НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра спеціалізованих комп’ютерних систем

**РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА**

# з дисципліни“Архітектура комп’ютера”

***Тема: "Обробка інформації в ЕОМ на програмному,***

***мікропрограмному і апаратному рівнях".***

Виконав

студент гр. КВ-92

Степанюк М.Ф.

Київ 2012

1. **Завдання**

Розробити програму обчислення функції

,

де  - операція, що задана в табл.1. Результат цієї операції умножається на . Логічні операції є порозрядними.

Операнди  є 16-розрядними додатними цілими числами (старший знаковий розряд дорівнює нулю). Операнди вводяться з пристрою введення (ПВв). Послідовність введення визначається самостійно. Результат (в 32-розрядному форматі) виводиться в пристрій виведення (ПВив), а також записується в дві сусідні комірки пам’яті . Адреси регістрів стану (РС) і даних (РД) зовнішніх пристроїв подані в табл. 2, адреси пам’яті задати самостійно. Послідовність даних, що вводяться та виводяться, визначається розробником. Система команд містить лише одноадресні команди. Структура одно адресної команди подана на рис.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| Ф | Операція | ТА | Адреса в ОП |  |
| 15 | 14……...11 | 10 | 9……………..0 |  |

***Рис. 1.1***

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.1.2.



***Рис. 1.2 – Формат регістру стану зовнішнього пристрою***

Початкові умови згідно з варіантом

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| а7 | а6 | | а5 | а4 | а3 | а2 | а1 | |
| 0 | 0 | | 1 | 1 | 1 | 1 | 0 | |
| Адреси РС | | | | Тип адресації | Операція # | | | |
| ПВв | | ПВив | | Пряма | Спосіб множення | | | - |
| 12H | | 92H | |  | 4-й | | | - |

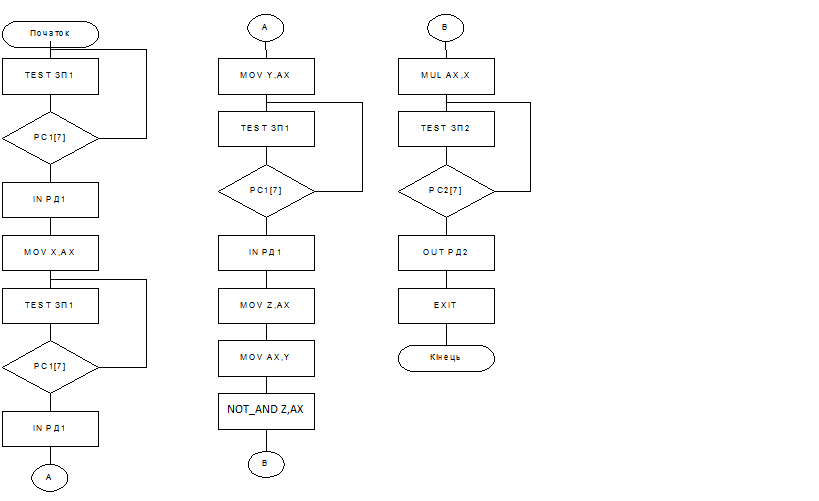
**2 Архітектура та алгоритм обчислення**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Структура НОЗП | | | Структура ПМК | |
| R0 |  | РЗП (регістри загального призначення) | 0 | Вектори команд |
| … | … |
| R6 | 15H |
| R7 | ЛК | Робочі регістри | … |  |
| R8 | РК | 20H | Мікропрограма |
| R9 | ВС | … |
| R10 |  | 66H |
| R11 |
| R12 | Операнд [0...7] |
| R13 | Операнд [8...15] |
| R14 | Адреса операнда |
| R15 | Аккумулятор |
| RQ |  |

***Рисунок 2.1 Модель програміста***

|  |  |  |  |
| --- | --- | --- | --- |
| Структура ОП | | Структура ЗП | |
| 00000H |  | 0000H |  |
| … | … |  |
| 00020H | Програма обчислення | 0012H | РС (пристрій вводу) |
| … | 0013H |
| 0004BH | 0014H | РД (пристрій вводу) |
| … |  | 0015H |
| 00050H | Дані | … |  |
| … | 0092H | РС (пристрій виводу) |
| 00069H | 0093H |
| … |  | 0094H | РД (пристрій виводу) |
| FFFFFH | 0095H |
|  | | … |  |
| FFFFH |

