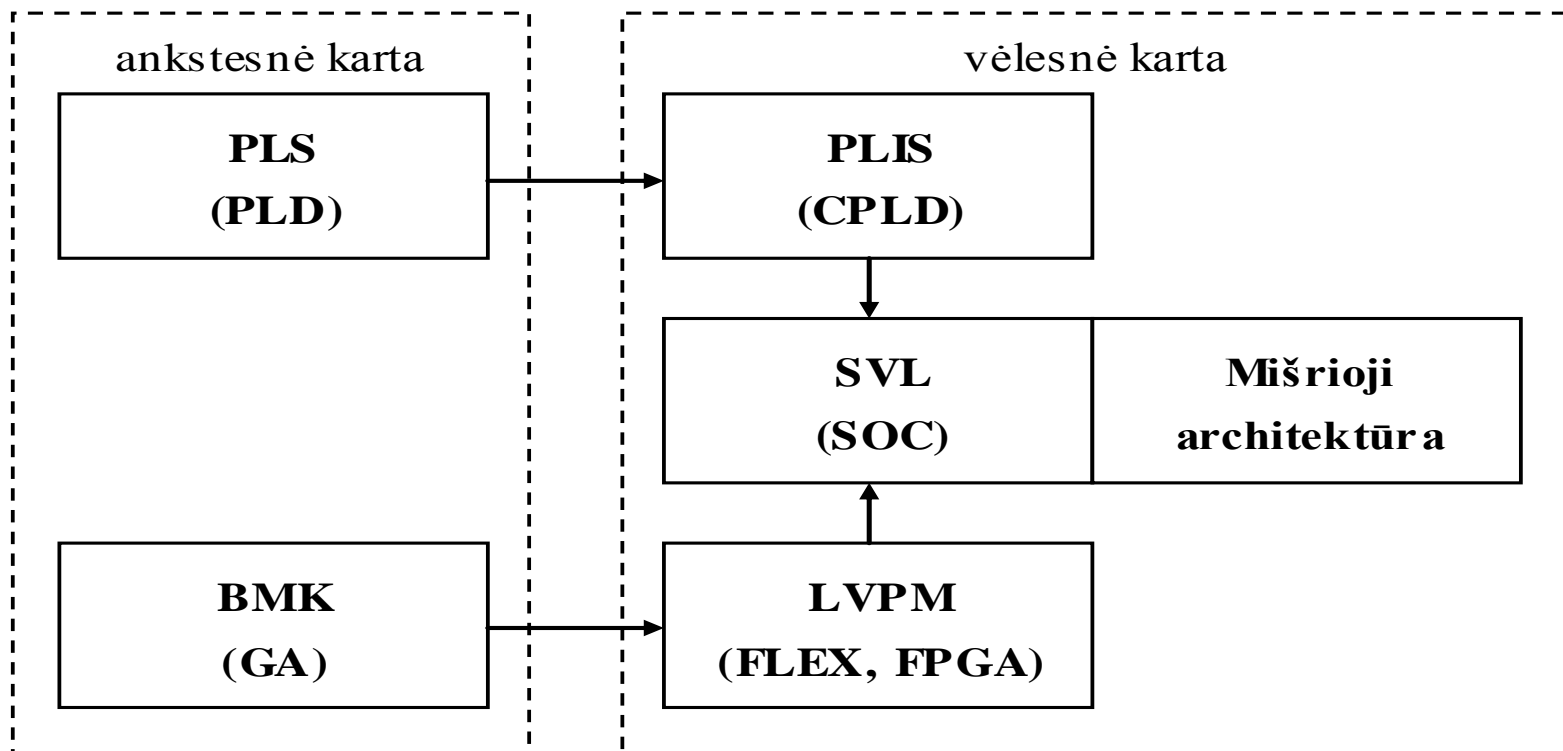


OA – operatyvioji atmintinė (RAM)
PA – pastovioji atmintinė (ROM)

periferinės BL

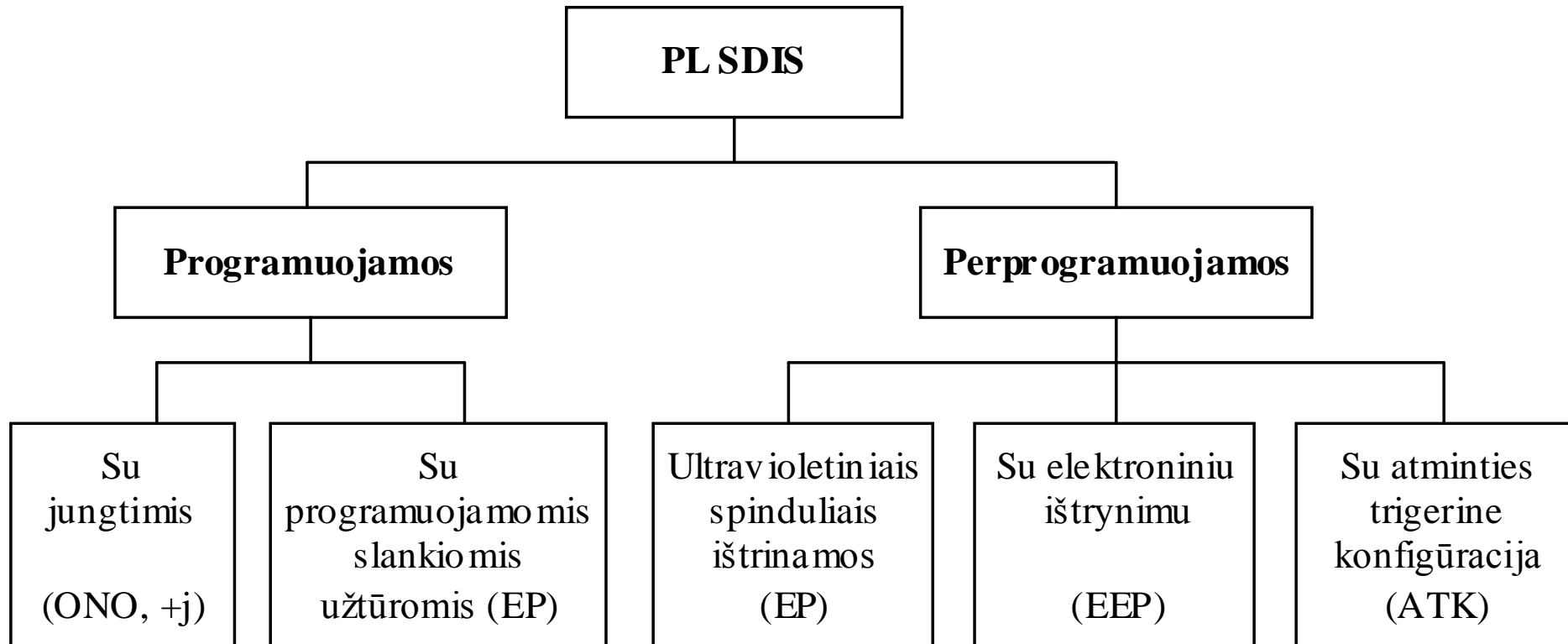
Programuojamas loginės integrinės schemos PLIS (CPLD

- Tobulinant PLS sukurtos PLIS (CPLD), o tobulinant BMK – lanksčiosios programuojamos ventilinės matricos LPVM (FLEX arba FPGA), o didinant integracijos lygį viename kristale sukurtos sistemos viename luste SVL (SOC)



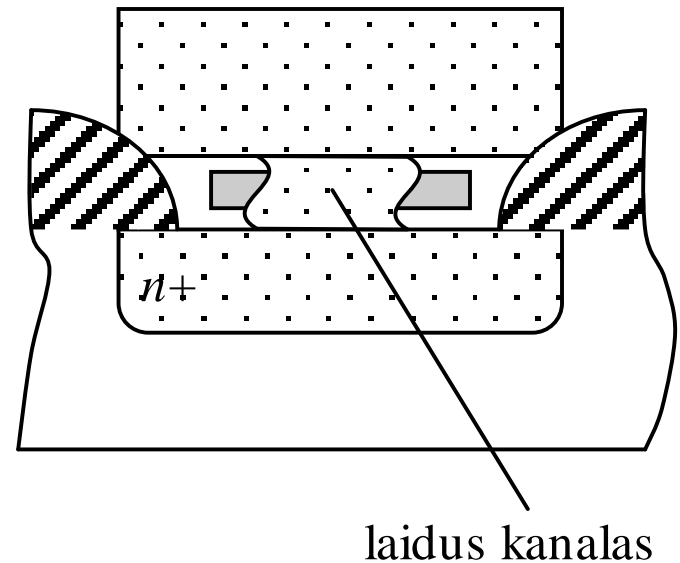
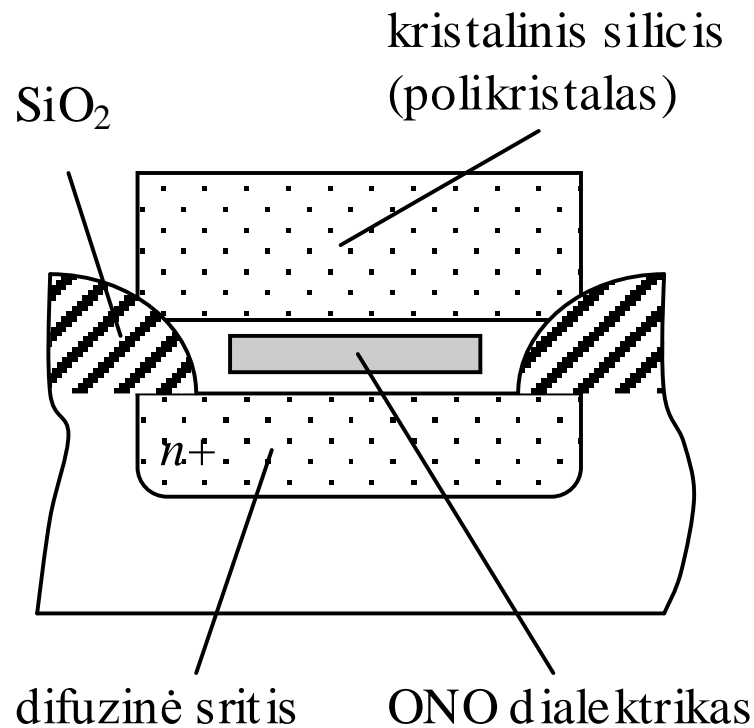
Programuojami elementai

- Programuojamos ar perprogramuojamos logikos DIS/SDIS mikroschemas vadinsime PL SDIS.
- Naudojamos jungtys: oksido-nitrato-oksido (ONO); elektriškai programuojamos jungtys su slankiomis užtūromis (EP).



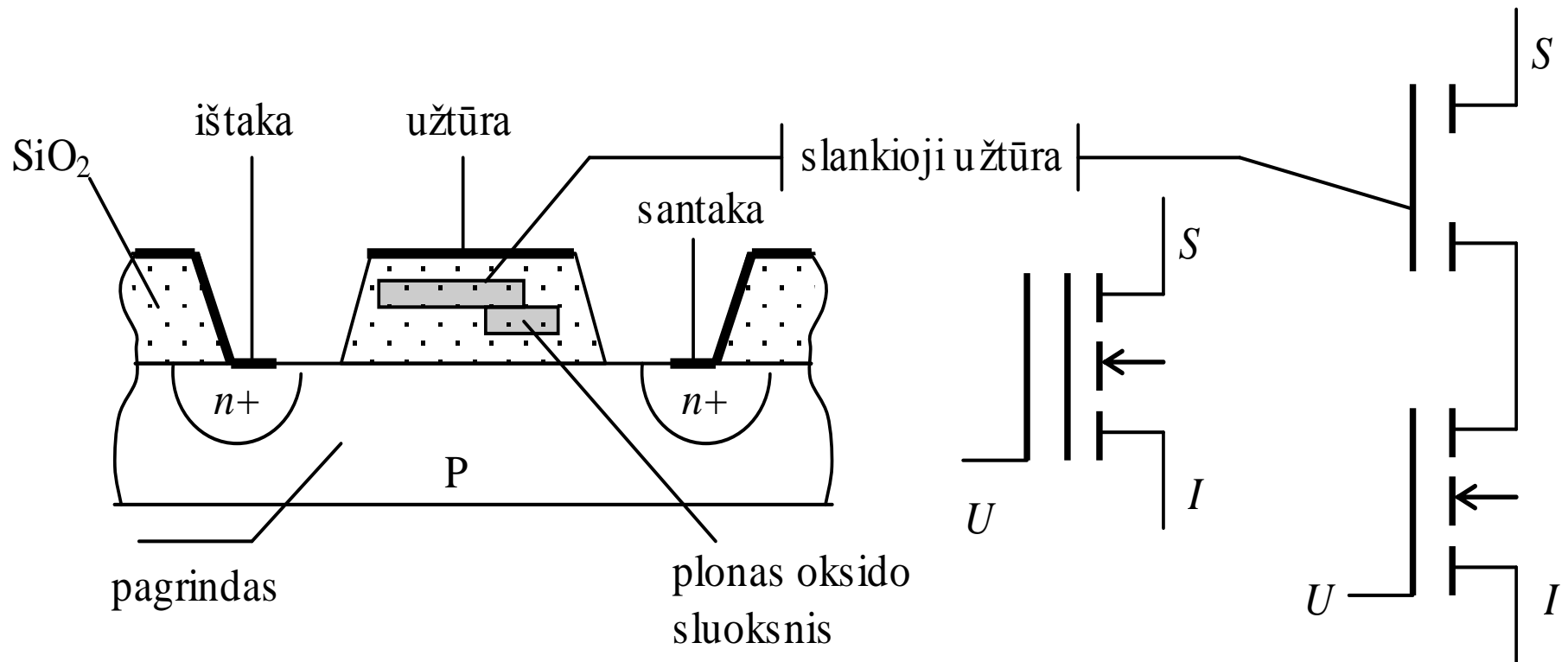
Oksido-nitrato-oksido (ONO) jungtys

- ONO jungtys užima labai mažai vietos ant kristalo ($<1\mu\text{m}$) ir užtikrina gerus parametrus (neužprogramuotos jungties srovė apie 10^{-5} A , užprogramuotos jungties varža apie $100\ \Omega$, parazitinė talpa $<10^{-14}\text{ F}$).
- Jungties varža priklauso nuo programavimo srovės dydžio.



“Slankios užtūros” jungtis

- Jungties elementai – MOP tranzistoriai su slankia užtūra.
- Tarp užtūros ir santakos prijungiamoje aukšto įtampa (≤ 50 V) įvyksta plono SiO_2 sluoksnio lavininis pramušimas ir elektronų injekcija į slankiąją užtūrą, o išjungus įtampą elektronai išsilaikys, tranzistorius dirbs kaip jungtis.

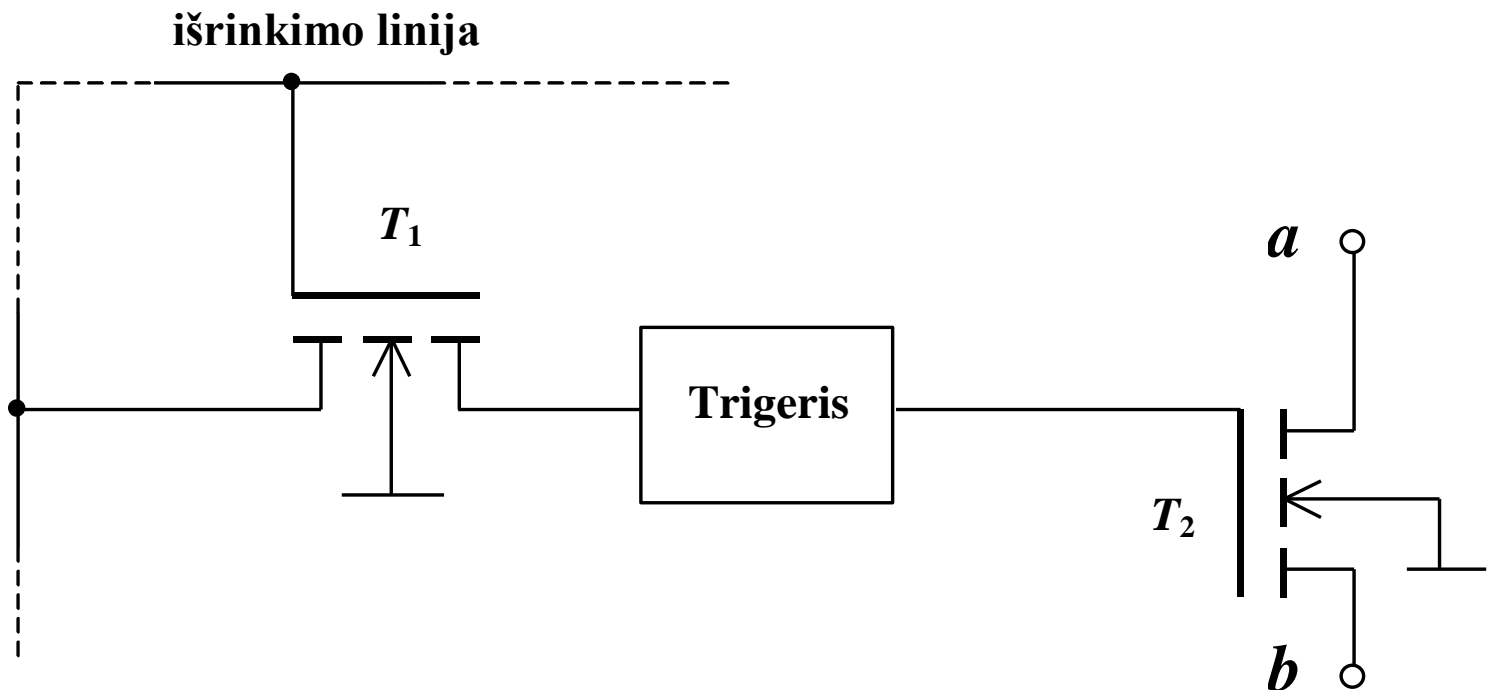


Elektriškai perprogramuojamos jungtys

- Jungties perprogramavimui reikia apšvitinti užtūras ultravioletiniais spinduliais. Elektronai įgauna energijos ir nugalėję oksido potencialinį barjerą išlekia iš slankios užtūros. Tai elektriškai programuojamos (EP) jungtys.
- Jei trinant informaciją tarp užtūros ir santakos prijungiama priešingos krypties, negu programuojant, įtampa, turime elektriškai programuojamas ir ištrinamas jungtis (EEP).
- EEP jungčių žymiai geresni elektriniai programavimo ir ištrynimo parametrai gaunami, kai vietoje silicio oksido įvedamas silicio nitridas (SiN).
- Jungčių perprogramavimo ciklą galima atlikti 10^4 - 10^6 kartų.
- EEP technologija vis tobulėja ir išstumia EP technologiją.

