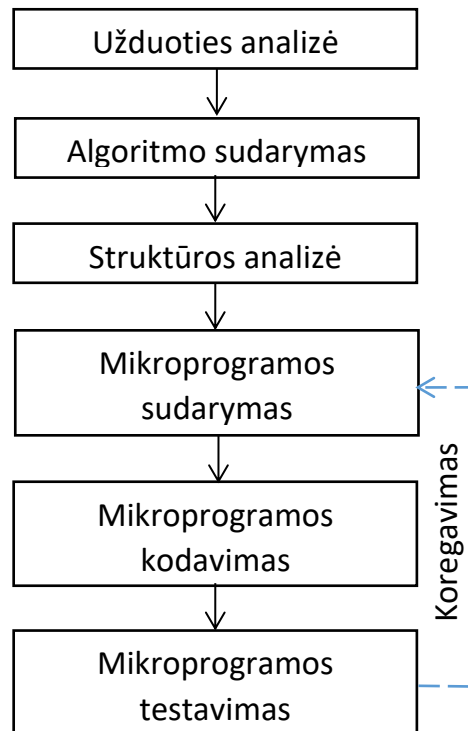


## Mikroprogramos sudarymo pavyzdys

Rekomenduojama tokia užduoties sprendimo tvarka:



1 pav. Užduoties sprendimo tvarka

Plačiau paaiškinsime kiekvieną žingsnį, iliustruodami sprendžiamu pavyzdžiu.

### Užduoties analizė

Spręsimė tokią užduotį: paskaičiuoti išraiškos

$$F = (a^2 - c)/(a + b)$$

rezultatą, kai  $a < 0$ ,  $b > 0$ ,  $c > 0$ , be to,  $b < |a|$ ,  $a^2 > c$ , neigiami skaičiai pateikiami papildomuoju kodu (rezultatas turi būti pateiktas taip pat papildomuoju kodu).

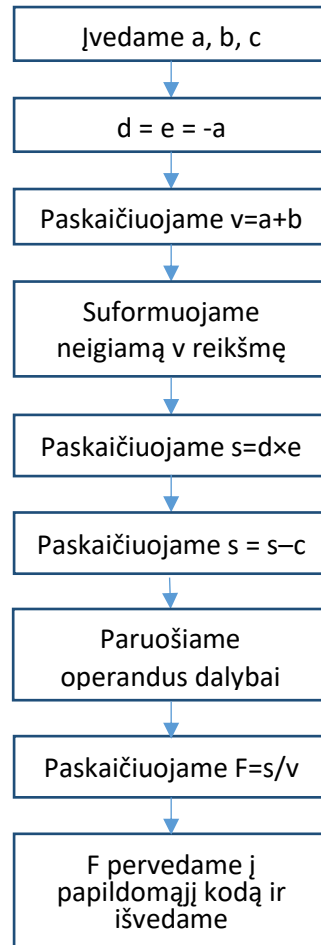
Kadangi  $b < |a|$ , suma  $(a + b)$  bus neigiama, skirtumas  $(a^2 - c)$  – teigiamas, rezultatas – neigiamas, tad jį reikės pervesti į papildomąjį kodą.

### Algoritmo sudarymas

Užduoties sprendimo (grubus) algoritmas pateiktas 2 paveiksle. Paaiškinsime jo žingsnius:

1. Įvedame pradinis duomenis, pateiktus papildomuoju kodu.

