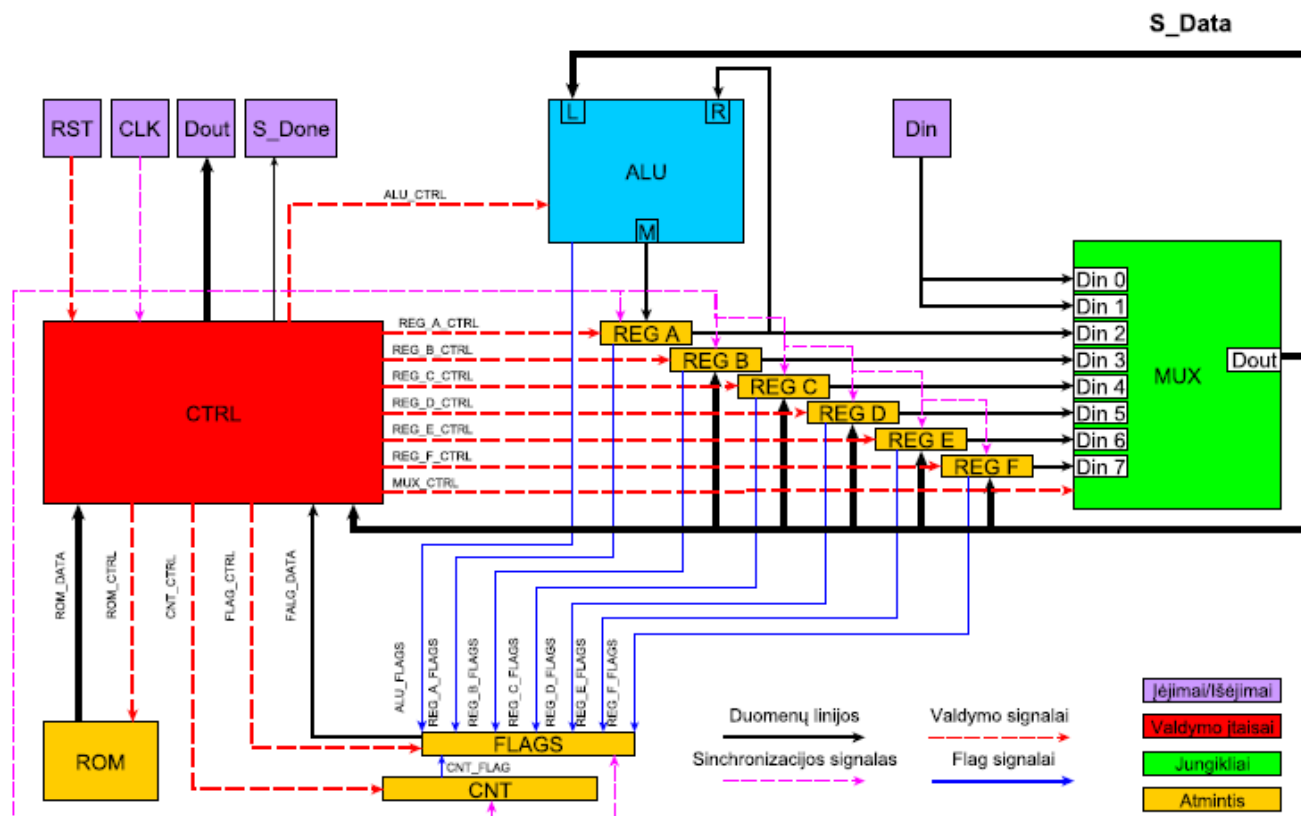


Mikroprogramuojama struktūra

Laboratoriniam darbui paruoštas mikroprogramuojamos struktūros, kuri parodyta 1 paveiksle, modelis VHDL kalba.



1 pav. Mikroprogramuojama struktūra

1 paveiksle pateikta struktūra, kurioje parodyti pagrindiniai jos komponentai ir jų sąryšiai. Tai supaprastinto procesoriaus variantas. Su išorine aplinka šis procesorius siejamas tokiais ryšiais (paveiksle jie pažymėti alyvine spalva):

- **RST** – struktūros komponentų nustatymo į pradinę būseną signalas;
- **CLK** – sinchronizacijos signalas;
- **Din** – 16 bitų duomenų įvestis;
- **Dout** – 16 bitų duomenų išvestis;
- **S_Done** – darbo pabaigos signalas.

Procesorių sudaro šie komponentai:

- valdymo signalų formavimo schema **CTRL**,
- aritmetinis loginis įtaisas **ALU**,
- mikrokomandų atmintis **ROM**,
- 6 bendros paskirties registrai **REG A—REG F**,

- multiplekseris **MUX**,
- skaitiklis **CNT**,
- požymių (loginių sąlygų) registras **FLAG**.