Tranzistoriniai jungikliai su triggerine atmintinės konfigūracija

- Valdomas triggerio programuojamas jungimo taškas PJT (PIP) T_2 sujungia (užtrumpina) arba atjungia (nutraukia) schemos dalį tarp taškų a ir b .
- Perprogramuoti jungiklius galima neribotą kiekį kartų.

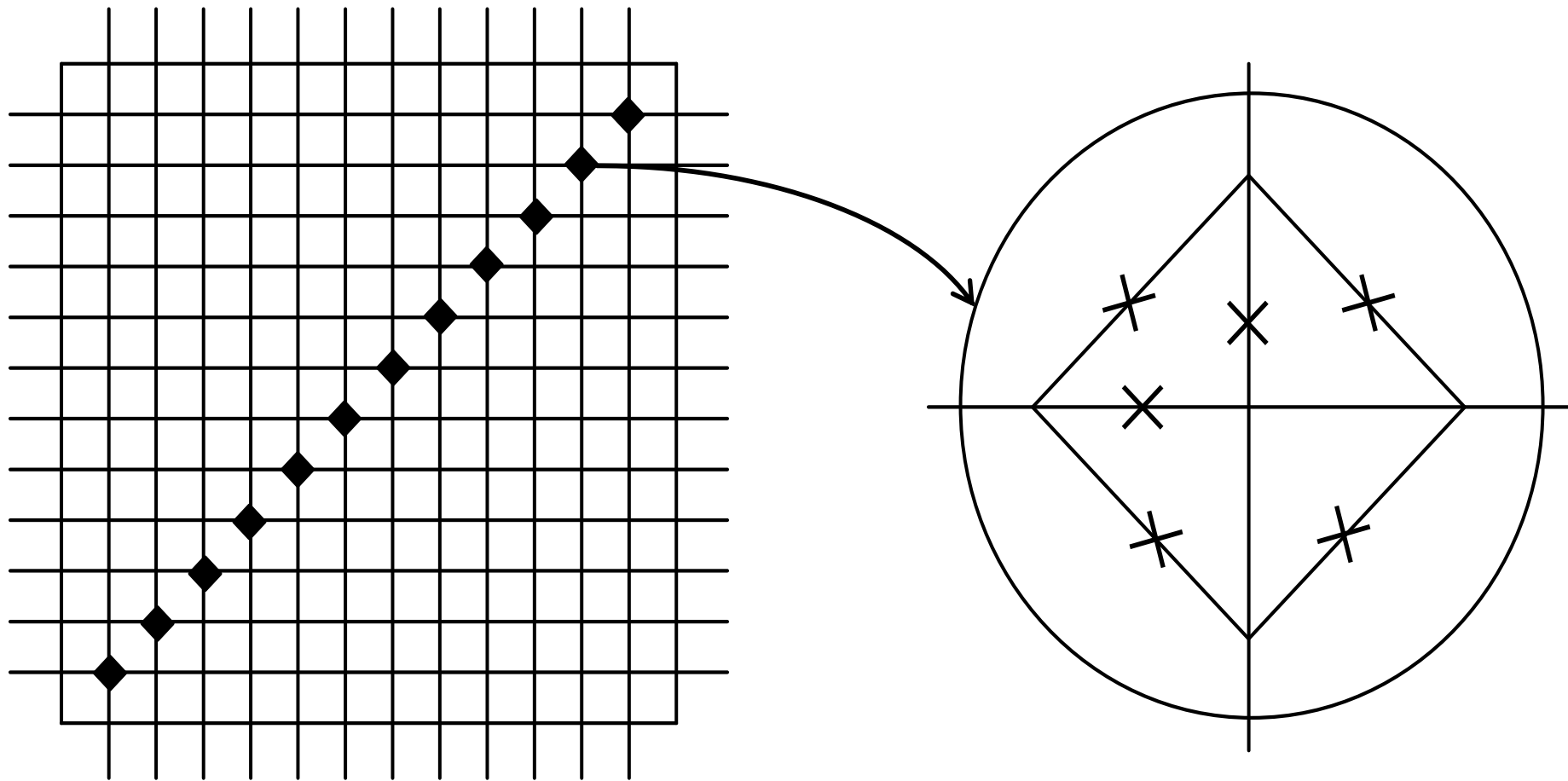


Šiuolaikiniai PL SDIS

- Šiuolaikinių PL SDIS architektūra panaši į BMK.
- Vidinėje kristalo srityje patalpinti sudėtingi konfigūruojami loginiai blokai LB, o periferijoje – periferiniai įvesties/išvesties blokai PB.
- LB svarbios charakteristikos yra grūdėtumas ir funkcionalumas.
- Grūdėtumas tai skaidymo lygis, nurodantis kokio sudėtingumo gali būti elementai, iš kurių surenkamos norimos schemas.
- Smulkiausias grūdėtumas bus, kai LB sudarys tranzistorių su p ir n kanalais grandinėlės.

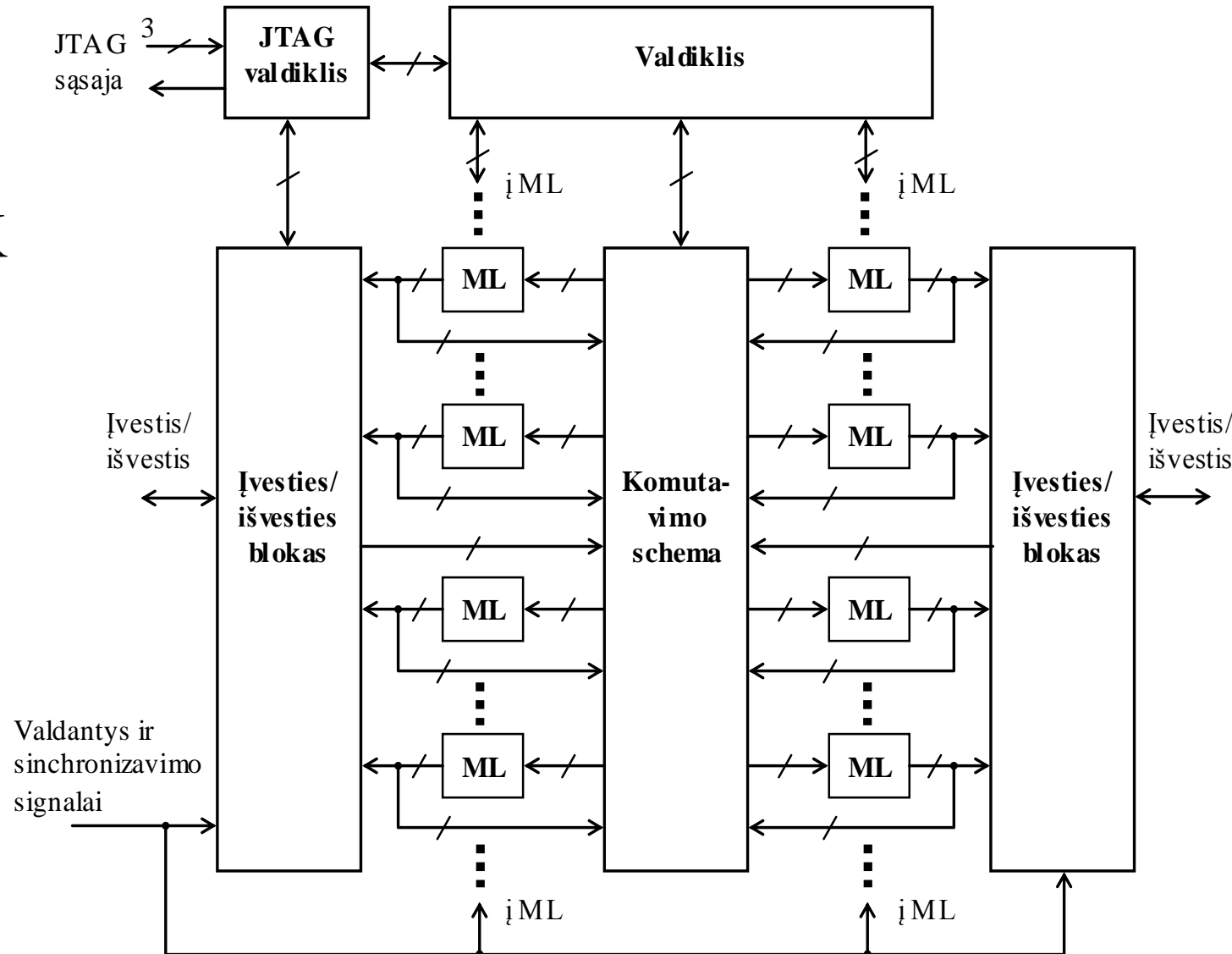
Sujungimo bloko struktūra

- Kryžiuokais parodyti programuojami ryšio taškai.
- Leidžia perduoti signalus į kairę, į dešinę, į viršų, į apačią tarp gretimų linijų, pakeisti signalų perdavimo kryptį.



Sudėtingos programuojamos loginės integrinės schemos SPLIS

- SPLIS buvo sukurta tobulinant PLS ir BMK struktūras.
- Tai schemas, kurių pagalba galima realizuoti sistemas luste.



SPLIS architektūra

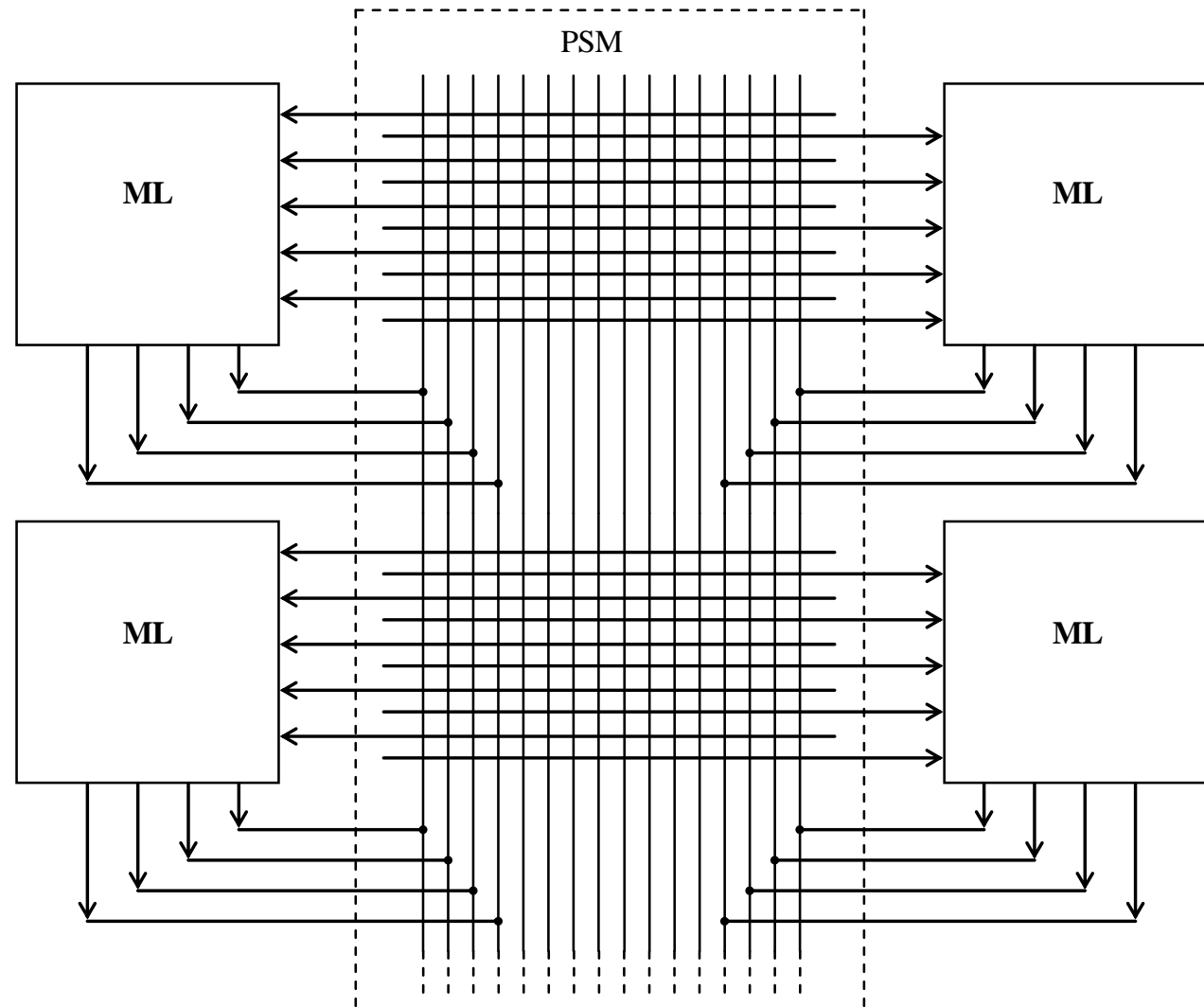
- SPLIS architektūra susideda iš programuojamos sujungimo (komutavimo) matricos PSM (PIA), aibės makroląstelių ML (FB) ir įvesties/išvesties blokų.

Makroląstelės turi PLS, BMK struktūras su plčiom funkcinėm galimybėm.

SPLIS realizuota nepertraukiama ryšių sistema, ir visi ryšiai yra identiški.

Makroląstelių komutavimo schema

- PSM realizuota kaip viena matrica, leidžianti sujungti bet kurios ML išvestis su kitų ML įvestimis.
- Kiekvienos ML viduje gali būti sava (lokali) komutavimo sistema.

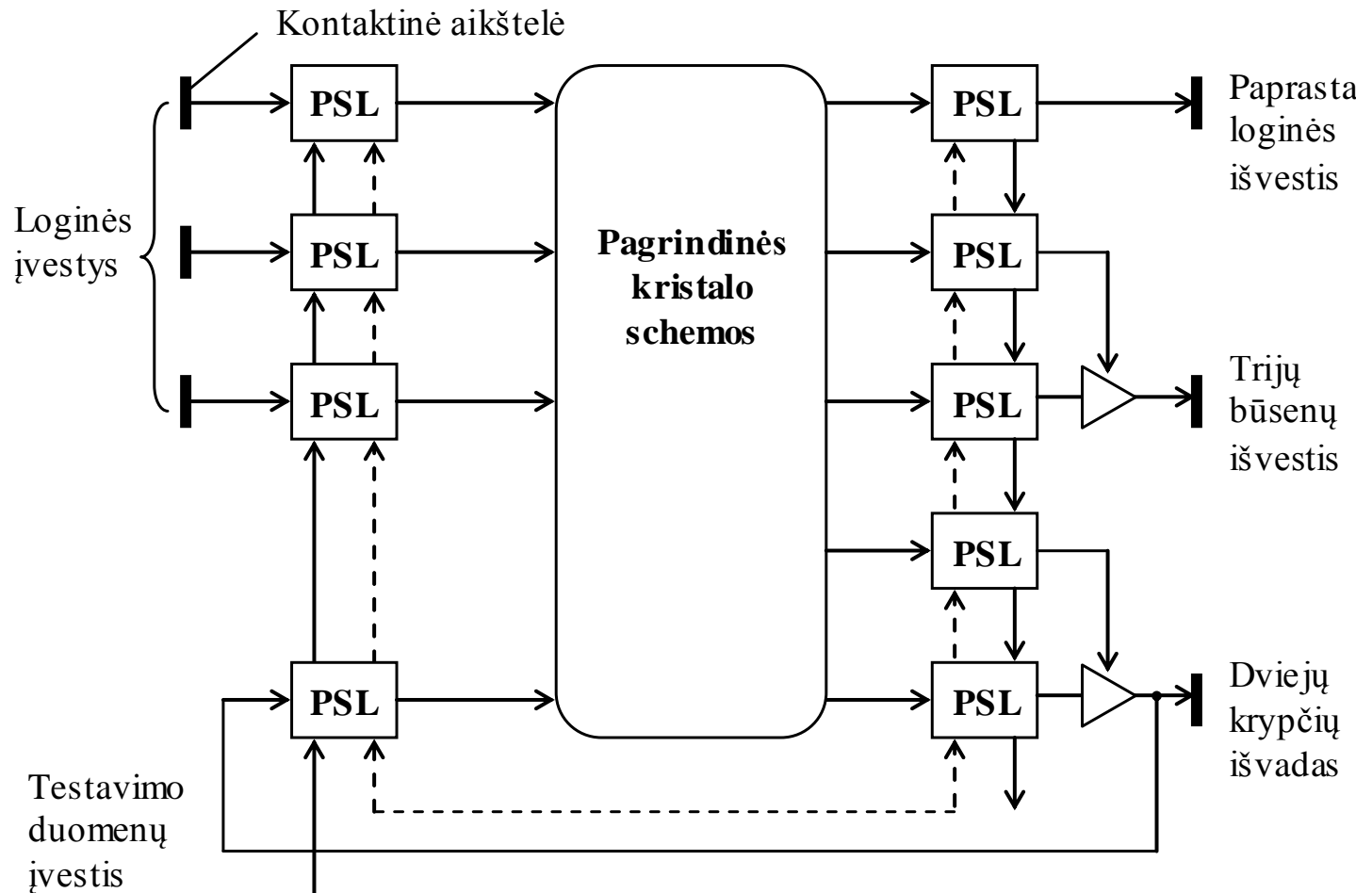


SPLIS realizavimas

- SPLIS architektūroje plačiai naudojama konfigūruojama triggerinė atmintis ir EEPROM elementai. tuo pačiu įsitvirtina ir SPLIS mišri architektūra, kurioje panaudota PLS, PLIS, LPVM struktūros.
- Beveik visose SPLIS yra testavimo sistema JTAG.
- Mikroschemų viduje yra specialios periferinio skenavimo ląstelės PSL ir jų darbą valdančios schemas (JTAG valdiklis) .
- Testavimo režime PSL susijungia nuosekliai ir sudaro postūmio registrą vykdančią ir papildomas funkcijas.