2. Kadangi  $a^2 = a \times a$  ir naudosime skaičių modulių daugybą, daugybai paruošiamo operandus, į **d** ir **e** įrašydami  $-a$ . Tam teks atlikti veiksmus  $d = e = \text{not}(a)+1$ .
3. Pirmiausia paskaičiuosime vardiklio  $(a + b)$  reikšmę. Vykdydami dalybą, vardiklis atimamas arba pridedamas, todėl reikia turėti ir neigiamą vardiklio reikšmę.
4. Paskaičiuosime sandaugos  $d \times e$  reikšmę.
5. Paskaičiuosime skaitiklio  $(a^2 - c)$  reikšmę; tam pirmiausia suformuodami  $-c$  papildomajame kode.
6. Paruošiamo operandus dalybai, vardiklį ir neigiamą jo reikšmę perstumdami į žodžio pradžią, nes to reikalauja dalybos algoritmas.
7. Atliksime dalybą, naudodami skaitiklio ir vardiklio modulius.
8. Suformuojame galutinį rezultatą, pervesdami jį į papildomąjį kodą, ir išvedame.



2 pav. Užduoties sprendimo (grubus) algoritmas

Toliau pateiksime modulių daugybos ir dalybos algoritmus.

### ***Daugybos algoritmas***

Paprastumo dėlei dauginsime skaičių modulius, tad korekcijų nereikės. Daugybos nuo žemiausių skilčių stumiant dauginamąjį algoritmas pateiktas 3 paveiksle. Paaiškinsime jo žingsnius:

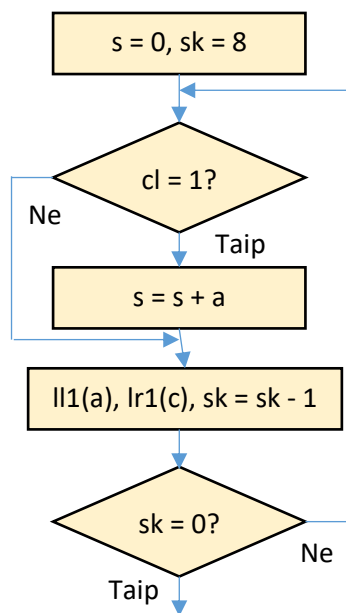
1. Nustatome pradinę sandaugų sumos **s** ir skaitiklio **sk** reikšmes.
2. Tikriname žemiausiąją **c** skiltį (**cl**).
3. Jei ji lygi 1, prie **s** pridedame **a**.
4. Daugiklį **c** pastumiame į dešinę, dauginamąjį **a** – į kairę. Sumažiname skaitiklio **sk** turinį.
5. Kol skaitiklis **sk** nelygus 0, grįžtame į 2 žingsnį.

### ***Dalybos algoritmas***

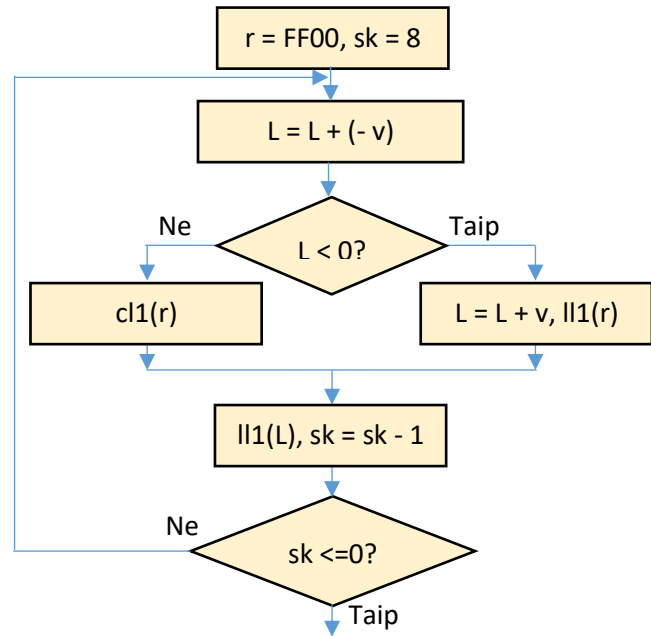
Paprastumo dėlei dalinsime skaičių modulius. Dalybos atstatant liekaną algoritmas pateiktas 4 paveiksle. Paaiškinsime jo žingsnius:

1. Nustatome pradinę rezultato **r** ir skaitiklio **sk** reikšmes. Pradinė liekana **L** bus dalijamasis – skaitiklis **s**.
2. Iš liekanos **L** atimame daliklį (vardiklį) **v**.
3. Tikriname liekanos **L** ženklą.
4. Jei liekana teigiama, rezultatą **r** cikliškaite pastumiame į kairę.
5. Jei liekana neigiama, rezultatą **r** logiškaite pastumiame į kairę ir atstatome liekaną, prie **L** pridėdami **v**.
6. Liekaną **L** pastumiame į kairę. Sumažiname skaitiklio **sk** turinį.
7. Kol skaitiklis **sk** nelygus 0, grįžtame į 2 žingsnį.

Paaiškinsime rezultato formavimo principą, kurio pasirinkimą lėmė modeliuojamos struktūros ypatumai. Dalybos rezultatas (dalmuo) formuojamas nuosekliai bitas po bito, naudojant postūmius į kairę. Jei pradinė dalmens reikšmė lygi 1111111100000000, ciklinis postūmis į kairę į žemiausiąją dalmens skiltį perkelia 1, o loginis postūmis – 0. Tokiu principu per 8 ciklus suformuojamos visos dalmens skiltys.



3 pav. Modulių daugybos algoritmas



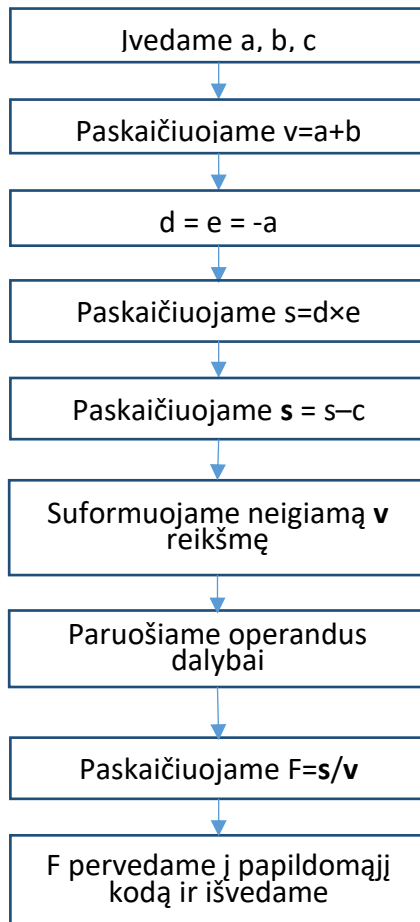
4 pav. Modulių dalybos algoritmas

### Struktūros analizė

Duomenims ir tarpiniams bei galutiniam rezultatams saugoti turime penkis registrus – B, C, D, E, F (registras A naudojamas ALU rezultatui fiksuoti, tad jį rezervuojame tik šiam tikslui). Kadangi daugybai reikia pakeisti **a** ženklą, o vardiklio **v** reikšmė reikalinga tiek teigiama, tiek ir neigiama, sudarytąjį algoritmą modifikuosime, pakeisdami veiksmų tvarką taip, kaip parodyta 5 paveiksle.

Tuomet duomenims ir tarpiniams rezultatams saugoti naudosime tokius registrus:

- **a**, **b** ir **c** įvesime į registrus B, C ir D;
- paskaičiavę **a+b**, vardiklio reikšmę **v** įrašysime į registrą E;
- **|a|** įrašysime į registrus B ir F;
- pakeitę **c** ženklą (nes  $a^2 - c = a^2 + (-c)$ ), jo reikšmę paliksime registre D;
- paskaičiavę  $a^2 = a \times a$ , sandaugos reikšmę paliksime registre A;
- prie jos prisumavę  $(-c)$ , gautą skaitiklio reikšmę įrašysime į registrą B;
- $-v$  įrašysime į registrą C;
- dalybos metu liekaną laikysime registre A;
- dalmenį **r** kaupsime registre F.