***Рис 2.2 Розподіл загальної пам’яті***



***Рис 2.3 Алгоритм обчислень***

**3 Програма обчислень**

|  |  |  |
| --- | --- | --- |
| Команда | Код | Операція |
| in ax,[mem] | 0 0 0 1 | Зчитування з ЗП |
| out [mem],ax | 0 0 1 0 | Запис у ЗП |
| mov [mem],ax | 0 0 1 1 | Пересилання з пам’яті до акумулятора |
| mov ax,[mem] | 0 1 0 0 | Пересилання з акмулятора до пам’яті |
| swap ax,reg | 0 1 0 1 | Обмін регістрів |
| exit | 0 1 1 0 | Вихід |
| not\_and [mem],ax | 0 1 1 1 |  |
| test ax,[mem] | 1 0 0 0 | Перевірка готовності ЗП |
| jz [mem] | 1 0 0 1 | Умовний перехід |
| mul [mem],ax | 1 0 1 0 | Множення |

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



***Рис. 4.1 Структурна схема ЕОМ***

**5. Мікропрограма обчислення функції.**

\Початкова настройка системи

LINK l1:ct

LINK l2:rdm

LINK l3:rdd

LINK ewh:16

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

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

ACCEPT dev[2]:O,092h,094h,12,2

ACCEPT dev\_buf[1]:0fff4h,24ffh,31ach

ACCEPT R7:0020h

ACCEPT R10:03ffh

\Програма

DW 0020h:0001100001010110% \test [0056h]

DW 0022h:0001000001011010% \jz [005Ah]

DW 0024h:0010100001010110% \in r15,[0056h]

DW 0026h:0011100001100100% \mov [0064h],r15

DW 0028h:0001100001010110% \test [0056h]

DW 002Ah:0001000001011100% \jz [005Ch]

DW 002Ch:0010100001010110% \in r15,[0056h]

DW 002Eh:0011100001100110% \mov [0066h],r15

DW 0030h:0001100001010110% \test [0056h]

DW 0032h:0001000001011110% \jz [005Eh]

DW 0034h:0010100001010110% \in r15,[0056h]

DW 0036h:0011100001110110% \mov [0068h],r15

DW 0038h:0100000001100110% \or r15,[0066h]

DW 003Ah:0100100001100100% \mul [0064h]

DW 003Ch:0001100001011000% \test [0058h]

DW 003Eh:0001000001100000% \jz [0060h]

DW 0040h:0010000001011000% \out [0058h]

DW 0042h:0000100000000000% \swap r15,r13

DW 0044h:0001100001011000% \test [0058h]

DW 0046h:0001000001100010% \jz [0062h]

DW 0048h:0010000001011000% \out [0058h]

DW 004Ah:0101000000000000% \exit

\Данi

DW 0056h:006Ah \(IN)

DW 0058h:006Ch \(OUT)

DW 005Ah:0020h \Адреси повернення

DW 005Ch:0028h

DW 005Eh:0030h

DW 0060h:003Eh

DW 0062h:0046h

DW 0064h:006Eh \(X)

DW 0066h:0070h \(Y)

DW 0068h:0072h \(Z)

DW 006Ah:0012h \IN

DW 006Ch:0092h \OUT

DW 006Eh:0000h \X

DW 0070h:0000h \Y

DW 0072h:0000h \Z

\Встановлення векторів мікрокоманд

ORG 0h \Вектор початку програми

{cjp nz,go;}

ORG 2h \Вектор команди swap r15,r13

{cjp nz,swap;}

ORG 4h \Вектор команди jz [mem]

{cjp nz,zjump;}

ORG 6h \Вектор команди test [mem]

{cjp nz,testdevice;}

ORG 8h \Вектор команди out [mem],r15

{cjp nz,output;}

ORG 0Ah \Вектор команди in r15,[mem]

{cjp nz,input;}

ORG 0Ch \Вектор команди mov r15,[mem]

{cjp nz,movToReg;}

ORG 0Eh \Вектор команди mov [mem],r15

{cjp nz,movToMem;}

ORG 10h \Вектор команди orr r15,[mem]

{cjp nz,orr;}

ORG 12h \Вектор команди mul

{cjp nz,mul;}

ORG 14h \Вектор команди exit

{cjp nz,exit;}

ORG 20h

go

\Блок завантаження команди з ОП

{or nil,r7,r7;ewl;oey;}

{xor nil,r7,r7;ewh;oey;}

{r;cjp rdm,cp;or r8,z,bus\_d;}

\Аналiз формату команди

{and nil,r8,8000h;load rm,flags;}

{cjp not rm\_z,exit;}

\Аналiз типу адресації

{and nil,r8,400h;load rm,flags;}

{cjp not rm\_z,exit;}

\Блок вибірки адреси операнда

{and r14,r8,03ffh;}

\Селектор мiкрокоманд

{or nil,r8,z;oey;JMAP;}

\Мiкропiдпрограма завантаження з пам'ятi

FromMem

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

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

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

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

{crtn nz;}

\Команда Swap r15,r13

swap

{or r11,r15,z;}

{or r15,r13,z;}

{or r13,r11,z;}

{cjp nz,next;}

mul

{cjs nz,FromMem;}

{or r11,z,17;}

{xor nil,r14,r15; load rm,flags;}

{or nil,r14,r14; cjp not no ,ll1;}

{sub r14, 0,r14,nz;}

ll1 {or nil,r15,r15; cjp not no ,ll2;}

{xor r15,r15,8000h;}

ll2 {cjp not rm\_c,ll3;}

{add r13,r13,r14,z;}

ll3 {or srl,r13,r13,z;}

{or sr.9,r15,r15,z;}

{sub r11,r11,z,z; load rm, flags;cem\_c;cem\_n;}

{cjp not rm\_z, ll2;}

{cjp not rm\_n,fin;}

{xor r15,r15,0FFFFh;}

{xor r13,r13,0FFFFh;}

{add r15,r15,1,z;}

{add r13,r13,0,rm\_c;}

fin {cjp nz,next;}

\Команда mov r15,[mem]

movToReg

{cjs nz,FromMem;}

{or r15,r14,z;}

{cjp nz,next;}

\Команда mov [mem],15

movToMem

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

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

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

{w;cjp rdm,cp;or nil,r15,z;oey;}

{cjp nz,next;}

\Команда or r15,[mem]

orr

{cjs nz,FromMem;}

{xor r15, r15, 0ffffh;}

{and r15,r15,r14;}

{cjp nz,next;}

\Команда jz [mem]

zjump

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

{cjs nz,FromMem;}

{sub r7,r14,1;}

no\_j {cjp nz,next;}

\Команда in r15,[mem]

input

{cjs nz,FromMem;}

{add r14,r14,2,z;}

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

{i;cjp rdd,cp;or r15,bus\_d,z;}

{cjp nz,next;}

\Команда out [mem],r15

output

{cjs nz,FromMem;}

{add r14,r14,2,z;}

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

{o;cjp rdd,cp;or nil,r15,z;oey;}

{cjp nz,next;}

\Команда test [mem]

testdevice

{cjs nz,FromMem;}

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

{i;cjp rdd,cp;or r14,bus\_d,z;}

{or nil,r14,z;load rn,flags;}

{cjp nz,next;}

\Завантаження адреси наступної команди

Next

{add r7,r7,2,z;}

{cjp nz,go;}

\Команда exit

exit {}

**6. Висновок**

В процесі виконання роботи була розроблена система команд для обчислення заданої функції з введенням вхідних даних та виведенням результату через зовнішні пристрої. Робота побазована на принципі мікропрограмного керування, алгоритм обчислення арифметичної функції був розбитий на окремі мікропідпрограми ( система команд високого рівня ), складені з команд системи команд низького рівня ( мікрокоманди Міколасемблеру ), яка зберігається в мікропрограмній пам’яті.

**7. Список літератури**

1. Жабін В.І., Ткаченко В.В. Цифрові автомати. Практикум. – К.: „ВЕК+”, 2003. -160 с.

2. Комп’ютерна схемотехніка. Проектування обчислювальних систкм з мікропрограмним керуванням: Метод. вказ. до виконання лаб. робіт/ Уклад.: Дичка І.А., Жабін В.І., Тарасенко В.П. – К.: НТУУ „КПІ”, 2006. – 64 с.

3. Жабін В.І., Жуков І.А., Ткаченко В.В., Клименко І.А. Мікропроцесорні системи: Навч. посібник. – К.: Вид-во „СПД Гуральник О.Ю.”, 2009. – 492 с