1. Aritmetinis loginis įtaisas (ALU)

Tai pagrindines aritmetines ir loginę operacijas atliekanti kombinacinė schema. ALU turi dvi duomenų įvestis: kairiąją (L) ir dešiniąją (R), bei rezultato išvestį (M). Pilnas ALU įvesčių ir išvesčių sąrašas pateikiamas lentelėje 1.2.

1 lentelė. ALU įvestys ir išvestys

<i>Įvestis</i>	<i>Paskirtis</i>
<i>ALU_CMD</i>	Komanda (5 bitų)
<i>ALU_Din_L</i>	Kairioji duomenų įvestis (16 bitų)
<i>ALU_Din_R</i>	Dešinioji duomenų įvestis (16 bitų)
<i>Išvestis</i>	<i>Paskirtis</i>
<i>FLG_ALU_CMD</i>	Pernaša, išvedama į požymių registrą
<i>ALU_Dout</i>	Rezultato išvestis (16 bitų)

ALU atliekamos operacijos yra apibrėžtos 2 lentelėje. Vieno ciklo metu ALU gali atlikti tik vieną aritmetinę operaciją. Jei vienoje mikrokomandoje bus pažymėta daugiau nei viena ALU operacija, ALU įvykdys tą, kurios kodas yra didžiausias.

2 lentelė. ALU atliekamos operacijos

<i>ALU_CMD</i>	<i>Komanda</i>	<i>Operacija</i>
000	$M = L + R$	Sudėtis
001	$M = \overline{L}$	Invertuojamas L
010	$M = \overline{R}$	Invertuojamas R
011	$M = L + 1$	L padidinimas vienetu
100	$M = L - 1$	L sumažinimas vienetu
101	$M = R + 1$	R padidinimas vienetu
110	$M = R - 1$	R sumažinimas vienetu
111	$M = L \oplus R$	L xor R

Pastaba. Modelyje realizuoto ALU operacijos išrinkimo bitai yra penki. 2 lentelėje nurodytoms operacijoms išrinkti yra panaudoti tik paskutiniai trys bitai; naudojant visus 5 bitus, ALU galėtų atlikti 32 skirtingas operacijas. Ši išplėtimo galimybė skirta studentams, kurie norėtų papildyti ALU operacijų sąrašą pagal individualius poreikius.

2. Registrai

Procesoriuje yra 6 universalūs 16 bitų registrai, identifikuojami raidėmis **A, B, C, D, E, F**. Registrų įvestys ir išvestys yra apibrėžtos 3 lentelėje.

3 lentelė. Registrų įvestys ir išvestys

Įvestys	Paskirtis
<i>CLK</i>	Sinchronizavimo signalas
<i>RST</i>	Reset signalas
<i>REG_Din</i>	Duomenų įvestis (16 bitų)
<i>REG_CMD</i>	Komanda (3 bitai)

Išvestys	Paskirtis
<i>REG_FLAG_H</i>	15-o bito reikšmė (į požymių registrą)
<i>REG_FLAG_L</i>	0-o bito reikšmė (į požymių registrą)
<i>REG_Dout</i>	Duomenų išvestis (16 bitų)

Registras **REG A** atlieka akumulatoriaus funkciją. Į jį duomenys gali būti įrašomi tik iš ALU išėjimo. Registro **REG A** išėjimas yra tiesiogiai sujungtas su dešiniuoju (**R**) ALU įėjimu ir multiplekserio **Din 2** įėjimu. Registrų **REG B-REG F** duomenų įėjimai yra prijungti prie pagrindinės duomenų magistralės, o išėjimai atitinkamai prie multiplekserio **Din 2 – Din 7** įėjimų.

Kiekvienas registras gali atlikti 8 operacijas; jos ir jas inicijuojančios valdymo signalų reikšmės pateiktos 4 lentelėje.

4 lentelė. Registrų atliekamos operacijos

REG_CMD	Operacija	Veiksmas
000	$A = A$	Saugojimas
001	$A = Din$	Duomenų įvedimas
010	$A = LL1(A)$	Loginis postūmis į kairę per 1 bitą
011	$A = LR1(A)$	Loginis postūmis į dešinę per 1 bitą
100	$A = AL1(A)$	Aritmetinis postūmis į kairę per 1 bitą
101	$A = AR1(A)$	Aritmetinis postūmis į dešinę per 1 bitą
110	$A = CL1(A)$	Ciklinis postūmis į kairę per 1 bitą
111	$A = CR1(A)$	Ciklinis postūmis į dešinę per 1 bitą

3. Multiplekseris

Multiplekseris (MUX) yra įtaisas, turintis 8 duomenų ir 1 valdymo įėjimus, bei 1 duomenų išvestį. Visi MUX įėjimai ir išėjimai yra apibrėžti 5 lentelėje. Valdymo signalų reikšmės yra pateiktos 6 lentelėje.