2 pav. Modifikuotas sprendimo algoritmas

## Mikroprogramos sudarymas

VHDL aprašytos struktūros ypatumai reikalauja kai kuriuos atrodytų paprastus žingsnius realizuoti ne viena, o keliomis mikrokomandomis. Pavyzdžiui, jei reikia susumuoti dviejų registų B ir C turinius, įrašant sumą į registrą D, reikės tokių mikrokomandų:

	Prasmė	Mikrooperacijos
MK1	$A = 0$	Reset/RegA
MK2	$A = A + B$	Mux/RegB, ALU/L+R, RegA/ALU_M
MK3	$A = A + C$	Mux/RegC, ALU/L+R, RegA/ALU_M
MK4	$D = A$	Mux/RegA, RegD/Mux

Čia kiekviena mikrooperacija sąlyginai užrašyta nurodant komponentą ir to komponento mikrooperaciją, atskiriant pasvirusiu brūkšniu (toks pseudokodas gan patogus mikroprogramos analizei).

Mikroprogramai sutrumpinti galima panaudoti šio fragmento pirmosios ir paskutinės mikrokomandų suliejimą su kitomis. Pavyzdžiui, jei prieš MK1 esančioje MK registras A nereikalingas, A nustatymą į 0 galima įdėti į ją.

Kartu turi būti reikiamu būdu mikrokomandoms priskirti adresai.

Panašiu būdu, naudodami pseudokodą, užrašome visą mikroprogramą:

MK adr.	Mikrooperacijos	Komentarai
<b><i>Duomenų įvedimas</i></b>		
0	Mux/Din, RegB/Mux	$B = a$
1	Mux/Din, RegC/Mux	$C = b$
2	Mux/Din, RegD/Mux, Reset/RegA	$D = c, A = 0$
<b><i>Vardiklio skaičiavimas</i></b>		
3	Mux/RegB, ALU/L+R, RegA/ALU.M	$A = a$
4	Mux/RegC, ALU/L+R, RegA/ALU.M	$A = v = a + b$
5	Mux/RegA, RegE/Mux	$E = v$
<b><i> a  skaičiavimas</i></b>		
6	Mux/RegB, ALU/not.L RegA/ALU.M	$A = a$ inversijai
7	ALU/R+1, RegA/ALU.M	$A = -a (  a  )$
8	Mux/RegA, RegB/Mux, RegF/Mux	$B = F =  a $
<b><i>-c skaičiavimas</i></b>		
9	Mux/RegD, ALU/not.L, RegA/ALU.M	$A = c$ inversijai
10	ALU/R+1, RegA/ALU.M	$A = -c$
11	Mux/RegA, RegD/Mux	$D = -c$
<b><i>Daugyba ir <math>a^2 - c</math> skaičiavimas</i></b>		
12	Reset/RegA	$A = 0$
13	FL≠0? → 15	Tikriname daugiklį
14	Mux/RegD, ALU/L+R, RegA/ALU.M; → 19	$A = A + D$ ; skaitiklis <b>s</b>
15	CNT--; → 17	
16	Mux/RegB, ALU/ L+R, RegA/ALU.M; → 15	$A = A + B$
17	RegB/LL1, RegF/LR1	Postūmiai
18	CNT≠0? → 13	Ar daugybos ciklas baigtas?

