Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Архітектура комп’ютерів-1

Лабораторна робота №4

**«Обробка інформації в ЕОМ на програмному, мікропрограмному та апаратному рівнях»**

Виконала:

студентка групи ІО-64

Бровченко А. В.

Перевірив:

доц. Верба О. А.

Київ

2018 р.

***Мета роботи*** – вивчити етапи формування системи команд процесорів, різновиди форматів команд та способів адресації операндів. Навчитися розробляти мікроалгоритми і мікропрограми виконання етапів команд з використанням мнемонічного мікроасемблера .

***Підготовка до лабораторної роботи***

1. Вивчити загальну інформацію до виконання лабораторних робіт 3 і 4 на моделюючому комплексі ЕОМ з мікропрограмним управлінням (розділ 4).

2. Визначити свій варіант завдання у відповідності до табл. 6.1 – 6.4, в яких а7, а6, а5, а4, а3, а2, а1 – сім молодших розрядів двійкового номера залікової книжки студента (а1 – молодший розряд).

3. Подати архітектуру системи. Архітектура подається за допомогою моделі програміста, в якій вказуються регістри загального призначення, робочі регістри, представляється адресний простір ОП, НОЗП і ЗП.

4. Розробити на рівні операцій змістовний алгоритм обчислення функції

*,*

де «» – операція, що задана табл. 6.1; «» – операція множення, що визначена табл. 6.4.

Операнди *X, Y,* і *Z* у доповняльному коді мають 16 розрядів з урахуванням двох знакових розрядів. Операнд *X* вводиться з пристрою введення. Адреси портів зовнішнього пристрою (ЗП) задані табл. 6.2. Готовність ЗП до обміну даними з процесором перевіряти на програмному рівні з використанням команди умовного переходу. Операнди *Y,* і *Z* знаходяться в ОП. Результат *F* записуються в ОП словами по 16 розрядів (молодші розряди – за меньшими адресами). Послідовність введення даних в процесор визначається розробником. Вихідні дані задані в табл. 6.1-6.4. Відсутні дані, якщо необхідно, вибираються самостійно (їх вибір повинний бути обґрунтований).

На схемі алгоритму кожний блок (операторний або логічний) повинний відповідати одній команді процесора (додавання, множення, пересилання, введення, запис, умовний перехід і т.і.).

5. Сформувати систему команд процесора і розробити програму обчислень на базі команд одержаної системи. Необхідно обгрунтовати вибір системи команд, необхідних для обчислення функції. Програма обчислень представляється в кодах команд, а також у мнемонічному виді. Мнемоніка команд вибирається самостійно.

Формат одноадресних команд, а також регістра стану і даних ЗП показано на рис. 6.4 і 6.5. В командах з прямою адресацією операндів адреса ОП міститься в розрядах команди 9…0, тобто займає 10 розрядів. Це забезпечує доступ до ОП з адресами від 0 до 1023. Програма і дані повинні бути розміщені в цьому діапазоні адрес ОП. При використанні для прямої регістрової адресації адреса (номер) РЗП займає тільки молодші розряди адресного поля команди. Якщо для РЗП виділено в НОЗП 8 регістрів R0-R7. то для їх адресації достатньо 3 розряків. В адресному полі залишаеться 7 розрядів, тобто при регістровій адресації можно побудувати двоадресну команду. Розробник може використувати таку можливість.

* 1. Представити структурну схему ЕОМ, з зазначенням необхідних вузлів і керуючих сигналів, що використовуються при розробці мікропрограм. Мікропрограма описується на мікроассемблері системи COMPLEX і повинна пояснюватися відповідними коментарями.

**Завдання**

Розробити архітектуру комп’ютера, систему команд, структурну схему та мікропрограму реалізації етапів виконання команд.