Priklausomai nuo valdymo signalų, vienas iš įėjimų yra perduodamas į išėjimą.

4. Skaitiklis

Skaitiklis (**CNT**) yra 16 bitų registras, kuris sumažina savo turinį vienetu, gavęs valdymo signalą **CNT_CMD**. Skaitiklio įėjimai ir išėjimai yra apibrėžti 7 lentelėje. Skaitiklis yra naudojamas ciklams realizuoti mikroprogramoje. Kai skaitiklio turinys pasiekia kritinę reikšmę, skaitiklis išduoda signalą į požymių registro 13-ą bitą. Pradinė ir kritinės reikšmės yra

nurodomos skaitiklio programiniame kode.

5 lentelė. Multiplekserio įėjimai ir išėjimai

<i>Įvestys</i>	<i>Paskirtis</i>
<i>MUX_{CMD}</i>	Sinchronizavimo signalas
<i>MUX_Din0</i>	16 bitų duomenų įvestis
<i>MUX_Din1</i>	16 bitų duomenų įvestis
<i>MUX_Din2</i>	16 bitų duomenų įvestis
<i>MUX_Din3</i>	16 bitų duomenų įvestis
<i>MUX_Din4</i>	16 bitų duomenų įvestis
<i>MUX_Din5</i>	16 bitų duomenų įvestis
<i>MUX_Din6</i>	16 bitų duomenų įvestis
<i>MUX_Din7</i>	16 bitų duomenų įvestis
<i>Išvestis</i>	<i>Paskirtis</i>
<i>MUX_Dout</i>	16 bitų duomenų išvestis

6 lentelė. Multiplekserio operacijos

<i>MUX_CMD</i>	<i>Operacija</i>
000	S_Data = Din
001	S_Data = Din
010	S_Data = REG_A_Dout
011	S_Data = REG_B_Dout
100	S_Data = REG_C_Dout
101	S_Data = REG_D_Dout
110	S_Data = REG_E_Dout
111	S_Data = REG_F_Dout

7 lentelė. Skaitiklio įėjimai ir išėjimai

<i>Įvestys</i>	<i>Paskirtis</i>
<i>CLK</i>	Sinchronizavimo signalas
<i>CNT_CMD</i>	Skaitiklio turinio mažinimas
<i>RST</i>	Reset signalas
<i>Išvestis</i>	<i>Paskirtis</i>
<i>CNT_Flag</i>	Požymio signalas

5. Valdymo signalų formavimo schema

Ši schema, 1 paveiksle pažymėta **CTRL** simboliu, organizuoja procesoriaus darbą pagal vartotojo sukurtą mikroprogramą, esančią pastoviojoje atmintyje **ROM**. Kiekvieno takto pradžioje **CTRL** iš **ROM** išrenka vieną mikrokomandą (MK) ir į struktūros komponentus perduoda atitinkamus valdymo signalus. MK struktūra priklauso nuo naudojamo MK adresavimo būdo (žr. dokumentą „Mikroprogramavimo pagrindai.docx“).

8 lentelėje pateiktos visos valdymo signalų formavimo schemos įvestys ir išvestys.

8 lentelė. Valdymo signalų formavimo schemos įvestys ir išvestys

<i>Įvestys</i>	<i>Paskirtis</i>
<i>CLK</i>	Sinchronizavimo signalas
<i>RST</i>	Reset signalas
<i>ROM_Din</i>	MK komandos įvestis iš ROM
<i>Din</i>	16 bitų duomenų įvestis
<i>FlagV</i>	Požymių registro vektoriaus įvestis

<i>Išvestys</i>	<i>Paskirtis</i>
<i>CNT_CMD</i>	Skaitiklio valdymo signalas
<i>MUX_CMD</i>	3 bitų MUX valdymo signalas
<i>ALU_CMD</i>	5 bitų ALU valdymo signalas
<i>REG_A_CMD</i>	3 A registro valdymo signalas
<i>REG_B_CMD</i>	3 B registro valdymo signalas
<i>REG_C_CMD</i>	3 C registro valdymo signalas
<i>REG_D_CMD</i>	3 D registro valdymo signalas
<i>REG_E_CMD</i>	3 E registro valdymo signalas
<i>REG_F_CMD</i>	3 F registro valdymo signalas
<i>RST_COMP</i>	9 Reset signalai į individualius komponentus
<i>Done</i>	Mikroprogramos vykdymo pabaigos signalas
<i>CTRL_Dout</i>	16 bitų rezultato išvedimas