<b>Skaitiklio fiksavimas</b>		
19	Mux/RegA, RegB/Mux,	$B = s$
<b>Pasirengimas dalybai</b>		
20	Reset/RegA	$A = 0$
21	ALU/not.R, RegA/ALU.M	$A = \text{FFFFh}$
22	Mux/RegA, RegF/Mux	$r = F = \text{FFFFh}$
23	Mux/RegE, ALU/not.L, RegA/ALU.M	$A = v$ inversijai
24	ALU/R+1, RegA/ALU.M	$A = -v$
25	Mux/RegA, RegC/Mux	$C = -v$
26	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
27	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
28	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
29	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
30	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
31	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
32	RegC/LL1, RegE/LL1, RegF/LL1	Stumiamo $\leftarrow v, -v, r$
33	RegF/LL1, Reset/RegA, Reset/CNT	Stumiamo $\leftarrow r, A=0, \text{CNT}=8$
34	Mux/RegB, ALU/L+R, RegA/ALU.M	$A = s$
<b>Dalyba</b>		
35	Mux/RegE, ALU/L+R, RegA/ALU.M; $\rightarrow 37$	$s = s - v$
36	$\rightarrow 42$	I pabaigą
37	CNT--	
38	AH=0? $\rightarrow 39$	Tikriname liekanos ženklą
39	RegF/CL1, $\rightarrow 41$	$r \leftarrow 1$
40	RegF/LL1, Mux/RegC, ALU/L+R, RegA/ALU.M	$s = s + v, r \leftarrow 0$
41	RegA/LL1, CNT $\neq 0$ ? $\rightarrow 35$	Ar dalybos ciklas baigtas?
<b>Rezultato ženklo keitimas ir išvedimas</b>		
42	Mux/RegF, ALU/not.L, RegA/ALU.M	$A = r$ inversijai
43	ALU/R+1, RegA/ALU.M	$A = r$ pap.kode
42	Mux/RegA, DataOutput	Out RegA
43	Mux/RegA, DataOutput	Out RegA

## Mikroprogramos kodavimas

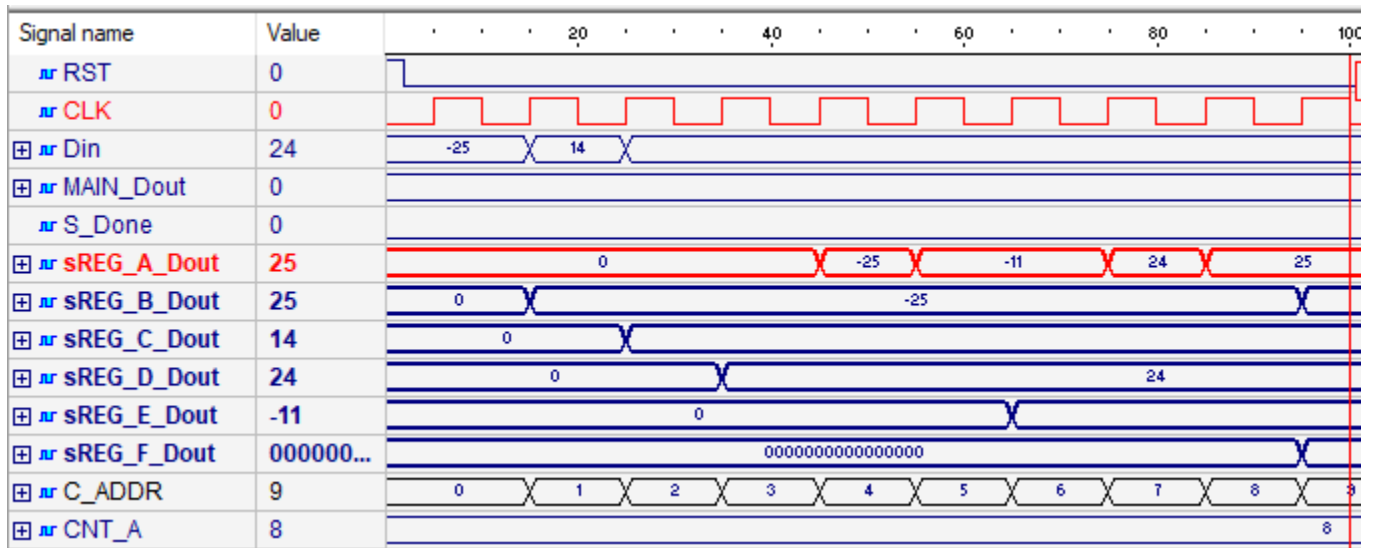
Koduojant tereikia atidžiai įrašyti „1“ mikrokomandos vietose, atitinkančiose vykdytinąs mikrooperacijas.