Система команд повинна забезпечувати обчислювання заданого алгебраїчного виразу на програмному рівні. Операнди є цілими 16-розрядними числами (один знаковий та 15 основних розрядів). Операнди вводяться у режимі програмного опиту готовності зовнішніх пристроїв. Результат виводиться у пристрій виводу. Послідовність даних, що вводяться та виводяться, визначається розробником. Система команд містить лише одноадресні команди (рис.1).



Рисунок 1.1 – Структура команди.

Кількість типів команд обирається розробником, виходячи з необхідності реалізації заданих обчислень з урахуванням вводу та виводу даних, а також перевірки готовності зовнішніх пристроїв. Введення кожного типу команд у систему команд повинно бути обґрунтованим. Зокрема, система команд повинна містити команди виконання заданих операцій, вводу та виводу даних, команди умовних та безумовних переходів, пересилання даних.

Обробка інформації на мікро програмному рівні повинна містити вибірку команди, розпакування команди, виконання операції та формування адреси наступної команди. Однакові блоки мікрокоманд бажано оформляти у вигляді мікропідпрограм. При програмному опитуванні дані регістру стану зовнішнього пристрою мають формат, зображений на рис.2.



Рисунок 1.2 – Формат регістру стану зовнішнього пристрою

Початкові умови згідно з варіантом

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер залікової книжки: 11010011001 | | | | | | |
| a7 | a6 | a5 | a4 | a3 | a2 | a1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| Адреса РС | | | Адресація | Операція | | |
| Порт вводу | | |
| 12h | | | (X - Y) \* Z | | |
| Адреса РД | | | Спосіб множення | | |
| Пряма |
| 13h | | | 2ий | | |

**2 Архітектура та алгоритм обчислення**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Структура НОЗП | | | Структура ПМК | |
| R0 | Результат | РЗП (регістри загального призначення) | 0 | Вектори команд |
| … |  | … |
| 15H |
| R7 | ЛК | Робочі регістри | … |  |
| R8 | РК | 20H | Мікропрограма |
| . . . |  | … |
| 66H |
| R14 | адреса операнда |
| R15 | Операнд |
| RQ |  |

Рисунок 2.1 Модель програміста

|  |  |  |  |
| --- | --- | --- | --- |
| Структура ОП | | Структура ЗП | |
| 00000H |  | 0000H |  |
| 0001H | Результат | … |  |
| 0002H |
| 0004H | Код пристрія вводу ЗП (РС) | 0012H | РС (пристрій вводу) |
| 0006Н | Дані |
| 0008H | 0014H | РД (пристрій вводу) |
| 000АH | Програма обчислення | … |  |
| 000СН |
| … |
| 0014H |
| 0016H |
| … |  |
| FFFFFH |  | FFFFH |  |

Рисунок 2.2 Розподіл загальної пам’яті

**3 Програма обчислень**

Для реалізації програми обчислень потрібні такі команди

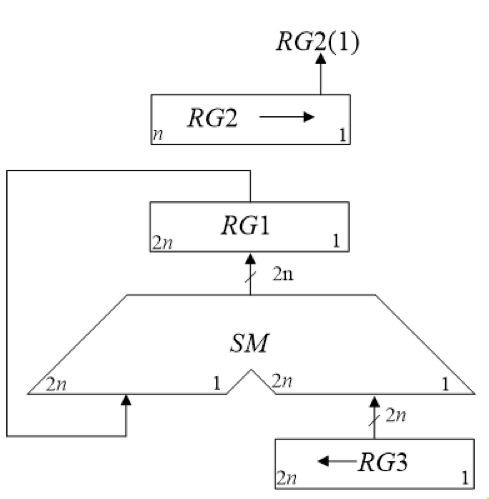
|  |  |  |
| --- | --- | --- |
| Команда | Код | Операція |
| testdevice | 0 0 0 0 | Тест ЗП |
| jz | 0 0 0 1 | Повертання , якщ ЗП не працює |
| input | 0 0 1 0 | Зчитування першого операнда з ЗП |
| sub | 0 0 1 1 | Віднімання другого операнду |
| mul | 0 1 0 0 | Множення операндів |
| res | 0 1 0 1 | Запис результатів |
| end | 0 1 1 1 | Вихід з програми |