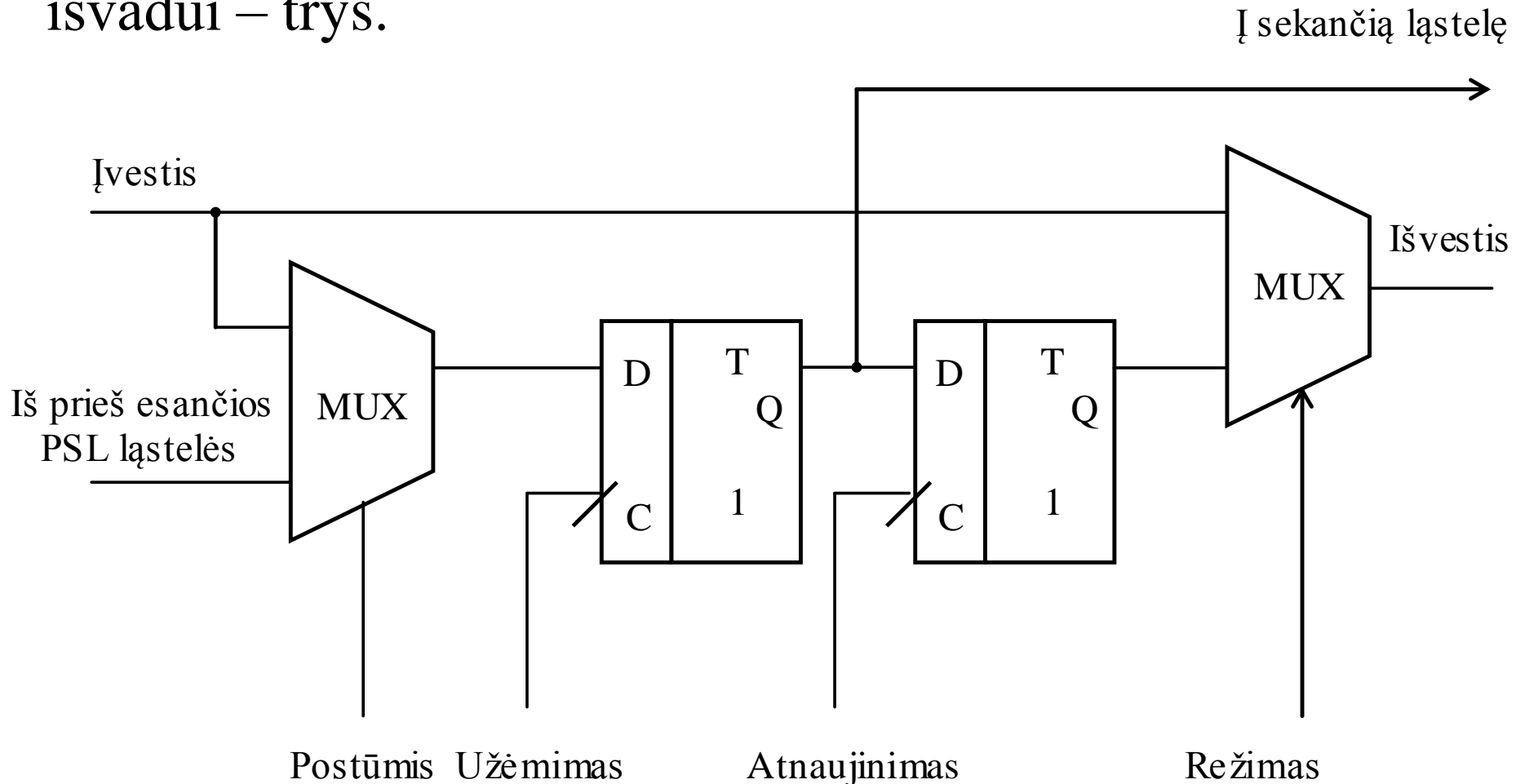
JTAG sąajos

- Testavimas naudojant JTAG standartą (IEEE Std 1149.1) dažnai vadinamas periferiniu skenavimu.



Periferinio skenavimo ląstelės schema

- PSL schema turi du multiplekserius ir du *D* trigerius.
- Trijų būsenų išvadui skiriamos dvi PSL, o dvikrypčiam išvadui – trys.



PSL funkcionavimas

- Priklausomai nuo valdančio signalo išvesties multiplekseris arba praleidžia signalą nuo įvesties iki išvesties arba į išvestį perduoda antrojo trigerio išvestį, o įvesties multiplekseris valdo įvesties signalo padavimą į pirmąjį trigerį nuo mikroschemos įvesties arba nuo prieš esančios PSL.
- Gali registro, sudaryto iš pirmųjų trigerių, reikšmę perrašyti į statinį registrą, sudarytą iš antrųjų trigerių. Taigi reikšmė, postūmis registre, sudarytam iš pirmųjų trigerių, neturės įtakos registro, sudaryto iš antrųjų trigerių, reikšmei.

JTAG funkcionavimas

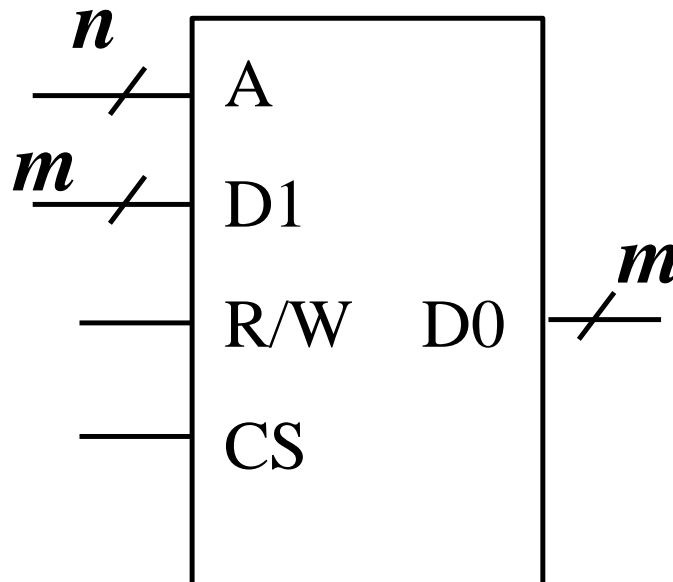
- Specialios paskirties mikroschemos testiniai išvadai yra 4 (kartais 5). Šių išvadų funkcinė paskirtis tokia:
- TDI (testinių duomenų įvestis) – PS nuoseklių duomenų įrašomo įvestis (priekiniu frontu);
- TDO (testinių duomenų išvestis) – nuoseklių duomenų išvedamo išvestis (neigiamu frontu);
- TCK (testinio sinchronizavimo įvestis) – sinchronizuoja TI valdiklio darbą;
- TMS (testavimo valdymo įvestis) – nustato testavimo režimą.
- Atitinkama signalų TDI ir TMS seka užtikrina JTAG valdiklio ir jo darbo komandų įvedimą.
- Įvedimi duomenys, vykdoma komanda, nuskaitomi rezultatai

Atmintinės

- Atmintinės yra skirtos saugoti informacijai, patenkančiai iš kitų įrenginių arba iš išorės per įvestis ir perduodamai į įrenginius per išvestis.
- Pagrindiniai atmintinių parametrai yra informacinė talpa, veikimo sparta, kaina.
- Atmintinių lygiai: registrinės, vidinės ir išorinės.
- Atmintinių pagrindiniai:
 - - ląstelės žodžio skilčių skaičius (8, 16, 32, 64);
 - - Informacinė talpa – tai maksimaliai galima saugomos informacijos apimtis išreiškiama bitais, žodžiais, baitais (priešdėliai kilo (K) reiškiantis daugiklį $2^{10}=1024$, mega (M) – daugiklį $2^{20}=104876=K*K$, giga (G) – daugiklį $2^{30}=1073.639.424=K*M$);
- Sąlygine kaina tai vieno bito, Kbito, Kbaito ar Mbaito

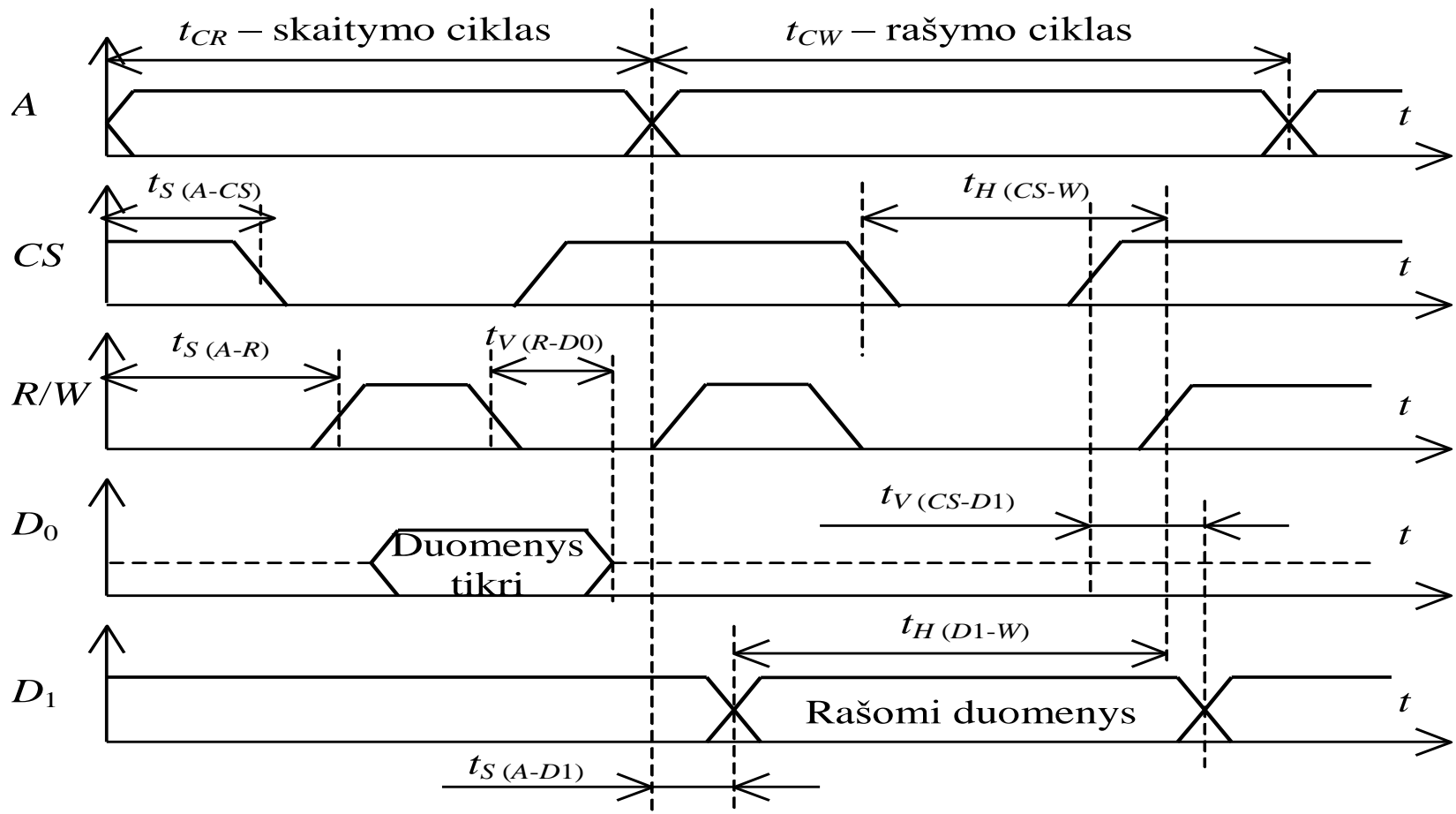
Atmintinių grandinės

- Atmintinių ryšio su išore pagrindinės grandinės: adresų (A – n skilčių); įvesties, išvesties (D_I , D_O); CS (leidžia ar neleidžia mikroschemai dirbti); R/W (nustato informacijos skaitymo ar rašymo operaciją).
- Gali būti naudojami ir kiti valdantieji signalai.

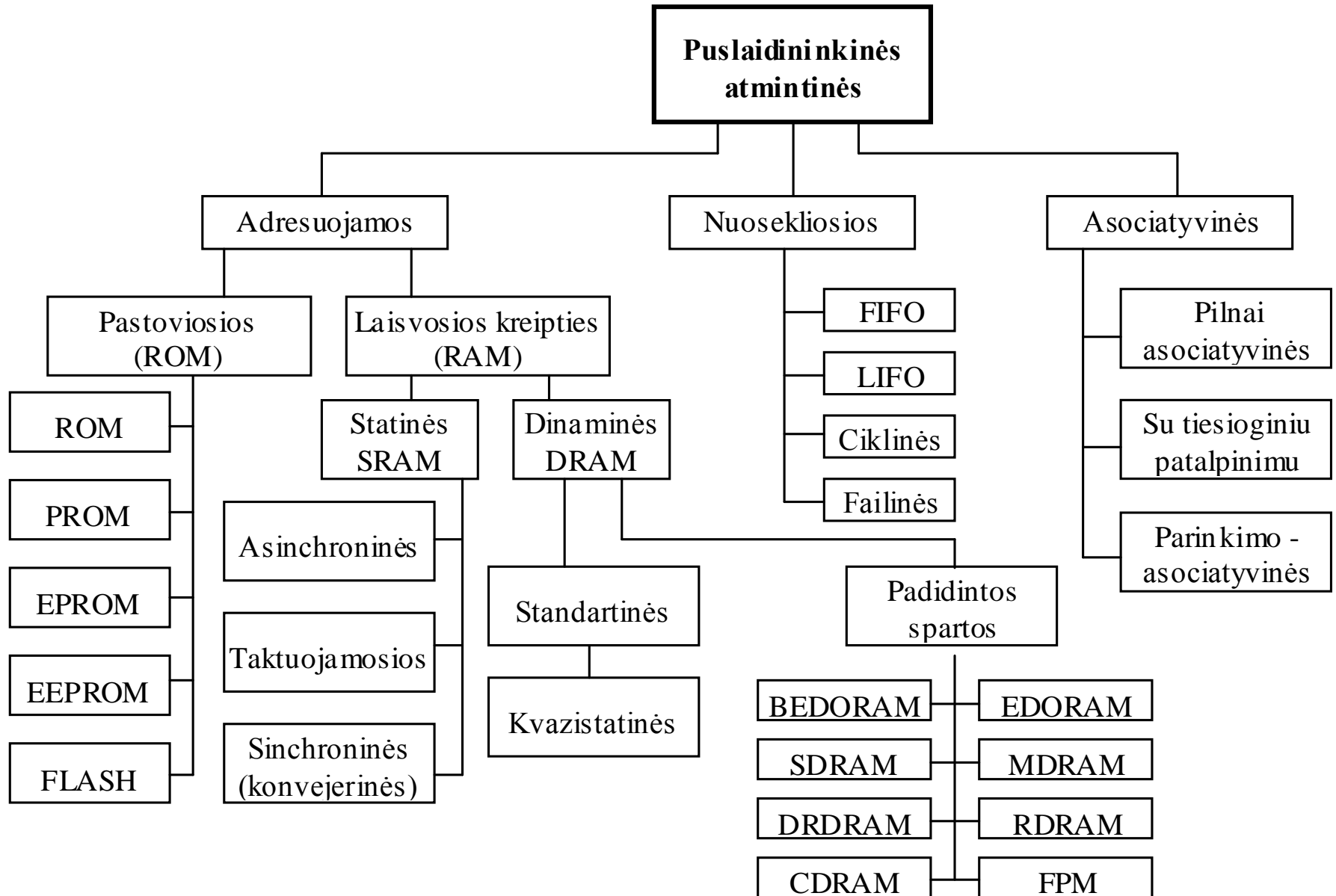


Atmintinės dinaminiai parametrai

- Paduodant į atmintines du signalus privalomi apribojimai:
nustatymo laikas $t_{S(A-B)}$; išlaikymo laikas $t_{H(A-B)}$;
išsaugojimo laikas $t_{V(A-B)}$.