6. Mikrokomandų atmintis

Mikrokomandų pastovioji atmintis (**ROM**), yra sistemos elementas, kuriame patalpinta vykdomoji mikroprograma. **ROM** turi **RST_ROM**, **ROM_CMD** įėjimus ir **ROM_DATA** duomenų išėjimą. Aukšto lygio **RST_ROM** signalu nuskaityta adresu 0 esanti mikrokomanda.

ROM_CMD nurodo, kelinta atminties eilutė (mikrokomanda) bus perduota į **ROM_DATA** išėjimą.

ROM žodžio (mikrokomandos) ilgis priklauso nuo MO skaičiaus ir naudojamos adresavimo būdo. Jei naudojama priverstinė adresacija, mikrokomandos yra lygus MO, LS ir NA koduojančių bitų skaičiaus sumai. Natūralios adresacijos atveju **ROM** mikrokomandos yra lygus MO skaičiui +1.

Modelyje maksimalus **ROM** žodžių skaičius – 256 (tai maksimalus mikroprogramos ilgis).

7. Požymių registras

Požymių registras (**FLAG**) yra komponentas, fiksuojantis loginių sąlygų reikšmes. Šio registro bitų paskirtis pateikta 9 lentelėje. *FLAG(1)* bitas parodo, jog A registro 15 (kairiausias – *REG_A_H*) bitas lygus 1, kas rodo, jog registre yra įrašytas neigiamas skaičius, ir yra naudingas, jei reikia tikrinti skaičiaus ženklą. *FLAG(1)* bitas parodo, jog A registro

žemiausioji (dešiniausioji – *REG_A_L*) skiltis lygi 1. Šis bitas yra naudojamas tikrinant žemiausiąją skaičiaus skiltį.

Analogiškai požymių registro 3-11 bitai atitinka registrų B-F aukščiausių ir žemiausių skilčių reikšmes.

Požymių registro įėjimo ir išėjimo signalai apibrėžti 10 lentelėje.

9 lentelė. Požymių registro bitai

<i>RST_COMP</i> bitas	<i>Paskirtis</i>
<i>FLAG</i> (1)	<i>REG_A</i> (15)
<i>FLAG</i> (2)	<i>REG_A</i> (0)
<i>FLAG</i> (3)	<i>REG_B</i> (15)
<i>FLAG</i> (4)	<i>REG_B</i> (0)
<i>FLAG</i> (5)	<i>REG_C</i> (15)
<i>FLAG</i> (6)	<i>REG_C</i> (0)
<i>FLAG</i> (7)	<i>REG_D</i> (15)
<i>FLAG</i> (8)	<i>REG_D</i> (0)
<i>FLAG</i> (9)	<i>REG_E</i> (15)
<i>FLAG</i> (10)	<i>REG_E</i> (0)
<i>FLAG</i> (11)	<i>REG_F</i> (15)
<i>FLAG</i> (12)	<i>REG_F</i> (0)
<i>FLAG</i> (13)	CNT=0
<i>FLAG</i> (14)	ALU pernaša
<i>FLAG</i> (15)	1 = 0
<i>FLAG</i> (16)	1 = 0
<i>FLAG</i> (17)	1 = 0
<i>FLAG</i> (18)	1 = 0

10 lentelė. Požymių registro įėjimai ir išėjimai

<i>Įvestys</i>	<i>Paskirtis</i>
<i>CLK</i>	Sinchronizavimo signalas
<i>RST</i>	Reset signalas
<i>Xin</i>	18 bitų įvestis
<i>Išvestys</i>	<i>Paskirtis</i>
<i>FLG_Dout</i>	18 bitų išvestis

8. Valdymo signalai

Išorinis **RST** signalas nustato pagrindinius komponentus į pradinę būseną, valdymo signalų formavimo schemai perduodant RST signalus šiems komponentams. Mikroprogramuotojas reikiamu metu gali atitinkamus komponentus nustatyti į pradinę būseną, naudodamas *RST_COMP* signalus (žr. 11 lentelę).