Таблиця 3.1 Програма обчислень

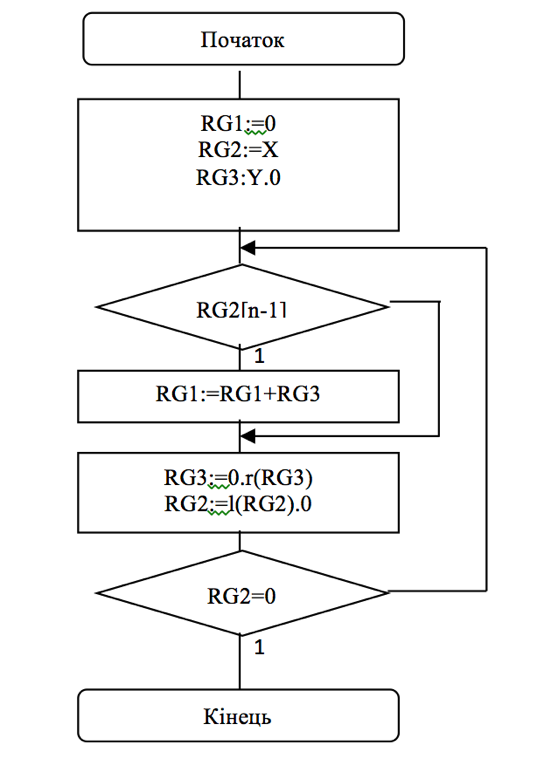
**4 Структура комп’ютера та мікропрограма обчислень**

Рисунок 4.1 Структурна схема ЕОМ

**Операційна схема другого способу:**



**Змістовний мікроалгоритм другого способу:**



Програмний код

\FORMULA F = (x - y) \* z

\nastroiyka shemy

link l1:ct

link l2:rdm

LINK l3:rdd

link ewh:10 \mladshiy razriad starshego adresa

link M: z,z,z,z,z,z,z,z,14,13,12,11

ACCEPT dev[1]:I,012h,014h,12,2

ACCEPT dev\_buf[1]:0006h

\dannye

dw 0001h: 0h\res1

dw 0002h: 0h\res2

dw 0004h: 0012h\i--x

dw 0006h: 0007h\y

dw 0008h: 0002h\z

DW 0020h:000Ah \ Return addr

\programma

dw 000Ah: 0000100000000100% \testdevise

dw 000Ch: 0001000000100000% \jz

dw 000Eh: 0001100000000100% \input

dw 0010h: 0010000000000110% \sub

dw 0012h: 0010100000001000% \mul

dw 0014h: 0011000000000001% \res

dw 0016h: 0011100000000000% \end

\mikroprogramma

ORG 0h {cjp nz,start;}

ORG 1h {cjp nz,testdevice;}

ORG 2h {cjp nz,zjump;}

ORG 3h {cjp nz,input;}

ORG 4h {cjp nz,subb;}

ORG 5h {cjp nz,mul;}

ORG 6h {cjp nz,res;}

ORG 7h {cjp nz,end;}

\ustanovka nachalnogo adresa

start {and r13, r13, z;}

{or r13, r13, 000Ah;}

\chtenie comandy

begin {ewh; oey; xor nil, r13, r13;}

{ewl; oey; or nil, r13, z;}

{cjp rdm, cp; r; or r14, bus\_d, z;}

\raspakovka comandy

\proverka formata

{and nil, r14, 8000h; load rm, flags;}

{cjp not rm\_z, end;}

\proverka tipa adresazyi

{and nil, r14, 0400h; load rm, flags;}

{cjp not rm\_z, end;}

\zagruzka adresa registra v RB

{oey;or nil,r14,z;load rb;load ra;}

\perehod po kodu operacyi

{oey;or nil, r14,z;jmap;}

\mikroprogramy komand

FromMem

{or nil,r14,z;ewl;oey;}

{r;or r12,bus\_d,z;cjp rdm,cp;}

{crtn nz;}

\ Command test

testdevice

{cjs nz,FromMem;}

{or nil,r12,z;oey;ewl;}

{i;cjp rdd,cp;or r12,bus\_d,z;}

{or nil,r12,z;load rn,flags;}

{cjp nz,formadd;}

\ Command jz [mem]

zjump

{cjp not rn\_z,no\_j;}

{cjs nz,FromMem;}

{or r13,r12,z;}

{cjp nz,begin;}

no\_j {cjp nz,formadd;}

\ Command in r2,[mem]

input

{cjs nz,FromMem;}

{add r12,r12,2,z;}

{or nil,r12,z;ewl;oey;}

{i;cjp rdd,cp;or r2,bus\_d,z;}

{cjp nz,formadd;}

\--------------mul------------------------—

mul {cjs nz,FromMem;}

{or r5, r12, z;}

{or R5, 8000h;} \ set marker bit in r1(X)

{and nil, R2, 8000h; load RM, FLAGS;}\ set all bit 1 in R1 if -Y

{cjp RM\_Z, MMUL;} \ or do nothing

{or R1, 0FFFFh;} \

MMUL {and nil, R5, 0001h; load RM, FLAGS;}\ Check if 0 bit X == 0

{cjp RM\_Z, link1;} \ and jump link1 if TRUE

{add R4, R4, R2, RM\_C; load RM, FLAGS;} \ SUM (R3-R4)=(R3-R4)+Y

{add R3, R3, R1, RM\_C;} \

link1 {add SRL, R5, Z;} \ shift registers

{add SLL, R2, Z;} \ RY< RX>

{add SL.25, R1, 0h;} \

{xor R6, R6;} \ Check if x = 0

{or R6, R5;} \ (without sing bits)

{and R6, 0FFFCh; load RM, FLAGS;} \

{cjp not RM\_Z, MMUL;} \ if not repeat sum

{and nil, R5, 0001; load RM, FLAGS;} \ if X.0 = 1

{cjp RM\_Z, formadd;} \ then add X

{xor R2, 0FFFFh;} \ first need convert it

{xor R1, 0FFFFh;} \ from -X to +X

{add R2, R2, 0001h; load RM, FLAGS;} \ then simple ADD

{add R1, R1, Z, RM\_C;}

{add R4, R4, R2, Z; load RM, FLAGS;}

{add R3, R3, R1, RM\_C;}

{cjp nz,formadd;}

\--------------sub-------------------------

subb

{cjs nz,FromMem;}

{xor R12, R12, 0FFFFh;} \ Y = -Y

{add r2,r2,r12,nz;} \ X - Y + 1

{cjp nz,formadd;}

\--------------res-------------------------

res

{cjs nz,FromMem;}

{xor nil, r14, Z; oey; ewl;}

{cjp rdm, CP; W; or nil, Z, r4; oey;}

{add r14,r14,1h,z;}

{xor nil, r14, Z; oey; ewl;}

{cjp rdm, CP; W; or nil, Z, r3; oey;}

{cjp nz,formadd;}

\---------------------------------------

\formir sled adresa i na vyborku komandy

formadd{add r13,r13,2,z;}

{cjp nz,begin;}

\-------------end---------------------—

\konec programmy

end {}

**Висновок**

В процесі розробки був розроблений інструментарій для забезпечення необхідних перетворень інформації, які необхідні для обчислень за даним алгебраїчним виразом з введенням та виведенням інформації на зовнішні пристрої. На мікропрограмному рівні була розроблена система команд та реалізований алгоритм організації процесу обчислень.