Atmintinių klasifikavimas

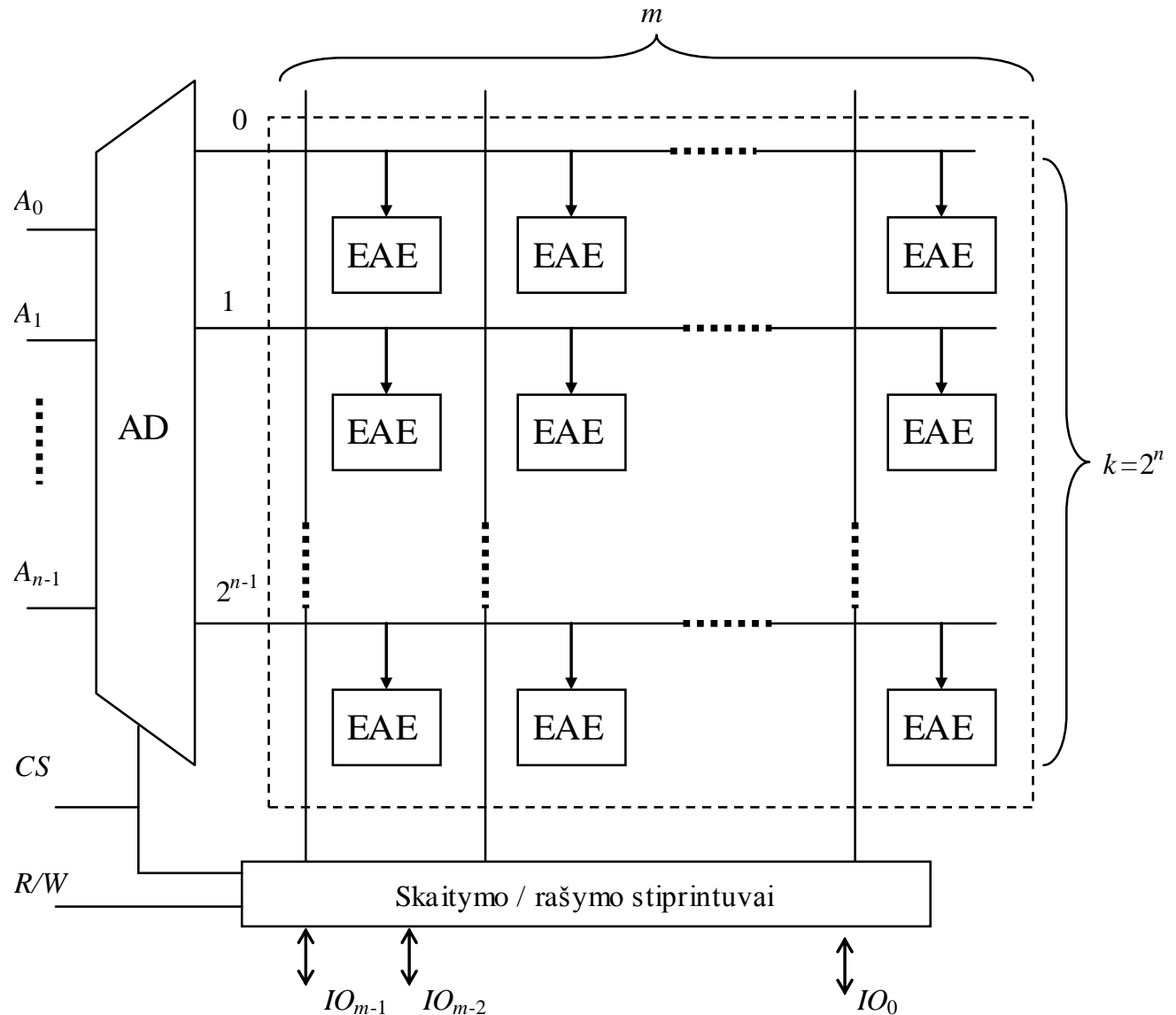


Adresuojamos atmintinės

- Adresuojamose- atminties ląstelę nurodo adresas.
- Laisvosios kreipties atmintinė (RAM) paprastai vadinama operatyviaja atmintine (OA), yra energetiškai priklausoma, tai yra juose saugojama informacija prarandama išjungus maitinimą. Jos skirstomos į statines (elementarūs elementai yra trigeriai) ir dinamines (duomenys saugomi kaip kondensatorių, sudarytų iš MOP elementų, krūviai).
- Pastovios atmintinės (ROM) turinys arba iš viso nesikeičia, arba retai ir tik įjungus specialųjį (rašymo) režimą, o dirbant normaliu režimu informacija skaitoma (energetiškai nepriklausoma).

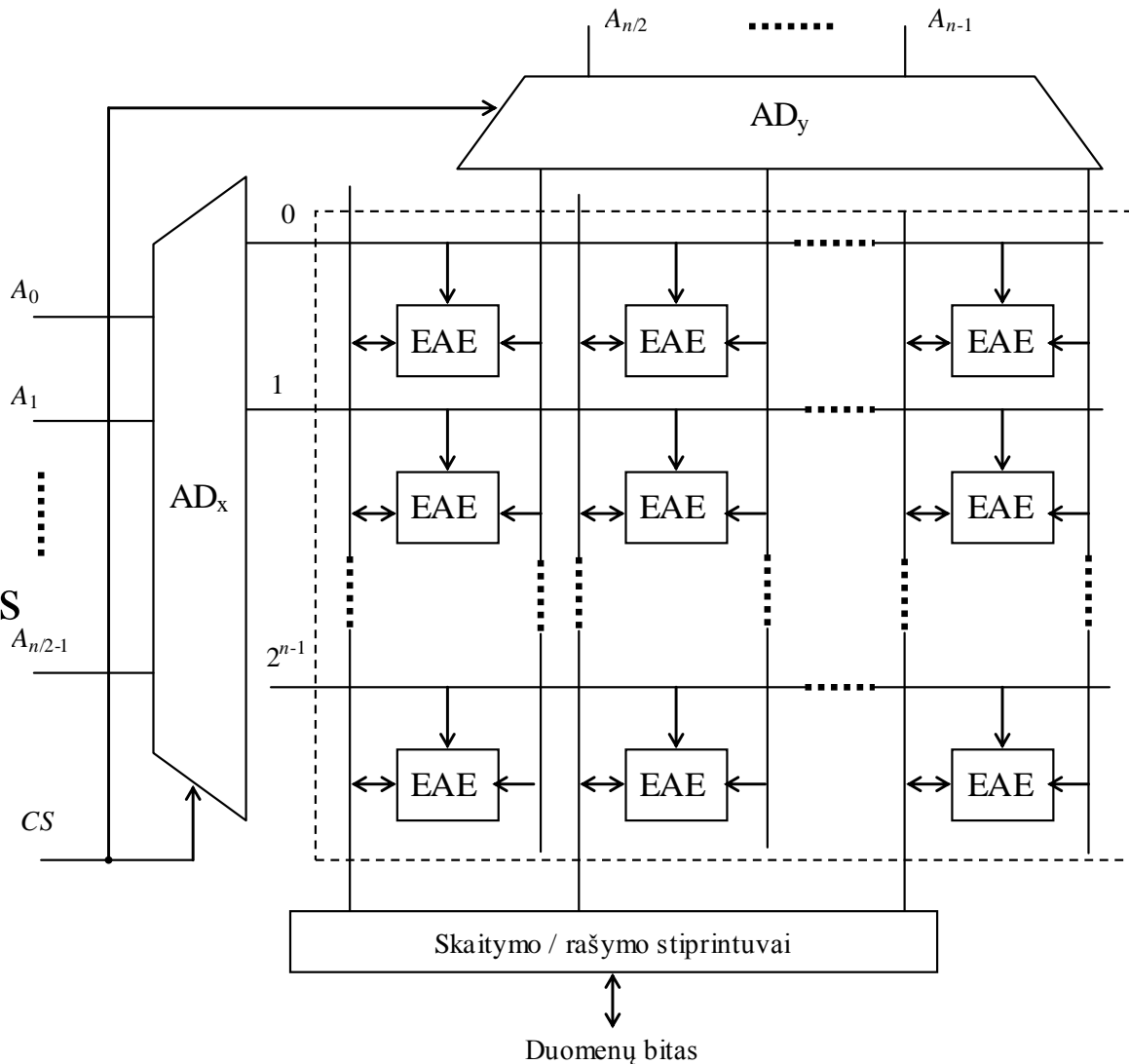
Adresuojamų atmintinių struktūros

- Dažniausiai naudojamos 2D, 3D ir 2DM struktūros.
- 2D struktūra pasižymi tuo, kad atminties ląstelės sujungtos į stačiakampę $M=k \times m$ matricą.



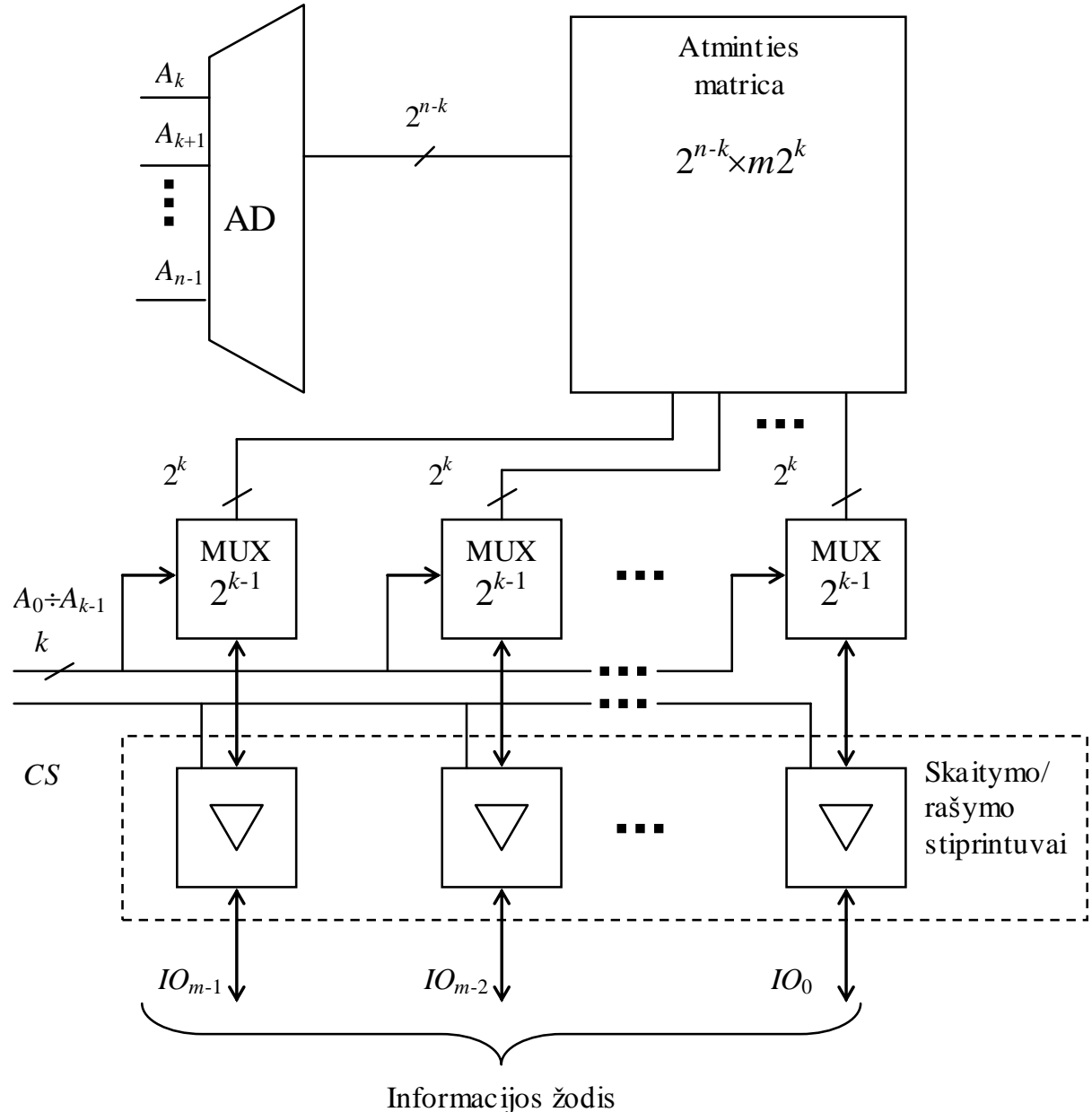
Atmintinės su 3D struktūra

- Dešifratorius žymiai paprastesnis (dviejų koordinačių atminties ląstelių išrinkimas).
- Adreso n skilčių kodas dalijamas į dvi dalis, kurios dekoduojama atskirai.
- Kai atmintinės žodžio ilgis m bitų, atmintinės struktūra pasidarys trimatė, nes dešifratoriai AD_x ir AD_y valdys po m elementarių atminties elementų.



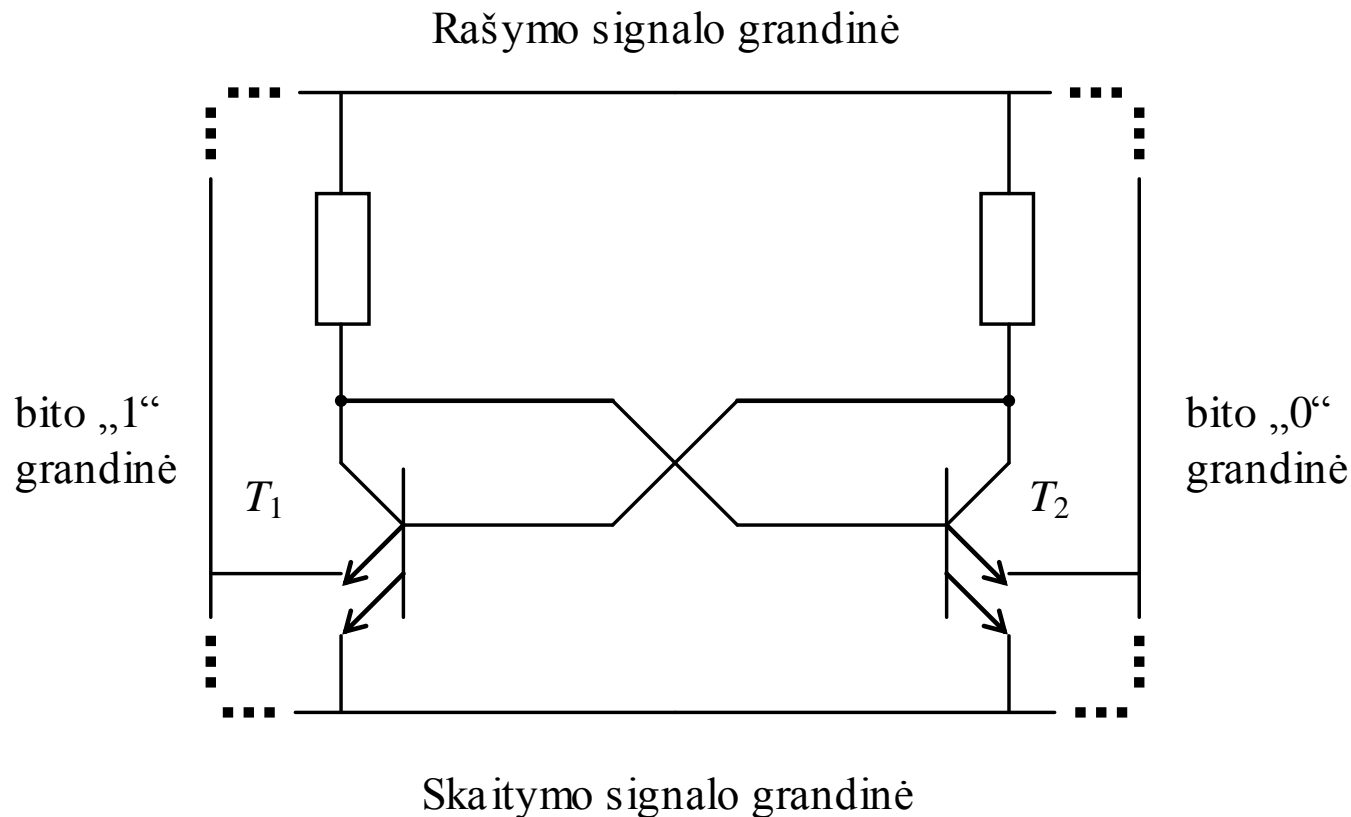
Atmintinės su 2DM struktūra

- $n-k$ adresų linijų paduodama į dešifratorių išrenkantį matricos eilutę, kurioje yra 2^k žodžių m skilčių ilgio.
- Vieną iš 2^k žodžių išrinka m multiplekserių su k adresų linijų.



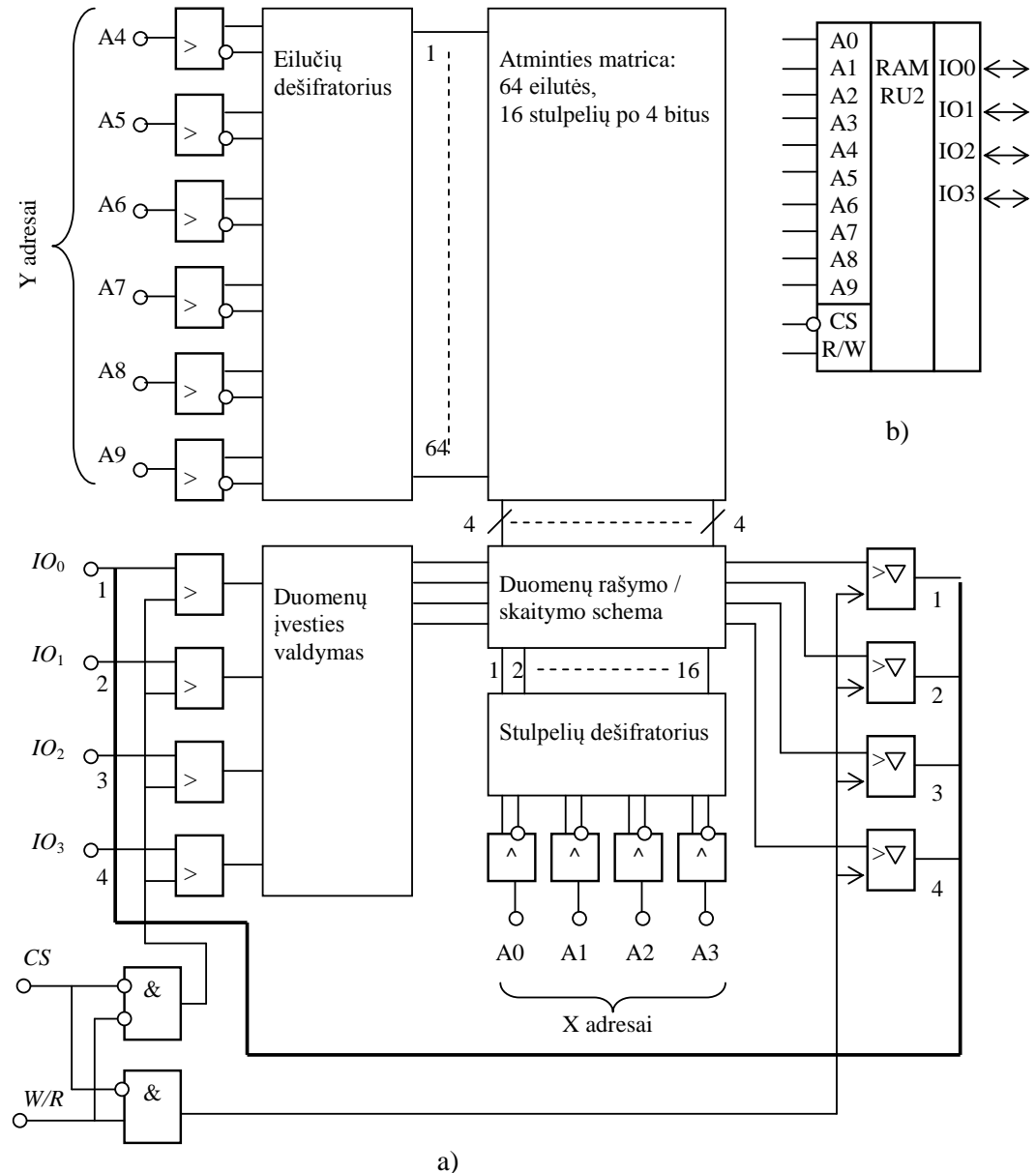
Statinės atmintinės SRAM

- SRAM veikimo sparta didelė, tačiau jos yra brangios.
- Elementarūs atminties elementai yra trigeriai (TTL, TTLŠ, I²L, SEL, n-MOP, KMOP).
- Informaciją saugo tranzistoriai T_1 („1“) ir T_2 („0“).