## Mikroprogramos testavimas

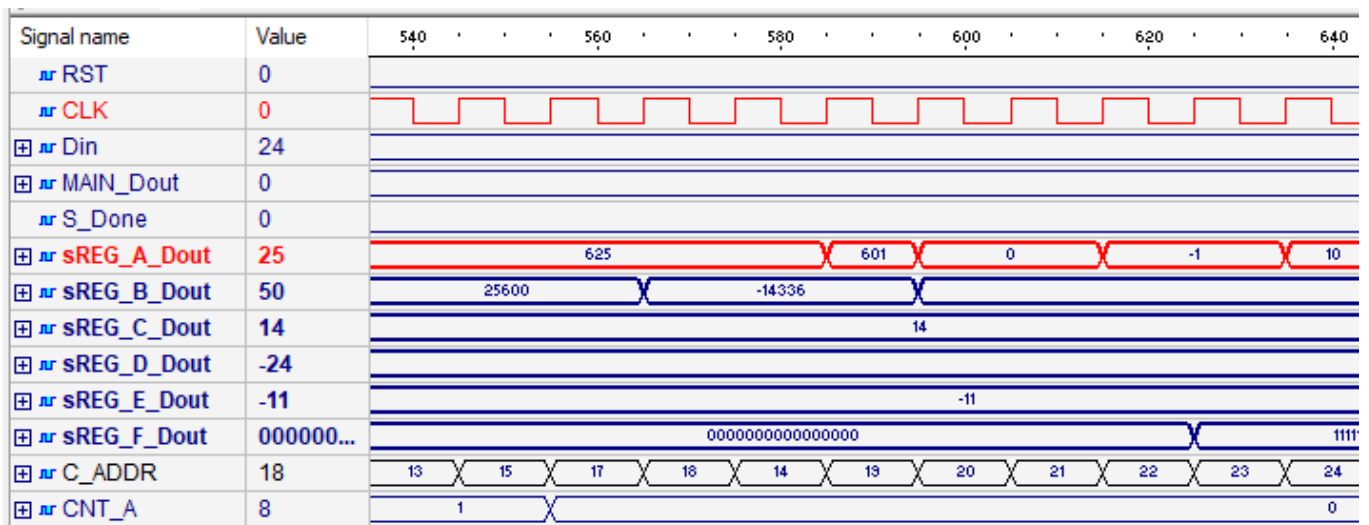
Modeliuosime išraiškos skaičiavimo mikroprogramą, pateikdami tokius testinius duomenis:  $a = -25$ ,  $b = 14$ ,  $c = 24$ . Turime gauti  $a+b = -11$ ,  $c = 49$ ,  $a^2-c = 601$ , o  $601/(-11) = 54$  (liekana 7),  $r = -54$ .

Rezultatams stebėti į ekraną papildomai išvesime visų registrų turinius, vykdytos mikrokomandos adresą ir skaitiklio turinį. Rezultatus matysime papildomajame kode (išskyrus RegF – dvejetainiu pavidalu).

Duomenų įvedimas ir  $a+b = -11$  (vardiklio) paskaičiavimas:

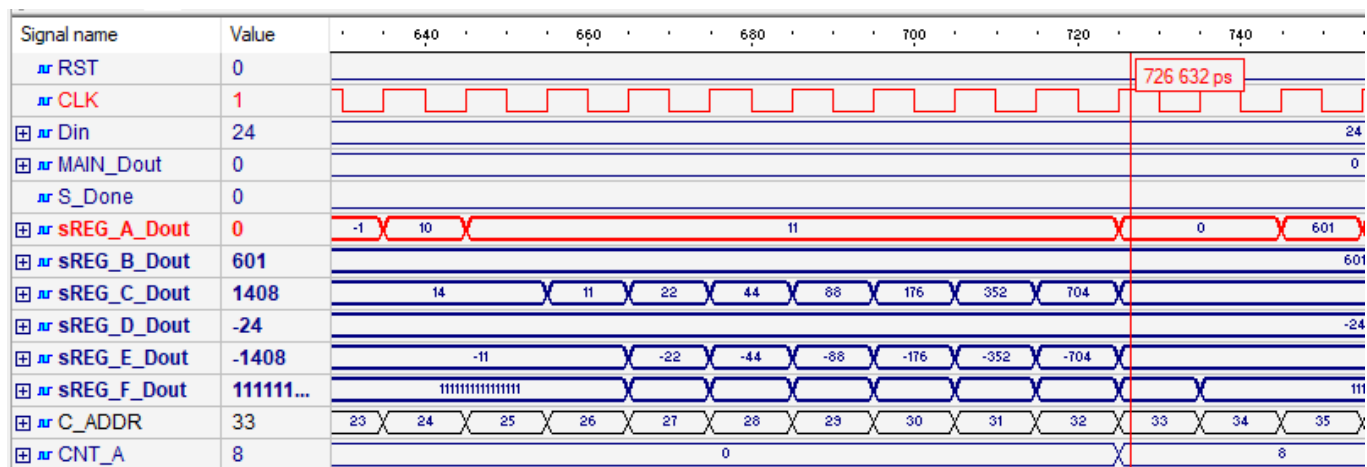


Daugybės ciklo pabaiga ir  $a^2-c = 625-24 = 601$  (skaitiklio) paskaičiavimas:

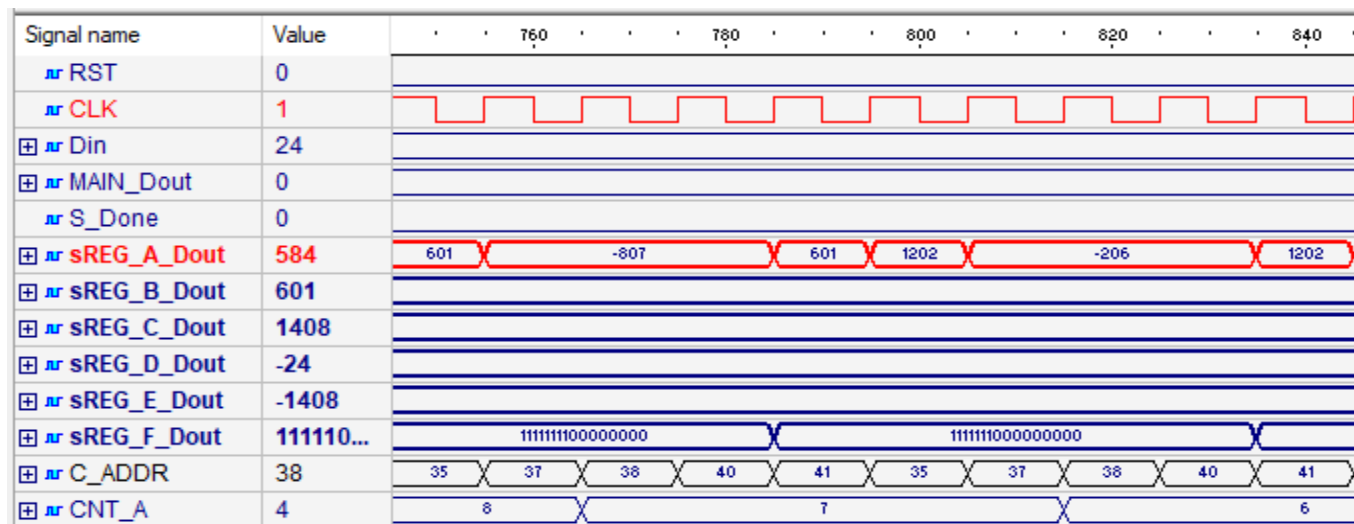


Pasirengimas dalybai:





Pirmasis dalybos ciklas:



Dalybos pabaiga ir rezultato (–54 papildomajame kode) formavimas:

