**Mikroprogramos sudarymo pavyzdys**

Rekomenduojama tokia užduoties sprendimo tvarka:

Koregavimas

Užduoties analizė

Algoritmo sudarymas

Struktūros analizė

Mikroprogramos kodavimas analizė

Mikroprogramos testavimas analizė

Mikroprogramos sudarymas analizė

1 pav. Užduoties sprendimo tvarka

Plačiau paaiškinsime kiekvieną žingsnį, iliustruodami sprendžiamu pavyzdžiu.

**Užduoties analizė**

Spręsime tokią užduotį: paskaičiuoti išraiškos

**F = (a×b)/(a+b)**

rezultatą, kai **b**<0, **a**>|**b**|, neigiami skaičiai pateikiami papildomuoju kodu (rezultatas turi būti pateiktas taip pat papildomuoju kodu).

Kadangi **a**>|**b**|, suma bus teigiama, sandauga **a×b** – neigiama, rezultatas - taip pat neigiamas.

**Algoritmo sudarymas**

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

1. Įvedame pradinius duomenis papildomuoju kodu.
2. Kadangi dauginsime skaičių modulius, daugybai paruošiame operandus, į **c** įrašydami –**b**. Tam teks atlikti veiksmus **c**=not(**b**)+1.
3. Pirmiausia paskaičiuosime vardiklio **a+b** reikšmę. Ją perstumsime į žodžio pradžią, nes to reikalauja dalybos algoritmas.
4. Vykdant dalybą, vardiklis atimamas arba pridedamas, todėl reikia turėti ir neigiamą vardiklio reikšmę.
5. Paskaičiuosime skaitiklio **a**×**c** reikšmę.
6. Atliksime dalybą, naudodami skaitiklio ir vardiklio modulius .
7. Suformuosime galutinį rezultatą. Kadangi sandauga yra neigiama, o suma bus teigiama, rezultatas bus neigiamas skaičius, todėl reikės jį išvesti papildomuoju kodu.

Įvedame a, b

c = -b

Paruošiame r reikšmę, paskaičiuojame v=a+b ir perstumiame į kairę

Suformuojame neigiamą v reikšmę

Paskaičiuojame s=a×c

Paskaičiuojame F=s/v

Pakeičiame F ženklą, 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 žemiausiųjų skilčių stumiant dauginamąjį algoritmas pateiktas 3 paveiksle. Paaiškinsime jo žingsnius:

1. Nustatome pradines 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. Prieš dalybą suformuosime pradinę dalmens **r**  reikšmę – konstantą 1111111100000000, o daliklį pastumsime į kairę per 8 skiltis, kad jis atsidurtų žodžio pradžioje.

Dalybos atstatant liekaną algoritmas pateiktas 4 paveiksle. Paaiškinsime jo žingsnius:

1. Nustatome pradinę skaitiklio **sk** reikšmę (8). Pradinė liekana **L** bus dalijamasis – skaitiklis **s**.
2. Iš liekanos **L** atimame daliklį (vardiklį) **v** ir sumažiname skaitiklio **sk** turinį.
3. Tikriname liekanos **L** ženklą.
4. Jei liekana neigiama, ją atstatome, prie **L** pridėdami **v**, o **r** pastumiame į kairę logiškai (taip suformuojamas „eilinis“ jo bitas, lygus 0).
5. Jei liekana teigiama, **r** pastumiame į kairę cikliškai (taip suformuojamas „eilinis“ jo bitas, lygus 1).
6. Liekaną **L** pastumiame į kairę.
7. Kol skaitiklis **sk** nelygus 0, grįžtame į 2 žingsnį.

s = 0, sk = 8

cl = 1?

s = s + a

LL1(a), LR1(c), sk = sk - 1

sk = 0?

Taip

Taip

Ne

Ne

L = s, sk = 8

L = L + (- v), sk = sk - 1

L < 0?

Ne

Taip

sk <=0? 0?

Taip

Ne

LL1(r), L = L + v

CL1(r)

LL1(L)

3 pav. Modulių daugybos algoritmas 4 pav. Modulių dalybos algoritmas

**Struktūros analizė**

Kintamiesiems saugoti naudosime tokius registrus:

* **a** įvesime į registrą B;
* **b** įvesime į registrą E;
* |**b**| laikysime registre C;
* paskaičiavę **a**+**b**, įrašysime į registrą E;
* **-**(**a**+**b**) įrašysime į registrą D;
* daugybos metu dalinių sandaugų sumą kaupsime registre A;
* dalybos metu liekaną laikysime registre A;
* dalmenį **r** kaupsime registre F.

**Mikroprogramos sudarymas**

VHDL aprašytos struktūros ypatumai reikalauja kai kuriuos atrodytų paprastus žingsnius realizuoti ne viena, o keliomis mikrokomandomis. Pavyzdžiui, jei reikia susumuoti dviejų registrų 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 tašku.

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

Faile SM\_mikroprograma.xlsx rasite mikroprogramą su dešinėje užrašytais komentarais. Pateiksime algoritmo žingsnių atitikimą mikrokomandoms:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 paveikslo algoritmas | |  | Daugybos algoritmas | |  | Dalybos algoritmas | |
| Žingsnis | Mikrokomandos |  | Žingsnis | Mikrokomandos |  | Žingsnis | Mikrokomandos |
| 1 | 0,1 |  | 1 | 23 |  | 1 | 29 |
| 2 | 2-4 |  | 2 | 24 |  | 2 | 30,31 |
| 3 | 5-18 |  | 3 | 27 |  | 3 | 33 |
| 4 | 19-22 |  | 4 | 26 |  | 4 | 34 |
| 5 | 23-28 |  | 5 | 28 |  | 5 | 35 |
| 6 | 29-36 |  |  |  |  | 6 | 36 |
| 7 | 41-45 |  |  |  |  | 7 | 36 |

Kartu turi būti reikiamu būdu mikrokomandoms priskirti adresai.

**Mikroprogramos kodavimas**

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

Mikroprogramą rasite faile ***SM\_mikroprograma.xlsx***.

**Mikroprogramos testavimas**

Modeliuosime išraiškos skaičiavimo mikroprogramą, pateikdami tokius testinius duomenis: **a** = 75, **b** = -49. Turime gauti **a**+**b** = 26, **a**×**b** = 3675, o 3675/26 = 141 (liekana 9), **F =** –141.

+75=4B, -49=FFCF, 3675 = E5B , 141 = 8D, -141 = FF73.

Modeliuojant gauta (reikšmės – šešioliktainės):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Dydis*** | ***Registras*** | ***Mikrokomanda*** | ***Reikšmė*** | ***Laikas (ns)*** |
| **a** | **B, F** | **0** | **4B** | **15** |
| **b** | **E** | **1** | **FFCF** | **25** |
| |**b**| | **C** | **4** | **31** | **55** |
| **a**+**b** | **A** | **7** | **1A** | **115** |
| **a**+**b** pastumtas | **E** | **16** | **1A00** | **195** |
| **a**+**b** pastumtas su prieš. ženklu | **D** | **19** | **E600** | **235** |
| **a**×**b** | **C** | **26** | **0E5B** | **545** |
| **a**×**b**/**(a**+**b)** | **F** | **28** | **008D** | **965** |
| **F** | **F, DOut** | **43** | **FF73** | **1015** |