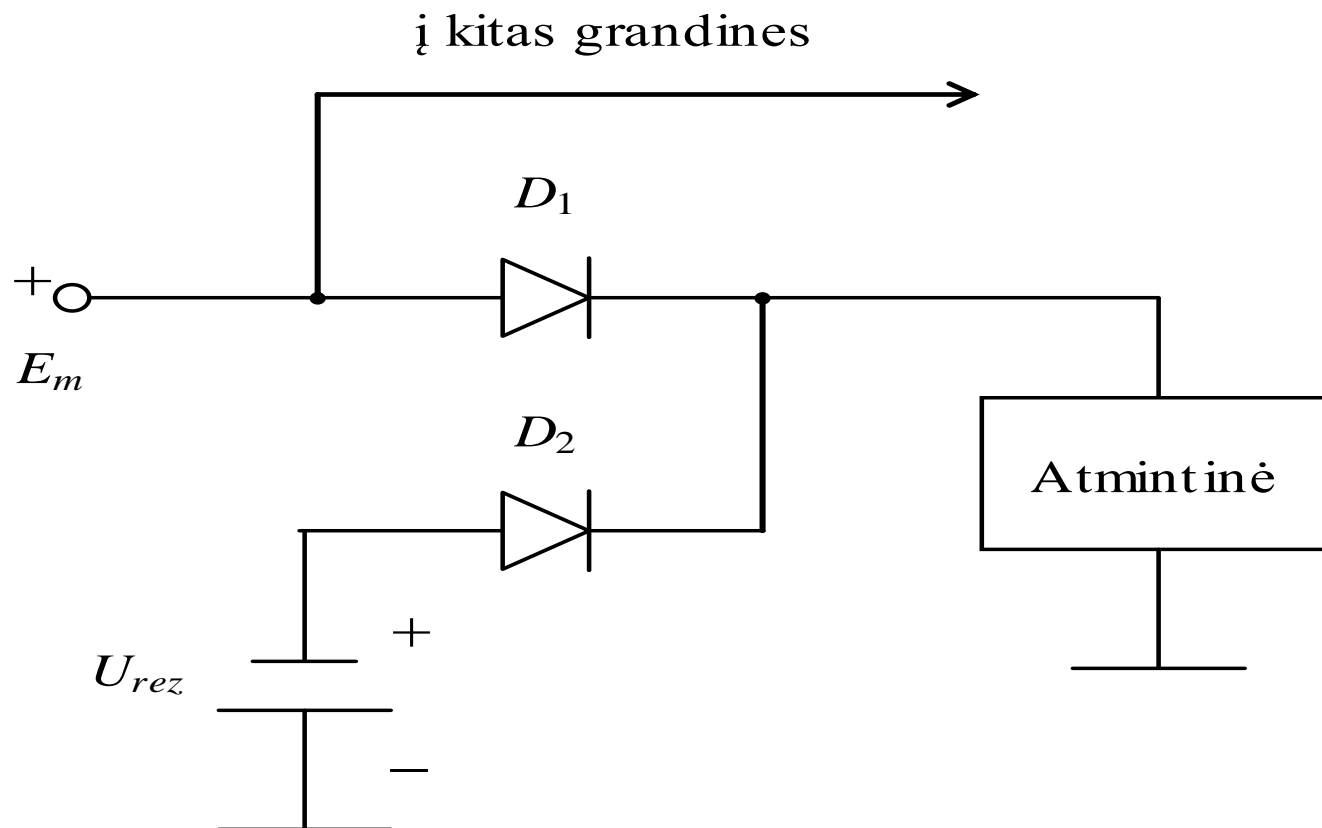
SRAM struktūra

- Tai 2MD struktūros su dvikrypčiais įvesties-išvesties išvadais ($1K \times 4$ talpos) atmintinė.
- Atmintinės matrica 64×64 .
- Reikalingi du adresų dešifratoriai



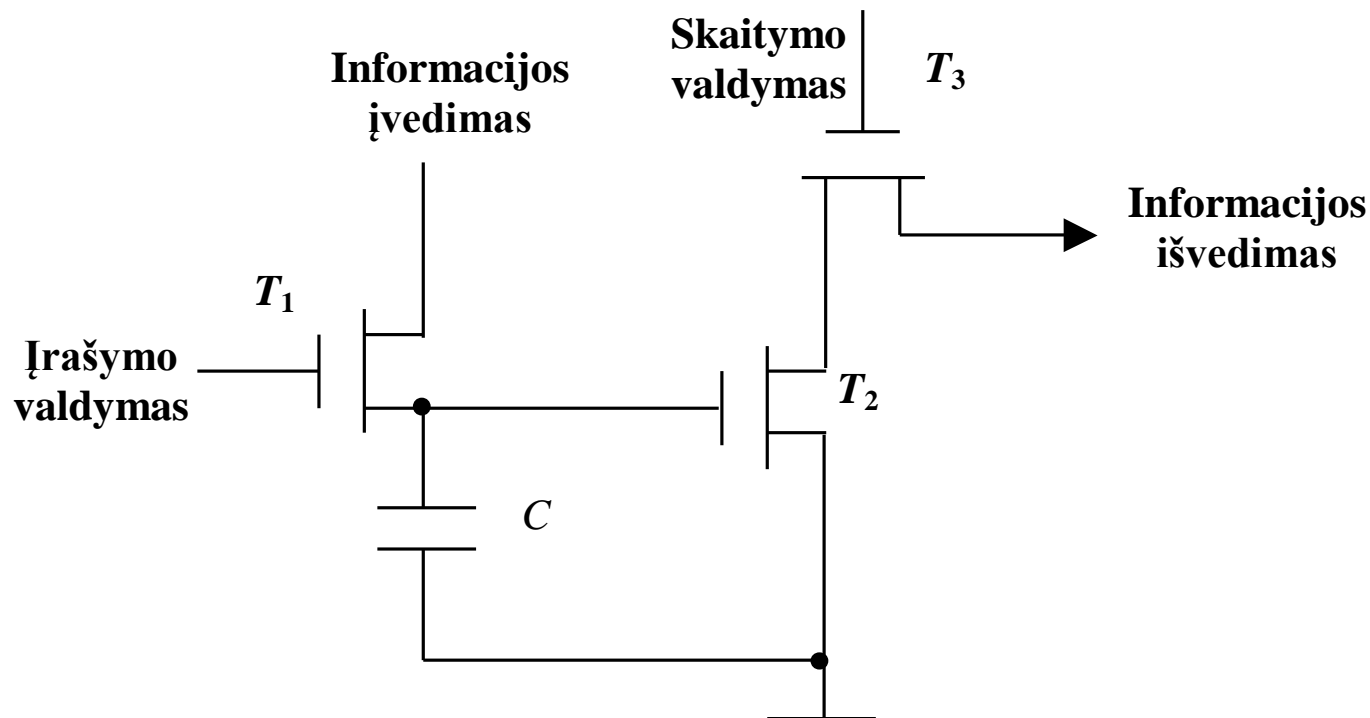
Rezerviniai maitinimo šaltiniai

- Energetinę priklausomybę panaikinti SRAM galima panaudojant rezervinį maitinimo šaltinį su $U_{rez} < E_m$.
- Sumažėjus ar dingus darbinei įtampai, atmintinė maitinama iš rezervinio šaltinio per diodą D_2 .



Dinaminės atmintinės DRAM

- DRAM informacijai saugoti išnaudoja parazitines talpas MOP struktūrose, todėl EAE paprastesni nei triggeriniai ir galima daugiau jų patalpinti kristale (4-6 kartus).
- DRAM talpa praranda krūvį dėl nuotekio, ir saugant duomenis reikia periodiškai regeneruoti informaciją (kas kelios milisekundės).
- EAE gali būti daugi tranzistoriniai ir vientranzistoriniai.

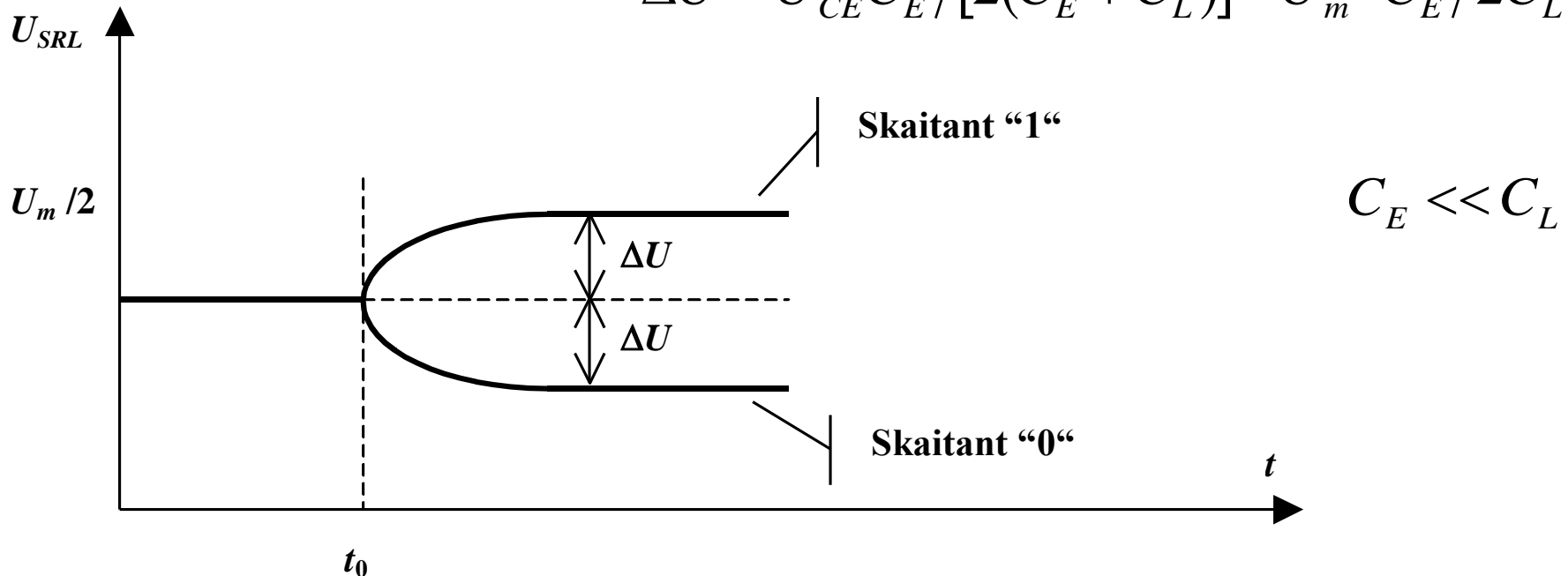


Vientranzistorinių EAE funkcionavimas

- Krūviui pasiskirstant tarp C_L ir C_E , skaitymo/rašymo linijos potencialas kinta dydžiu ΔU .
- Įtampos ΔU reikšmę nesunku apskaičiuoti $Q = C_L \cdot U_m / 2$

$$Q = (C_L + C_E)(U_{CE}/2 - \Delta U)$$

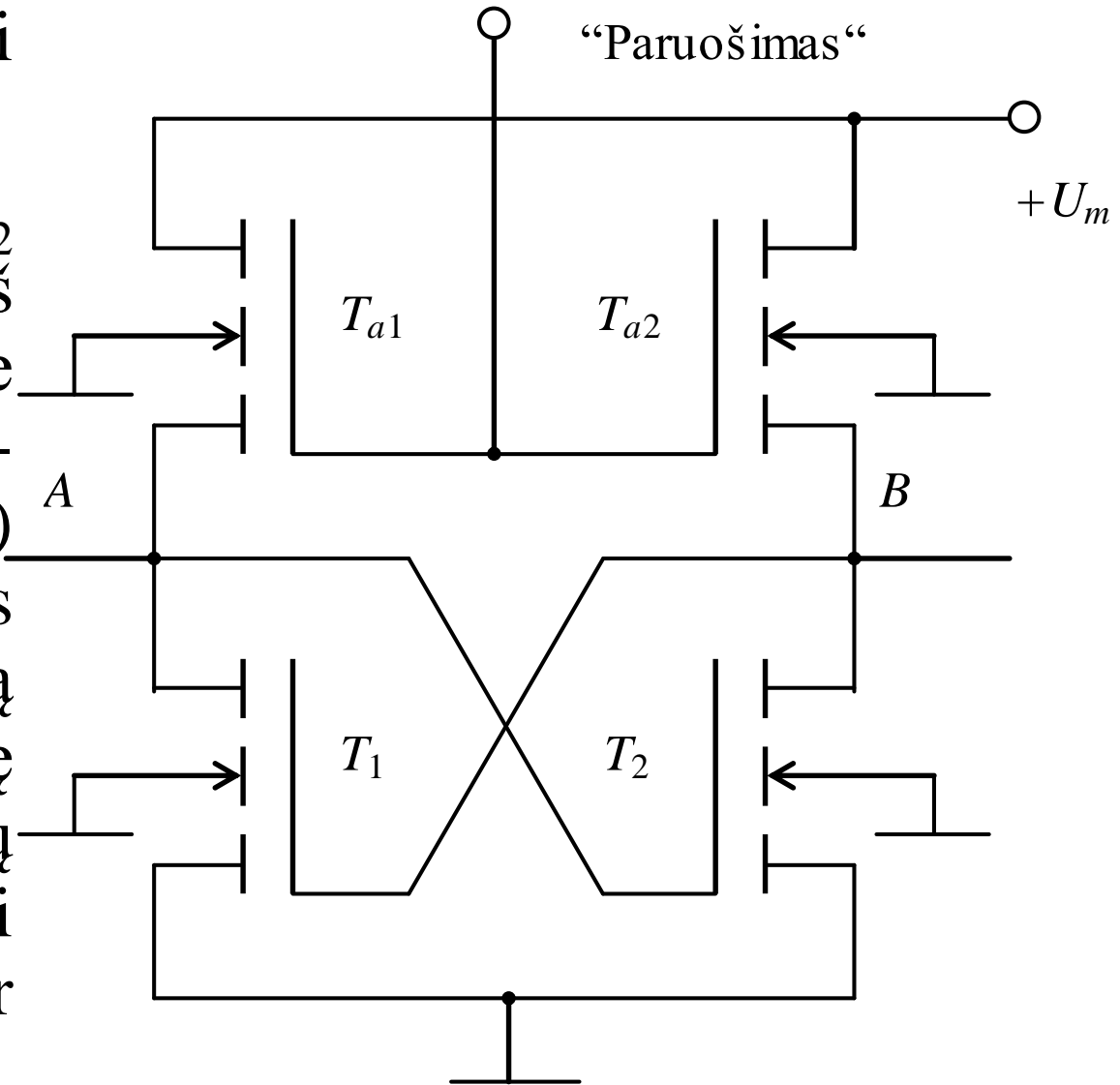
$$\Delta U = U_{CE} C_E / [2(C_E + C_L)] \approx U_m \cdot C_E / 2C_L$$



Stiprintuvas-regeneratorius

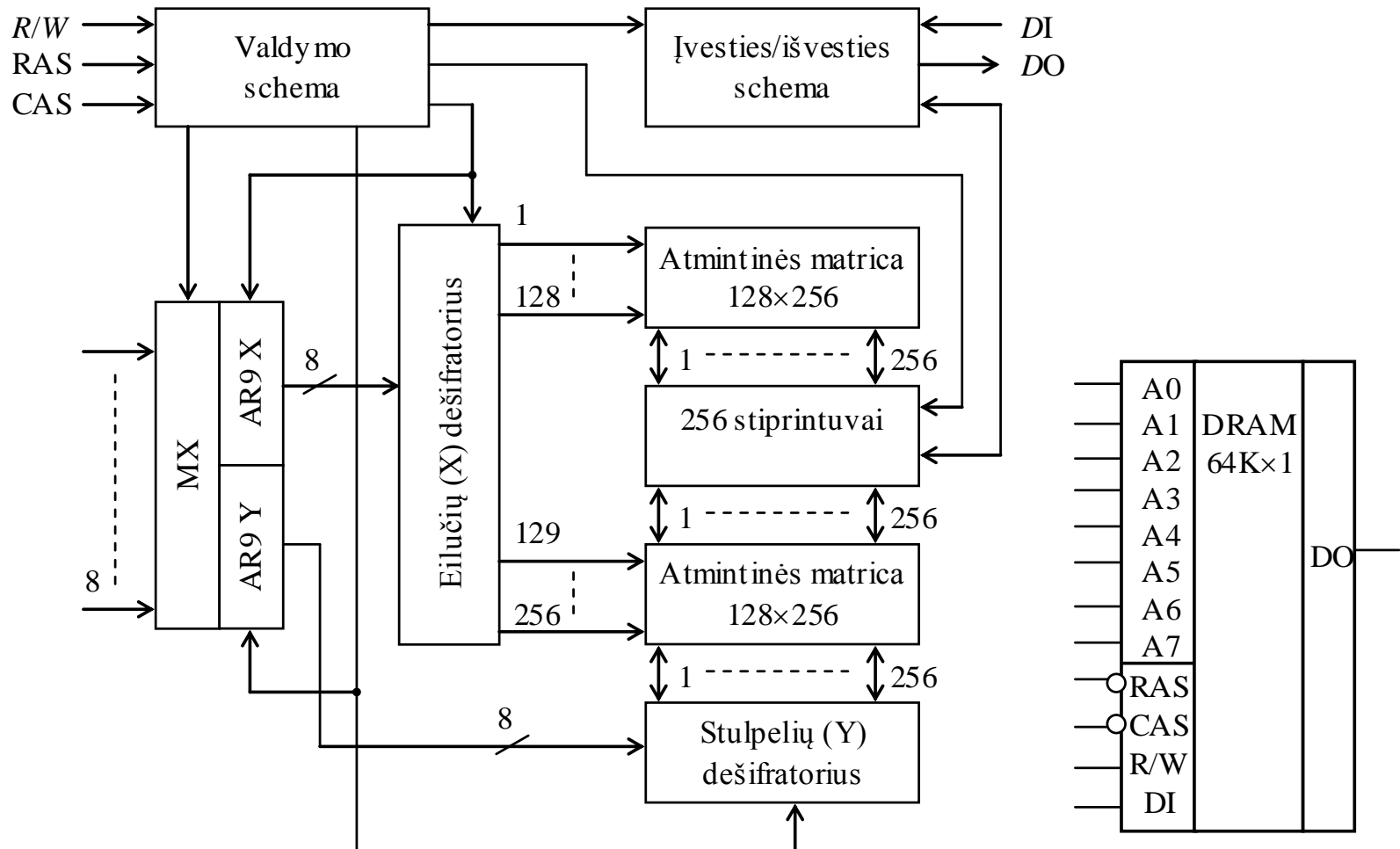
Jis veikia atėjus papildomam signalui „paruošimas“.

Atsidarius T_{a1} bei T_{a2} triggeris T_1 T_2 iš nestabilios (neiškritoje pusėje $U_m/2$, o kitoje $+U_m/2$ a $-U_m/2$) būsenos staigiai pereis į stabilią būseną, kurią nustatys buvę nevienodi tranzistorių T_1 , T_2 laidumai (nuskaitoma ir regeneruojama).



64K×1 DRAM struktūra

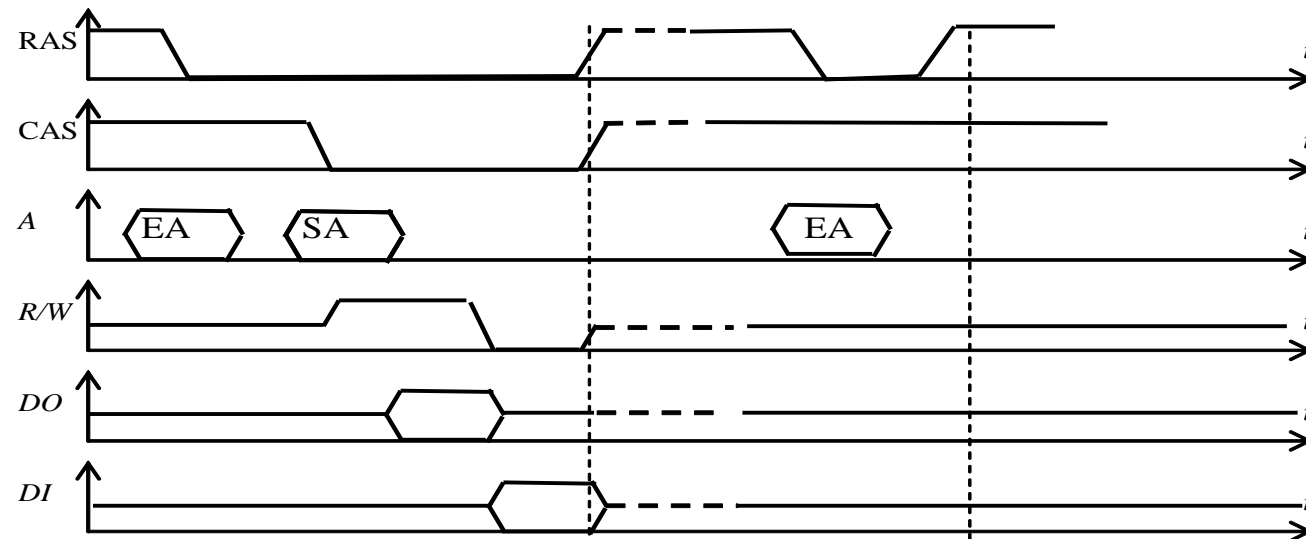
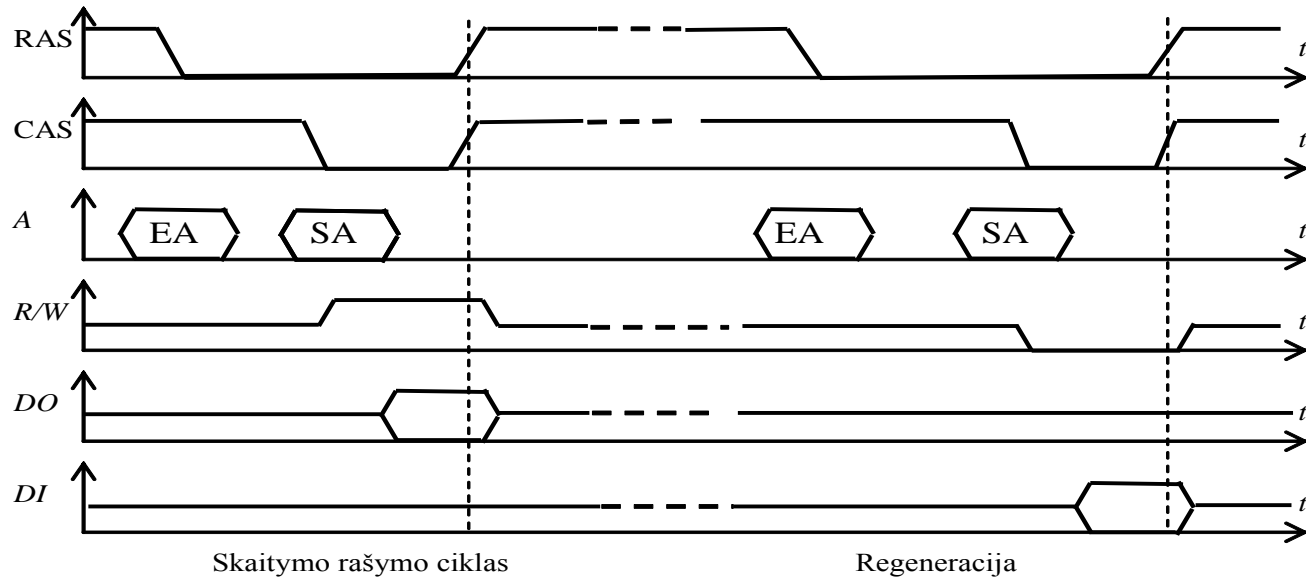
- Demultiplekseris pusinį eilutės adresą nukreipia į AR9X, o stulpelio – į AR9Y registrus.



DRAM funkcionavimas

Skaitymo ciklas

Rašymo ciklas



EA – eilutės adresas SA – stulpelio adresas

- Atmintinė gali dirbi skaitymo, rašymo, skaitymo-rašymo (modifikuotas), regeneravimo režimais.

DRAM funkcionavimas

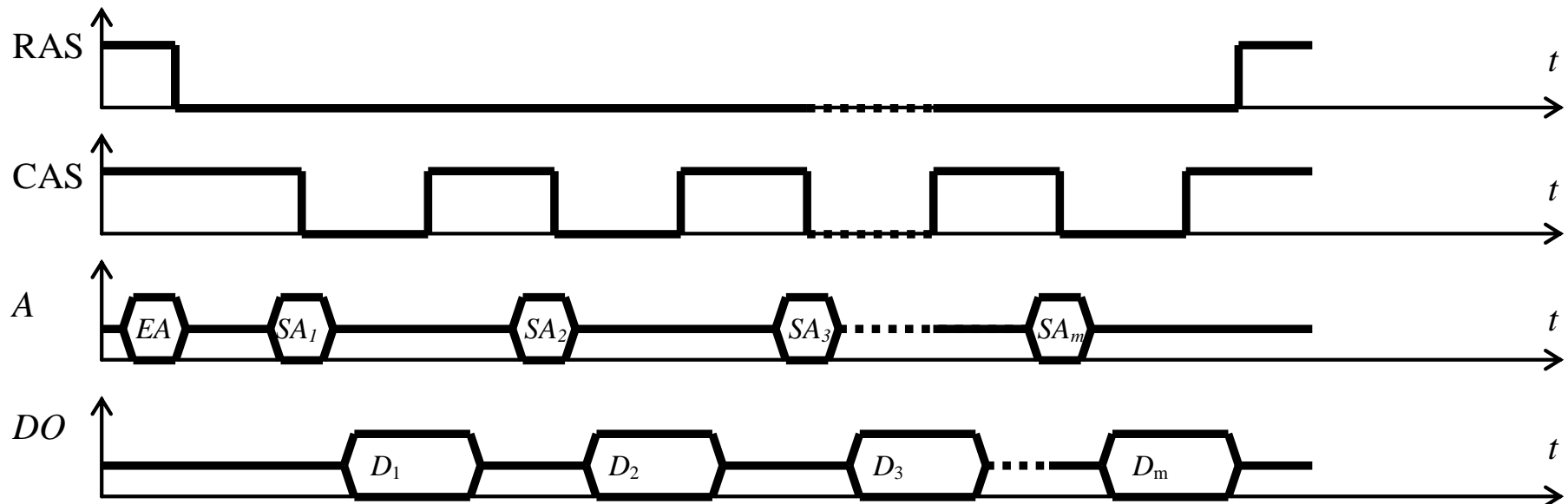
- Eilutės adresas EA nustatomas prieš pasirodant signalui *RAS*, o stulpelio adresas SA prieš pasirodant signalui *CAS*.
- *CAS* signalas atidaro *DO* grandinę duomenis skaitant, kitu metu *DO* yra trečioje (z) būsenoje.
- Į atmintinę informacija iš stiprintuvų įrašoma signalo *RAS* kylančiu frontu.
- Išrinktos eilutės elementariuose atminties elementuose saugojamą informaciją nuskaityto stulpelių stiprintuvai krentančiu *RAS* signalo frontu ir sustiprinta įrašoma kylančiu *RAS* signalo frontu, tai yra bet kurio kreipimosi ciklo metu informacija yra ir regeneruojama visuose išrinktos eilutės elementariuose atminties elementuose.

DRAM savybės

- DRAM atmintinės yra lėtesnės, gaminamos žymiai didesnės spartos, tačiau žymiai pigesnės nei SRAM.
- Naudojamos padidintos veikimo spartos DRAM, kuriose panaudoti veikimo spartą didinantys metodai, išnaudojantys savybę, kai kreipiamasi ne į vieną žodį, o į kelis, kurių adresai gretimi.
- FPM (*Fast Page Mode*) – sparčiosios puslapinio režimo atmintinės, kurių darbo pagrindas – informacijos laikymas toje pačioje eilutėje.

FPM DRAM

- FPM (*Fast Page Mode*) – sparčiosios puslapinio režimo atmintinės, kurių darbo pagrindas – informacijos laikymas toje pačioje eilutėje.
- Pirmiausia EA išrenka atitinkamą eilutę atminties matricoje, o po to SA – stulpelį ir kreipiantis į žodžius toje pačioje eilutėje keičiasi tik SA (ciklų trukmių santykiai yra 5-3-3-3 ir kreipimosi laikas 30÷40 ns).

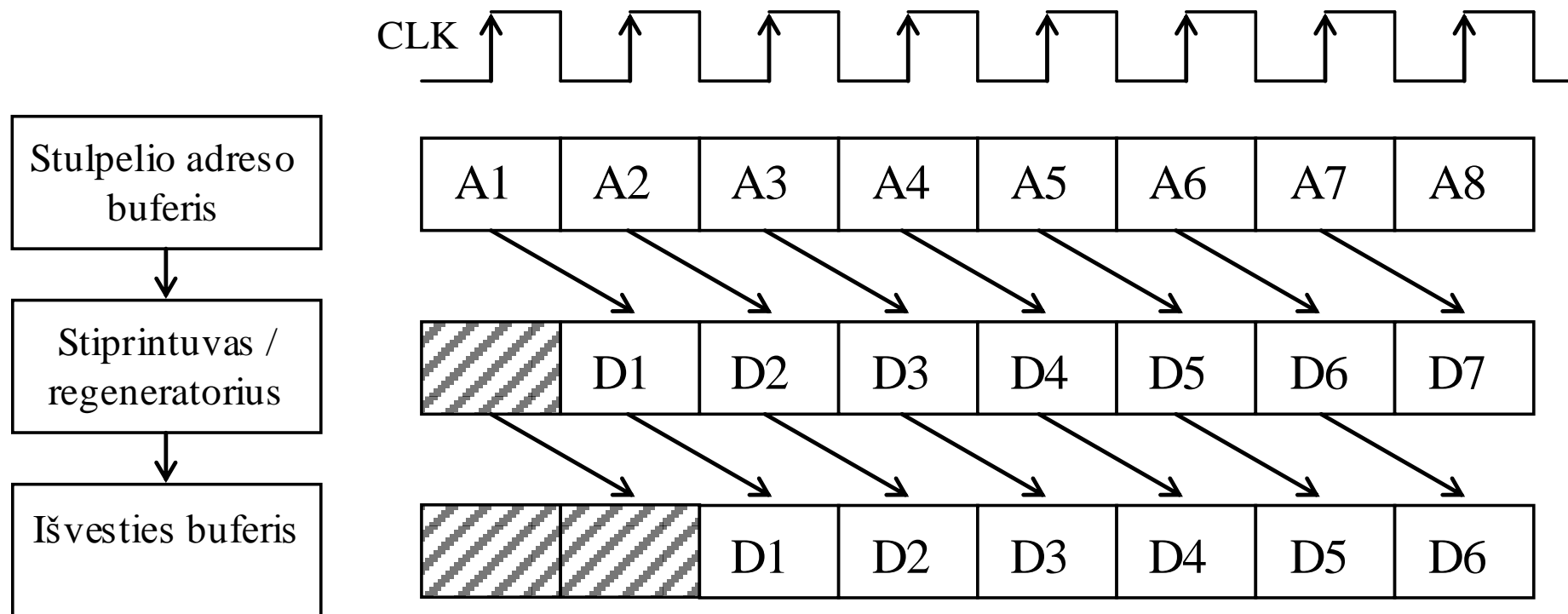


Išplėsto duomenų išvedimo DRAM

- EDORAM (*Extended Data Output RAM*) – išplėsto duomenų išvedimo atmintinės, kuriose informacija stiprintuvuose-regeneratoriuose (statiniam registre) lieka ir pasibaigus CAS (nuskaitant mažesnės trukmės) signalui (5-2-2-2).
- BEDORAM (*Burst EDORAM*) – atmintinė su išplėstu paketiniu kreipimusi, turinti papildomą SA skaitiklį, kuris formuoja SA reikšmės pridedant 1 (5-1-1-1).
- MDRAM (Multibank DRAM) – daugiabankinės atmintinės, kada atmintinė dalinama į bankus, į kuriuos kreipiamasi iš eilės. Kol skaitoma vieno banko informaciją, pasirošama kreiptis į kitą banką, todėl kreipiantis į kitą banką nereikia laukti (atmintinės veikimas spartėja net naudojant tik du bankus - vieną su nelyginiais, o kitą su lyginiais adresais).

Sinchroninės DRAM

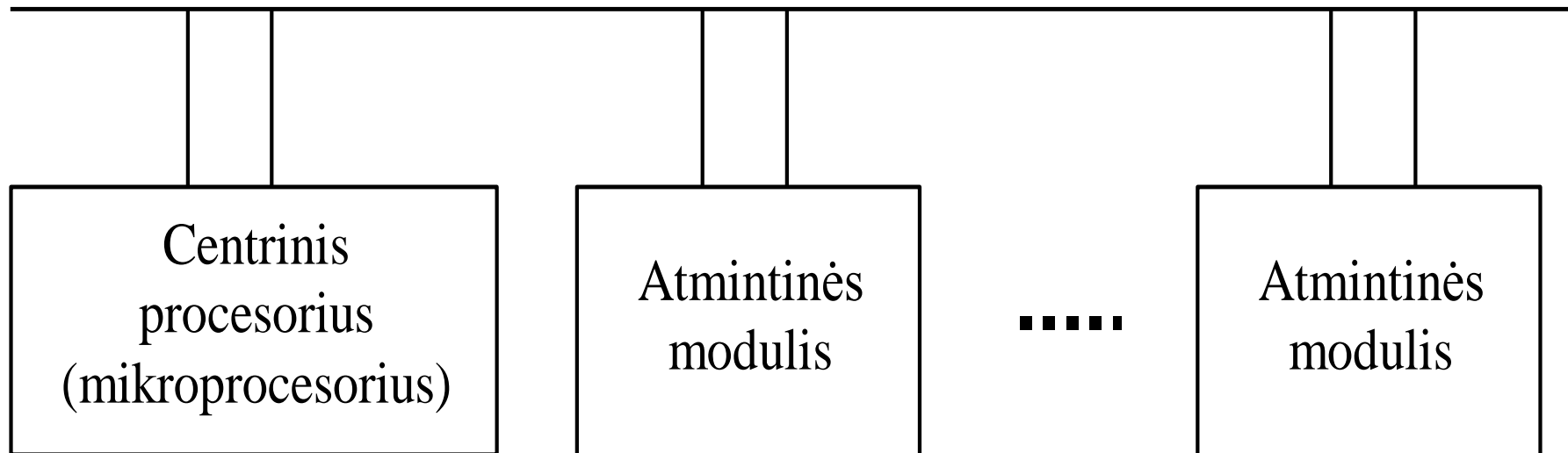
- SDRAM (*Synchronous DRAM*) – sinchroninės atmintinės yra sparčiausios atmintinės, nes jų valdymo signalai susieti su sinchronizavimo signalais ir naudojama konvejerizacija informacijos perdavimo trakte bei MDRAM struktūra. Pirmo žodžio kreipimosi laikas panašus, kaip ir paprastose DRAM atmintinėse tačiau kitų žodžių SA formuoja vidinis skaitiklis, ir informacijos žodžiai išvestyje pasirodo kiekvieno takto metu.