11 lentelė. Individualūs komponentų Reset signalai

<i>RST_COMP bitas</i>	<i>Valdomas komponentas</i>
<i>RST_COMP (0)</i>	<i>REG_A</i>
<i>RST_COMP (1)</i>	<i>REG_B</i>
<i>RST_COMP (2)</i>	<i>REG_C</i>
<i>RST_COMP (3)</i>	<i>REG_D</i>
<i>RST_COMP (4)</i>	<i>REG_E</i>
<i>RST_COMP (5)</i>	<i>REG_F</i>
<i>RST_COMP (6)</i>	CNT
<i>RST_COMP (7)</i>	ROM
<i>RST_COMP (8)</i>	FLG

Sinchronizacijos signalas yra naudojamas sistemos apibrėžtumo laipsniui padidinti. Komponentai, turintys savyje atminties elementus, atlieka savo operacijas lygiagrečiai, reaguodami į kylantį sinchronizacijos signalo frontą. **ALU** ir **MUX** nenaudoja sinchronizacijos signalo, nes yra kombinacinės logikos tipo ir keičia savo išėjimus, kai sulaukia kontrolės ar duomenų signalų pasikeitimo.

Dėl sinchronizacijos signalo ir natūralaus elementų vėlinimo nėra įmanoma vieno takto metu modifikuoti vieno komponento išėjimo ir panaudoti šiuos naujus duomenis kitame komponente. Šis apribojimas verčia duomenis paruošti naudojimui bent vienu sinchronizacijos signalo taktu anksčiau, nei kad juos reikės apdoroti kitam komponentui.

Reikia atkreipti dėmesį, jog MK naudojamas unitarinis kodavimo principas (kiekvieną mikrooperaciją atitinka atskiras bitas), kuris leidžia viename takte valdymo signalus lygiagrečiai perduoti nors ir visiems komponentams. Tačiau komponentams siunčiami valdymo signalai yra užkoduoti dvejetainiu kodu, taip sutaupant siunčiamos informacijos kiekį, tačiau apribojant galimybę komponentams atlikti daugiau nei vieną operaciją vieno takto metu.

Požymių signalai yra signalų grupė, kuriuos automatiškai formuoja komponentai (registrai, ALU, skaitiklis). Šie signalai nustato požymių registro bitų reikšmes. Kiekvienas signalas yra priskirtas individualiam požymių registro bitui (žr. 9 lentelę).

9. Mikrokomandų formatai

Mikrokomandų formatai skirtingi priverstinės ir natūralios adresacijos atvejais.

Mikrokomandų formatas priverstinės adresacijos atveju parodytas 2 paveiksle, o natūralios adresacijos atveju – 3 paveiksle.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
MUX							REG A							REG B							REG C							REG D							REG E						
Din	REG A	REG B	REG C	REG D	REG E	REG F	<=ALU.M	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1

43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
REG F							ALU							Reset									LS				N_ADDR										
<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	M = L+R	M= not L	M=not R	M=L + 1	M=L - 1	M=R + 1	M=R - 1	M= L xor R	REG_A	REG_B	REG_C	REG_D	REG_E	REG_F	CNT	ROM	FLAG	Data OUTPUT	CNT--	LS(3)	LS(2)	LS(1)	LS(0)	N_ADDR(7)	N_ADDR(6)	N_ADDR(5)	N_ADDR(4)	N_ADDR(3)	N_ADDR(2)	N_ADDR(1)	N_ADDR(0)

2 pav. Mikrokomandų formatas priverstinės adresacijos atveju

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
P	MUX							REG A							REG B					REG C					REG D										
MK tipas	<=ALU.M	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Mux	LL1	LR1	AL1	AR1	CL1	CR1

37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
REG E							REG F							ALU							Reset											
<= Mux	LL1	LR1	AL1	AR1	CL1	CR1	<= Din	LL1	LR1	AL1	AR1	CL1	CR1	M = L+R	M = not L	M =not R	M=L + 1	M=L - 1	M=R + 1	M=R - 1	M = Lxor R	REG_A	REG_B	REG_C	REG_D	REG_E	REG_F	CNT	ROM	FLAG	Data OUTPUT	CNT --

a) Operacinių mikrokomandų formatas (bitas **P** = 0)

1	2	3	4	5	6	7	8	9	10	11	12	13
P	LS				N_ADDR							
MK tipas	LS(3)	LS(2)	LS(1)	LS(0)	N_ADDR(7)	N_ADDR(6)	N_ADDR(5)	N_ADDR(4)	N_ADDR(3)	N_ADDR(2)	N_ADDR(1)	N_ADDR(0)

b) Perėjimo mikrokomandų formatas (bitas **P** = 1)

3 pav. Mikrokomandų formatai natūralios adresacijos atveju