Atmintinės modulis

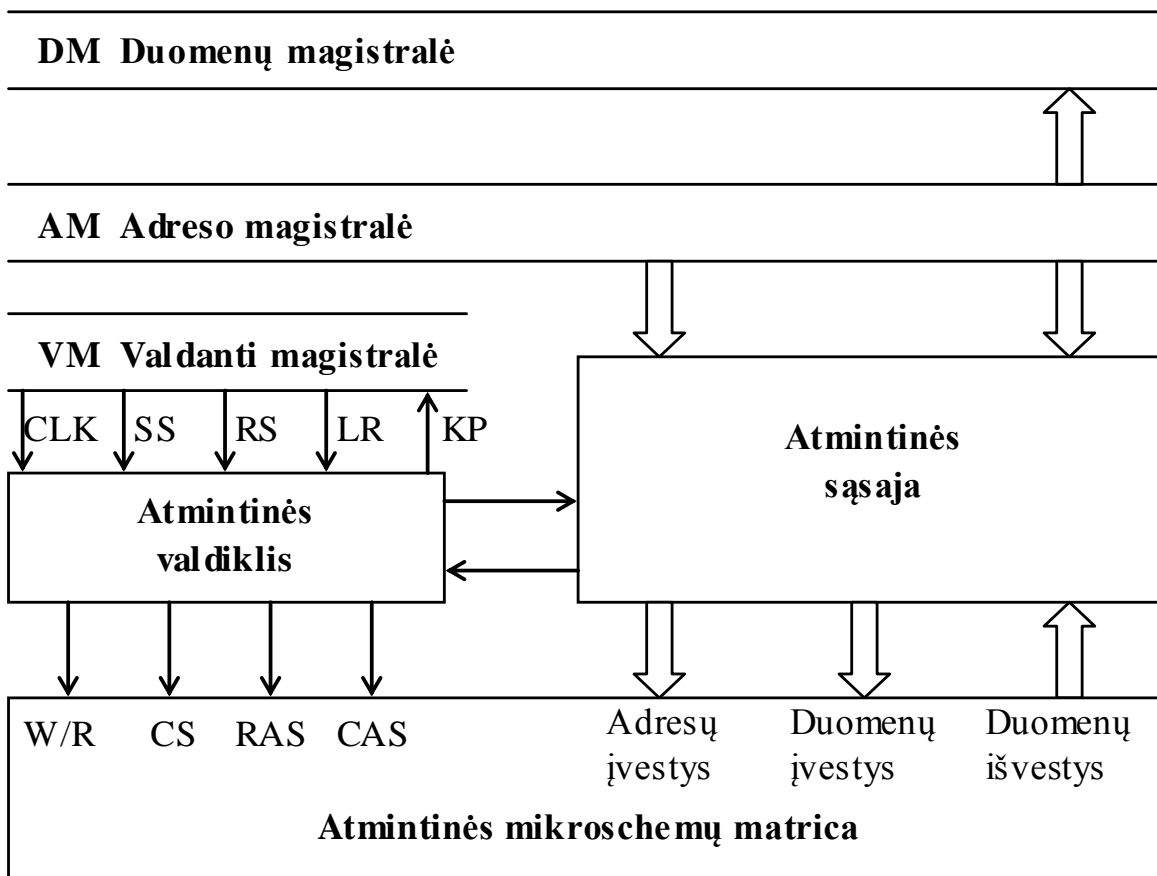
- Paprastai naudojami atmintinių moduliai, o ne atskiros atmintinių mikroschemos, kurie kompiuteryje jungiami prie sisteminės magistralės sudarytos iš duomenų magistralės (DM) ir adreso magistralės (AM).

Kompiuterio sisteminė magistralė



Atmintinės modulio struktūra

- Atmintinės modulį sudaro atmintinės mikroschemų matrica, atmintinės valdiklis, magistralės ir atmintinės sąsaja.
- Iš kompiuterio SS ir RS, LR, CLK ir kitiems signalams, išduodamas kreipties pabaigos signalas KP.

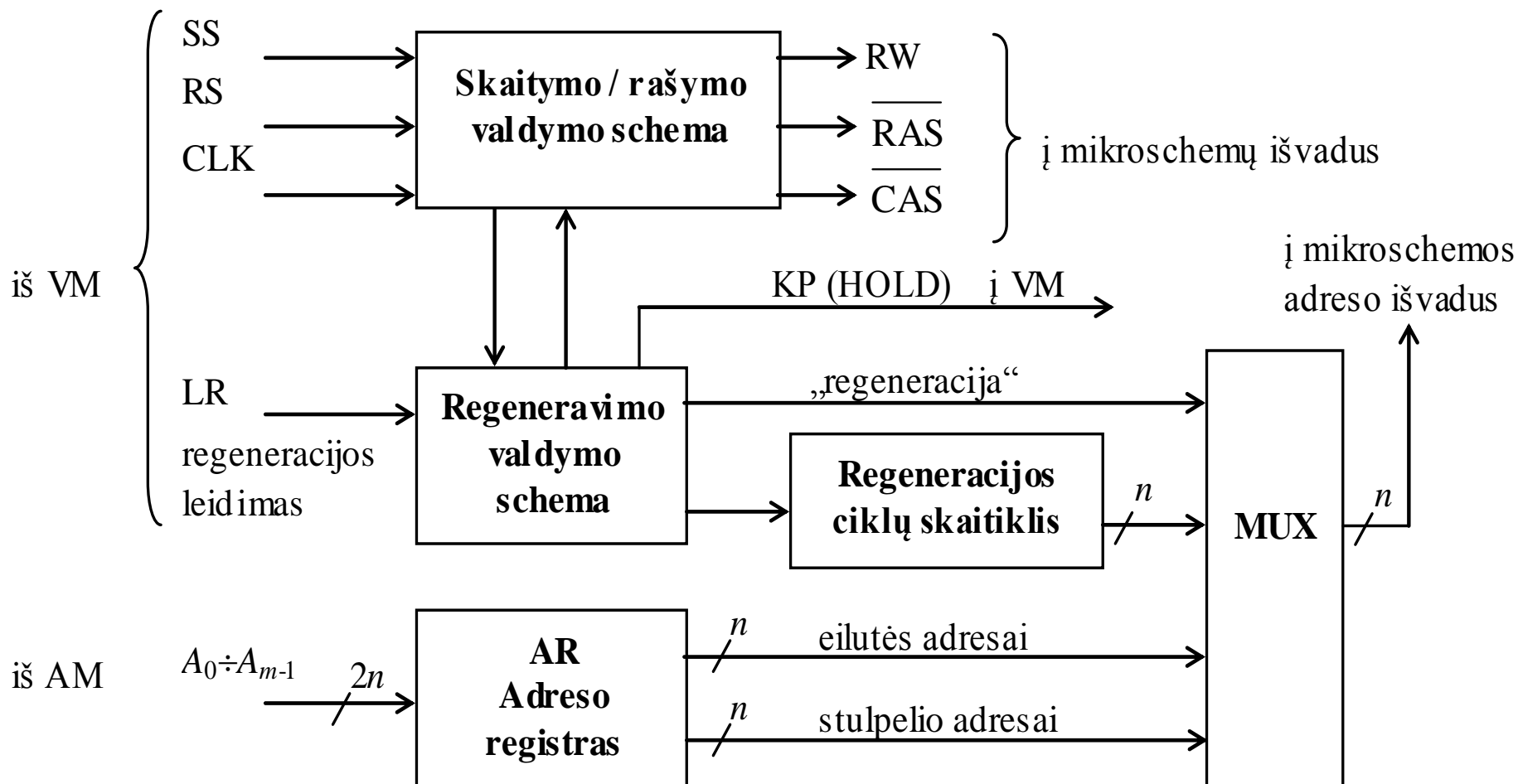


Atminties valdiklis

- Valdiklį sudaro dvi dalys: komandų ir signalų formavimo logika bei sinpchronizavimo schemas.
- Generuoja visus mikroschemų darbui reikalingus signalus (sinchronizavimo ir valdymo) ir išduoda kreipties pabaigos signalą KP bei reikalingus CS (RAS) signalus.
- Organizuojamas informacijos regeneravimas naudojant dinamines atmintines.
- Perduodant duomenis dažnai valdiklis juos kontroliuoja naudodamas lygiškumo (nelygiškumo) kodą, arba ištaiso klaidas (naudodamas Hemingo ar kitokį kodą).

Dinaminės atmintinės modulio valdiklis

- Regeneraciją sinchronizuoja sisteminis signalas CLK arba valdiklio vidinio impulsų generatoriaus signalai.



Atmintinės sąsaja

- Atmintinės sąsaja atlieka šias funkcijas:
- – suderina atmintinės modulio ir magistralės signalus;
- – priima atminties ląstelės adresą iš AM ir duomenis iš DM ir reikiamu momentu bei reikiamais signalų lygiais perduoda į mikroschemas;
- – priima duomenis iš atmintinės mikroschemų ir reikiamu momentu bei reikiamais signalų lygiais perduoda į DM;
- – sinchronizuoja duomenų mainus tarp atmintinės modulio ir kitų įtaisų.

Naudojami atmintinių moduliai

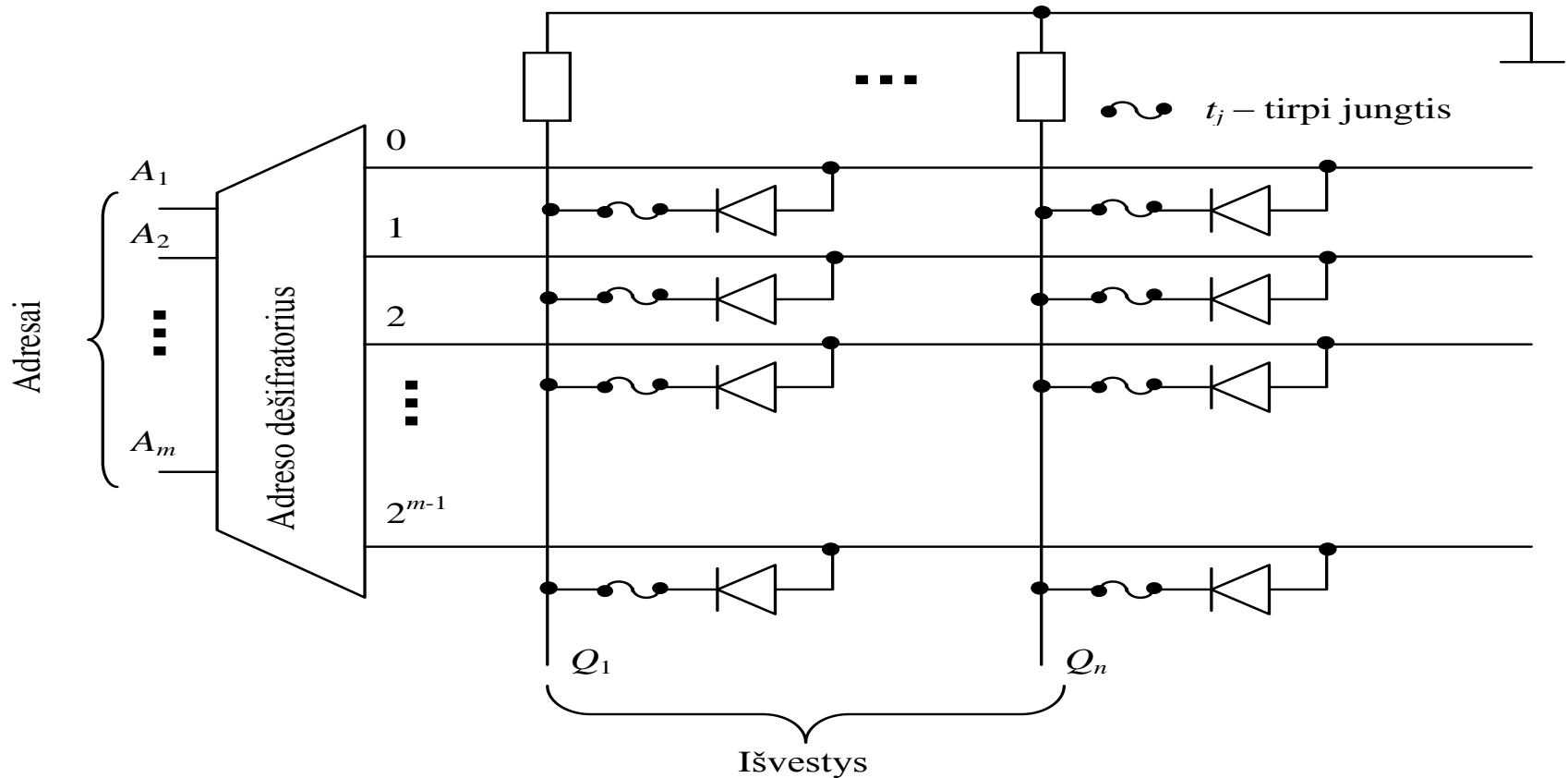
- Atmintinių moduliai yra standartizuoti: SIMM, DIMM ir RIMM.
- Realizuoti nedidelėse spausdintinio montažo plokštelėse.
- SIMM (*Signale in line memory module*) atmintinės žodis 32 bitai (dabar beveik nenaudojami nes reikia dviejų SIMM modulių).
- DIMM (*Dual in line memory module*) atmintinės žodis 64 bitai (plačiausiai naudojamos ir maitinami 5 arba 3,3V įtampa).
- RIMM (*Rambus in line memory module*) atmintinės žodis 64 bitai (brangios, nesuderinamos su DIMM nes turi atskirą sąsają todėl plačiai nenaudojamos).

Pastoviosios atmintinės PA (ROM)

- PA naudojamos nesikeičiančiai arba retai besikeičiančiai informacijai saugoti bei kombinacijų schemų (kodų keitiklių, valdiklių, dekodierių ir kt.) projektavimui.
- Jos gaminamos:
- Tik skaitomos atmintinės (**ROM**) (informacija įrašoma paskutiniuose jų gamybos proceso etapuose panaudojant kaukes).
- Programuojamos pastoviosios atmintinės **PROM** (gaminant sudaromos visos galimos tirpios jungtys, kurias vartotojai specialiais prietaisais ištirpina, tai yra įrašo informaciją).
- Perprogramuojamose atmintinėse **EPROM** (elektriškai įrašoma) ir **EEPROM** (elektriškai įrašoma ir ištrinama) galima visą informaciją ištrinti ir ją pakeisti nauja specialaus proceso metu.

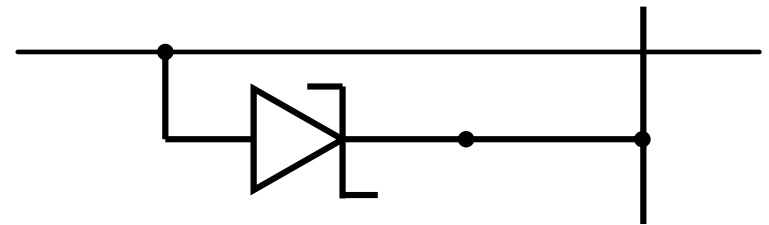
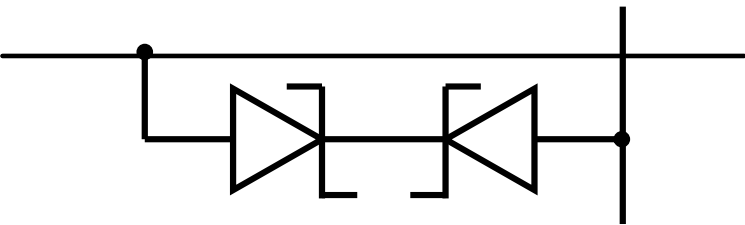
ROM su diodais

- Nuosekliai kiekvienam diodui yra įjungtas tirpus takelis tarp išvesties ir adresinės linijos. Gaminat pašalinamos nereikalingos jungtys arba išvis neįjungimas diodas.



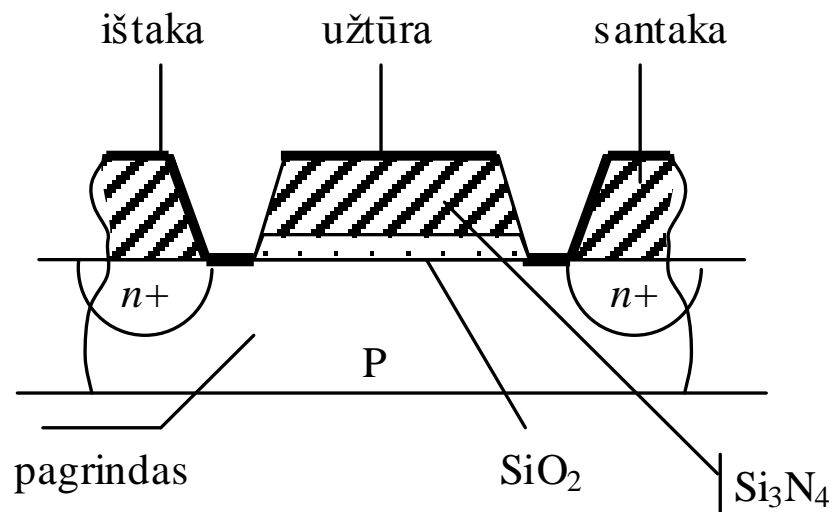
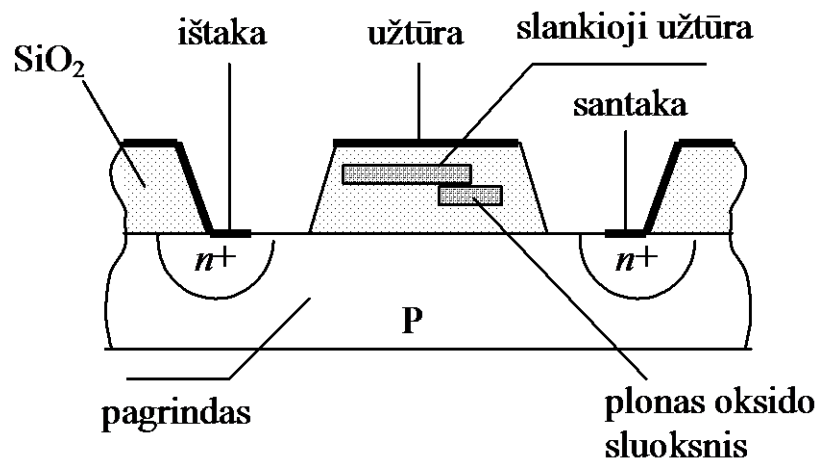
Programuojamos pastoviosios atmintinės PROM

- PROM turi tokią pat struktūrą, kaip ROM atmintinės, tik jas gaminant sudaromos visos galimos jungtys, kurias vartotojai specialiais prietaisais (programatoriais) (įrašo informaciją).
- Kompaktiškesnės yra dilektrinės pramušamos jungtys.
- Naudojami ir atminties elementai su dviem priešinga kryptimi sujungtais diodais saugančiais loginius nulių. Atmintinę programuojant, tai yra įrašant vieneta, paduodama aukštesnė atgalinė įtampa ir diodas pramušamas taip sudarant jungtį.



Perprogramuojamos atmintinės

- EPROM ir EEPROM galima visą informaciją ištrinti ir ją pakeisti nauja specialaus proceso metu.
- EPROM naudojami MOP tranzistoriai su slankia uždūra, o EEPROM MOP tranzistoriai su slankia uždūra arba MNOP su Si_3N_4 .
- Informacijos pakeitimas (ištrynimasis ir naujas įrašymas) užtrunka žymiai ilgiau, nei jos skaitymas.

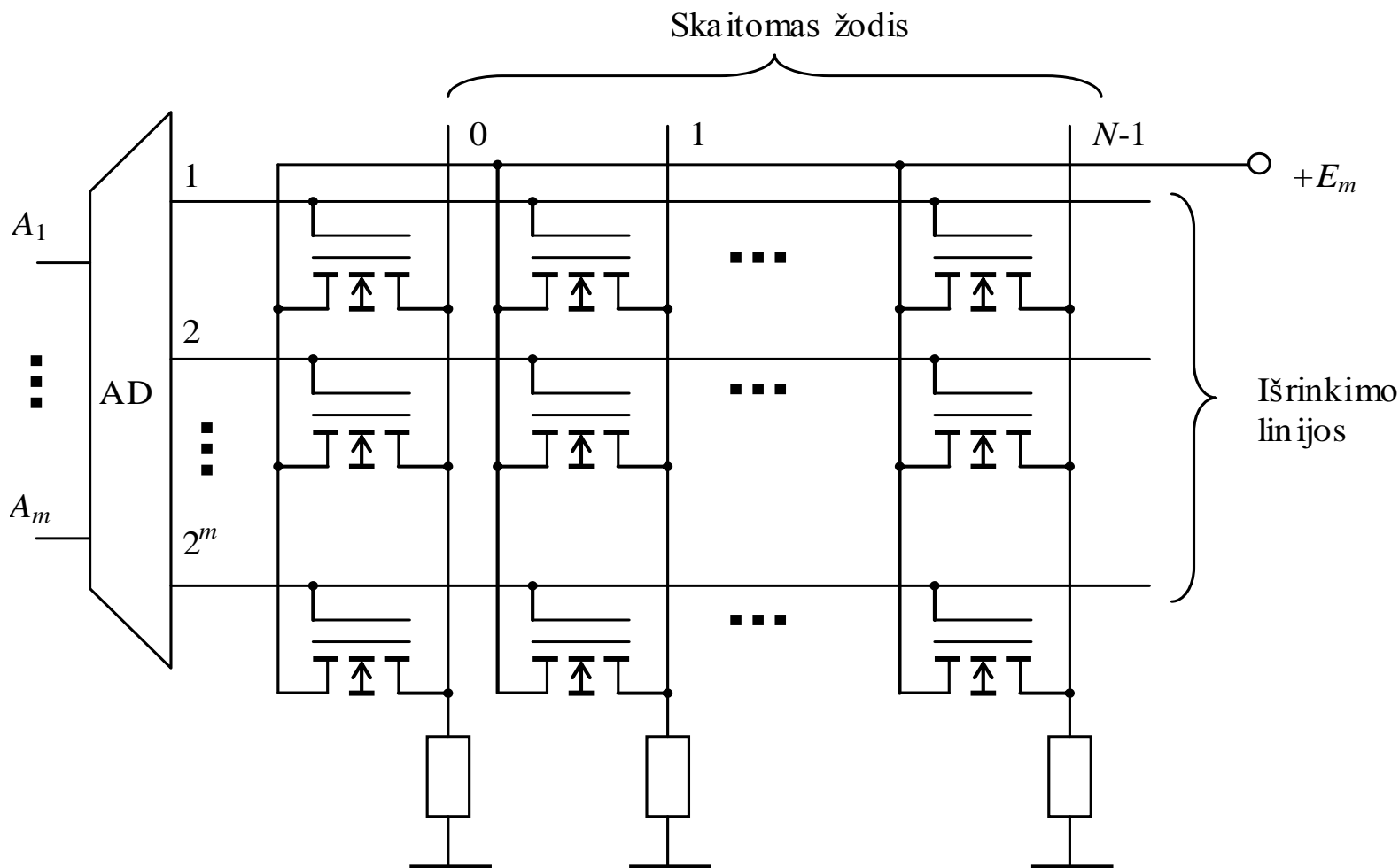


Elektriškai programuojami atminties elementai

- Informacijos nešėjas yra MNOP tranzistorius. Informacija įrašoma padavus į užtūrą įtampą, pakankamą sukurti tunelinį efektą ir perkelti krūvininkus per SiO_2 ir sluoksnių riboje sukaupia krūvį, kuris veikia tranzistoriaus slenkstinę įtampą.
- Slenkstinė įtampa padidėja tiek, kad tranzistoriaus užtūros darbinė įtampa (U^1) jo neatidaro, o jei krūvio nėra arba turi kitokio ženklo (+) krūvį, lengvai atidaromas darbinės įtampos. Tokie tranzistoriai pasižymi didesniu vėlinimu nei MOP su slankia užtūra.
- Norint ištrinti įrašytą informaciją, pakanka paduoti tarp užtūros ir pagrindo priešingos krypties įtampą nei įrašant informaciją.
- Po $10^4 \div 10^6$ informacijos perrašymų MNOP tranzistoriai pradeda sunkiai išsaugoti krūvį, tačiau jos yra energiškai

PA su slankios užtūros MOP tranzistoriais schema

- Atminties elementai (tranzistoriai) į atmintinės matricą jungiami pagal 2D arba 2DM struktūrą.

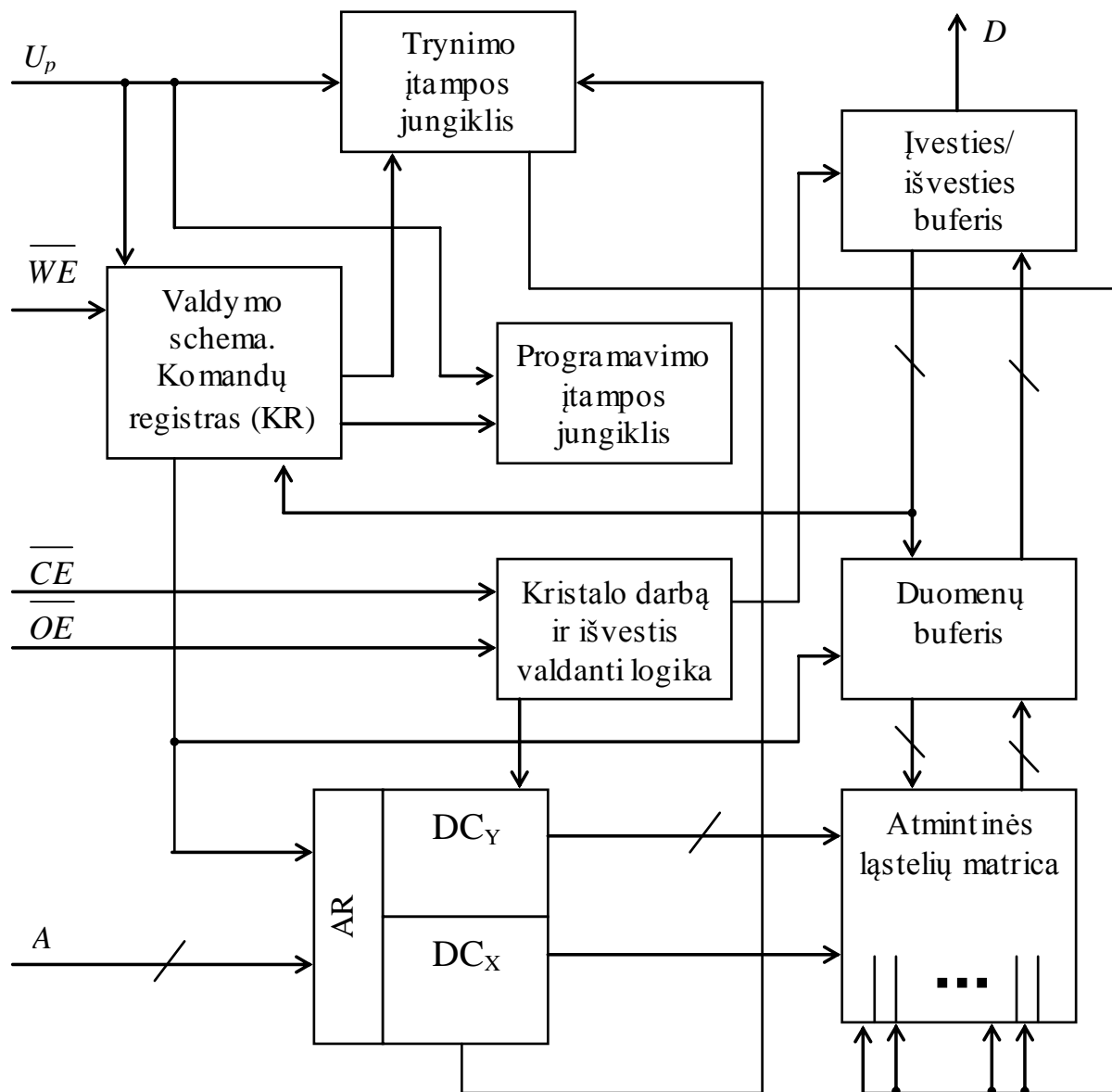


Momentinės atmintinės

- Momentinės atmintinės (*flash*) darbas panašus į elektriniu būdu ištrinamų EEPROM atmintinių, tačiau jų struktūros ypatybės leidžia ištrinti ne visos mikroschemos, o puslapių (didelio bloko) žodžius (256 baitų – 128 Kbaitų).
- Naudojamos tokios priemonės: įrašymo proceso nutraukimas procesoriui kreipiantis į atmintinę; informacijos įrašymo ir skaitymo paspartinimas, organizuojant komandų konvejerį; energijos vartojimo mažinimas nesikreipiant į atmintinę; atmintinėje saugomų žodžių ilgio programavimas (suderinti sąsajas); pritaikyti įvairioms maitinimo įtampoms (5 V, 3,3 V ir kt.); į atmintinės struktūrą įjungti sparčiuosius puslapinius buferius; įvairios apsaugos priemonės (nuo atsitiktinio ar nesankcionuoto kreipimosi bei informacijos ištrynimo); panaudoti mažesnes programavimo įtampas ir užtikrinti dideli informacijos įrašymo ciklu skaičių; sumažinti

Momentinių atmintinių struktūra

- Atmintinės valdomos komandomis, įrašytomis į specialų komandų registrą.
- Įrašymo komanda vykdoma tik padavus įtampą U_p (programavimo) į mikroschemos išvadą.

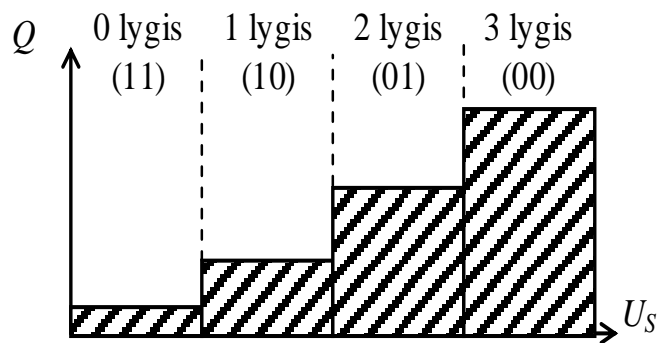


Momentinių atmintinių funkcionavimas

- Komanda įrašoma tik esant žemam U_p lygiui.
- KR turinys nustato įrašymo įtampos U_p jungiklius, darbo režimuos (trynimo, įrašymo ar skaitymo).
- Kai signalų CE, OE ir įtampos U_p žemas lygis, galima skaityti duomenis iš atminties ląstelių matricos ir identifikavimo kodus.
- Kai U_p aukštas lygis – leidžia skaityti informaciją „trinant/rašant“ (leidžia automatiškai suderinti trynimo ir rašymo algoritmus).
- KR neadresuojamas, kai U_p aukštas lygis, o sumažėjus U_p lygiui KR turinys nustatomas į nulį ir mikroschema gali tik skaityti atmintinės informaciją.
- Trinant puslapio baitai ištrinami ir patikrinama (taip pat ir rašant) ar ištrinta (įrašyta).

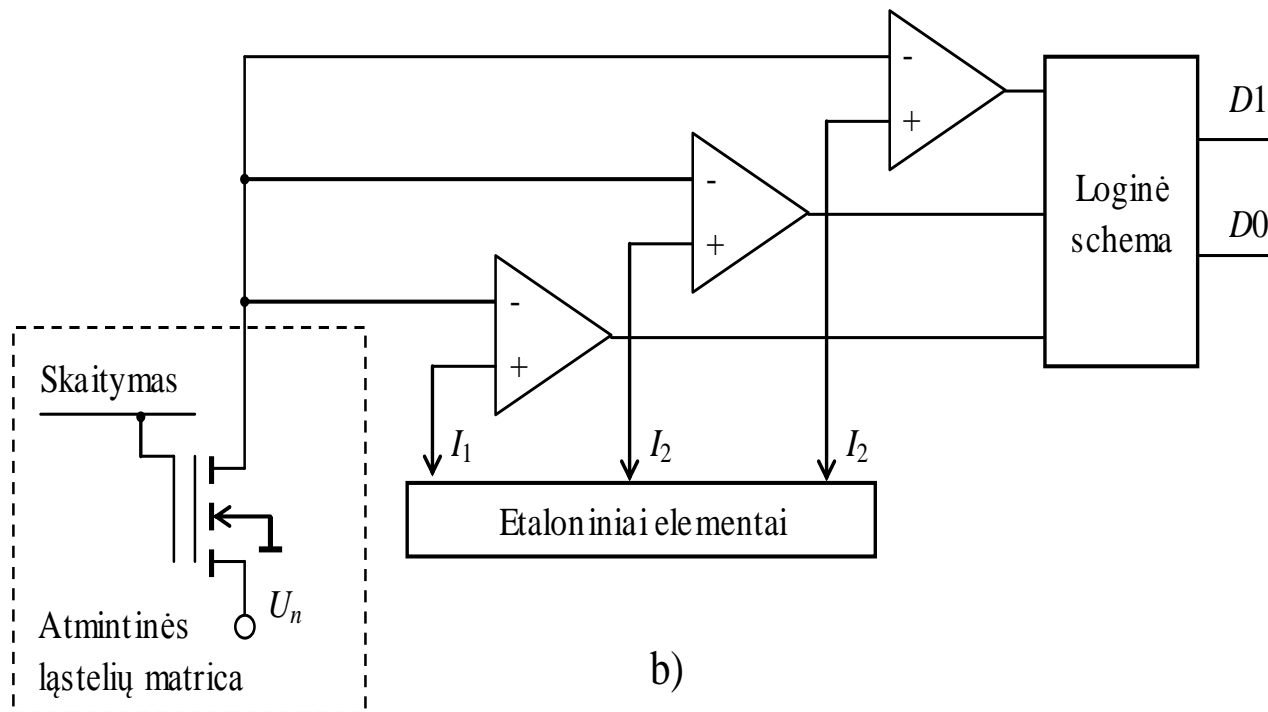
Keturių lygių atminties elementai

- Slankioje užtūroje fiksuojamas krūvio dydis.
- Fiksavus 4 lygius, viename elemente saugomi du bitai.
- Įrašant informaciją įvedamas vienas iš 4 galimų krūvių kiekių.



a)

EAE krūvis	Skaitymo stiprintuvų išvestys			Išvesčių duomenys
	1	2	3	
Q 1 1	1	1	1	1 1
Q 1 0	0	1	1	1 0
Q 0 1	0	0	1	0 1
Q 0 0	0	0	0	0 0



b)

Pastoviųjų atmintinių naudojimas

- Pagrindinė atmintinių paskirtis saugoti informaciją. Tačiau pastoviosios atmintinės gali būti naudojamos ir informacijos apdorojimo uždavinių sprendimui:
 - – realizuoti logines funkcijas;
 - – realizuoti baigtinius automatus;
 - – atlikti aritmetines operacijas ir funkcines priklausomybes.
- • PA realizuojama tobulos disjunktyvinės normalinės formos (DNF) funkcijos (nereikia jų minimizuoti, o išskleisti iki tobulos DNF).
- • Kai kuriais atvejais geriau naudoti PA, o ne loginius elementus.
- • Schemos vėlinimo laikas lygus informacijos skaitymo iš pastoviosios atmintinės laikui.

Baigtinis automatas su PA

- Pastoviąsias atmintines galima panaudoti vietoje kombinacinių schemų.

